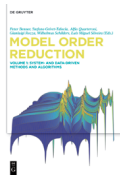
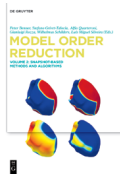
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.)
Model Order Reduction

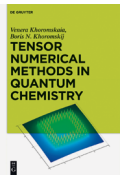
Also of Interest



Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.), 2020
ISBN 978-3-11-050043-1, e-ISBN (PDF) 978-3-11-049896-7,
e-ISBN (EPUB) 978-3-11-049771-7

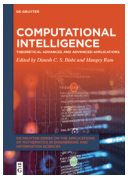


Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.), 2020
ISBN 978-3-11-067140-7, e-ISBN (PDF) 978-3-11-067149-0,
e-ISBN (EPUB) 978-3-11-067150-6



Tensor Numerical Methods in Quantum Chemistry
Venera Khoromskaia, Boris N. Khoromskij, 2018
ISBN 978-3-11-037015-7, e-ISBN (PDF) 978-3-11-036583-2,
e-ISBN (EPUB) 978-3-11-039137-4

Maxwell's Equations. Analysis and Numerics
Ulrich Langer, Dirk Pauly, Sergey Repin (Eds.), 2019
ISBN 978-3-11-054264-6, e-ISBN (PDF) 978-3-11-054361-2,
e-ISBN (EPUB) 978-3-11-054269-1



Computational Intelligence. Theoretical Advances and Advanced Applications
Dinesh C.S. Bisht, Mangey Ram (Eds.), 2020
ISBN 978-3-11-065524-7, e-ISBN (PDF) 978-3-11-067135-3,
e-ISBN (EPUB) 978-3-11-066833-9

Model Order Reduction

Volume 3: Applications

Edited by
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni,
Gianluigi Rozza, Wil Schilders, and Luís Miguel Silveira

DE GRUYTER

Editors
Prof. Dr. Peter Benner
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg
Germany
benner@mpi-magdeburg.mpg.de

Prof. Dr. Stefano Grivet-Talocia
Politecnico di Torino
Dipartimento di Elettronica
Corso Duca degli Abruzzi 24
10129 Turin
Italy
stefano.grivet@polito.it

Prof. Alfio Quarteroni
Ecole Polytechnique Fédérale de Lausanne (EPFL) and Politecnico di Milano
Dipartimento di Matematica
Piazza Leonardo da Vinci 32
20133 Milan
Italy
alfio.quarteroni@polimi.it

Prof. Dr. Gianluigi Rozza
Scuola Internazionale Superiore di Studi Avanzati - SISSA
Via Bonomea 265
34136 Trieste
Italy
gianluigi.rozza@sissa.it

Prof. Dr. Wil Schilders
Technische Universiteit Eindhoven
Faculteit Mathematik
Postbus 513
5600 MB Eindhoven
The Netherlands
w.h.a.schilders@tue.nl

Prof. Dr. Luís Miguel Silveira
INESC ID Lisboa
IST Técnico Lisboa
Universidade de Lisboa
Rua Alves Redol 9
1000-029 Lisbon
Portugal
lms@inesc-id.pt

ISBN 978-3-11-050044-8
e-ISBN (PDF) 978-3-11-049900-1
e-ISBN (EPUB) 978-3-11-049775-5
DOI https://doi.org/10.1515/9783110499001

Library of Congress Control Number: 2020944453

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

Cover image: Andrea Manzoni, MOX, Department of Mathematics, Politecnico di Milano
Typesetting: VTeX UAB, Lithuania
Printing and binding: CPI books GmbH, Leck

www.degruyter.com

Preface to the third volume of *Model Order Reduction*

The third volume of the *Model Order Reduction* handbook project offers several remarkable instances of applications of model order reduction (MOR) approaches to the solution of problems arising from the most diverse areas of application. Through these examples, we would like to provide the reader with an overview of the maturity of this emerging field and its readiness to address challenging problems of multifaceted complexity.

We start with several chapter contributions to classical fields of engineering.

The first one, by J. Eason and L. Biegler, is on model reduction in the optimization of a variety of heterogeneous chemical processes. In particular, two case studies are presented on CO_2 capture using nonlinear programming and NLP filter models.

The second chapter, by B. Lohmann et al., is on MOR in mechanical engineering. Four applications are discussed, concerning the reduction of a thermo-mechanical machining tool of a car body and driver's seat, of an elastic crankshaft, and a leaf spring model.

The third chapter, by E. Deckers et al., presents several case studies of MOR for acoustics and vibrations in mechanical applications. Two different viewpoints are developed: the application of MOR from a purely mathematical perspective and a consideration of expected properties of MOR based on physical arguments from the field of mechanics.

Two chapters devoted to microelectronics and electromagnetism, a very classical and successful arena for MOR methods, follow. The first of those, by B. Nouri et al., pursues a twofold goal: to describe the context in which the need for MOR arose in microelectronics, and to present an overview of their applications to address the issues of high-speed interconnects in microelectronics at various levels of the design hierarchy.

The next chapter, by D. Ioan et al., proposes a computer-aided consistent and accurate description of the behavior of electromagnetic devices at various speeds or frequencies, and describes procedures to generate compact electrical circuits featuring an approximately equivalent behavior.

The chapter by M. Yano is on model reduction in computational aerodynamics. The focus is on techniques that are designed to address nonlinearity, limited stability, limited regularity, and a wide range of scales that have been demonstrated successful for multidimensional large-scale aerodynamic flows.

The next two chapters address a somehow less conventional field of applications, that of life sciences. The chapter by B. Karasözen is on MOR in neurosciences, more specifically on the exploitation of models of large-scale neuronal networks to provide an accurate and fast prediction of patterns and their propagation in different areas of the brain.

The following chapter, by N. Dal Santo et al., introduces MOR methods to face some of the most challenging processes of the cardiovascular system. Two specific

https://doi.org/10.1515/9783110499001-201

applications are targeted: the simulation of blood flow past a carotid bifurcation and the computation of activation maps in cardiac electrophysiology.

The last five chapters address somewhat more methodological issues arising in various scientific, engineering, societal, and economics applications.

The chapter by J.-C. Loiseau aims at bypassing some difficulties of classical proper orthogonal decomposition approaches to the solution of fluid dynamics problems by using feature-based manifold modeling in which the low-dimensional attractor and nonlinear dynamics are characterized from experimental data: time-resolved sensor data and optional nontime-resolved particle image velocimetry snapshots.

In the chapter by R. Pulch, MOR is used in the framework of uncertainty quantification. Established methods like polynomial chaos, stochastic Galerkin, stochastic collocation, and quadrature sampling are reviewed for dynamical systems consisting of ordinary differential equations or differential algebraic equations. Demonstration of applicability is provided on test examples.

The chapter by X. Cheng et al. addresses MOR methods for networks that describe a wide class of complex systems composed of many interacting subsystems. First, clustering-based approaches are reviewed, with the aim of reducing the network scale. Then, methods based on generalized balanced truncation that reduce interconnection structures of a network and the dynamics of each subsystem are discussed.

The chapter by D. Hartmann et al. presents use cases where MOR is a key enabler for the realization of digital services and the reduction of simulation times and outlines the potential of MOR in the context of realizing the digital twin vision.

The last chapter, by B. Haasdonk, addresses the issue of software. In the first part, as neither full simulation models nor MOR algorithms are to be reprogrammed, but ideally are reused from existing implementations, the interplay of such packages is discussed. Then an overview of the most popular MOR software libraries is provided.

We are confident that the vast set of applications discussed here, combined with the broad variety of numerical techniques and software libraries available, will motivate the reader to embrace MOR approaches to successfully address complex applications arising in computational science and engineering.

Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza,
Wil Schilders, Luìs Miguel Silveira

Magdeburg, Germany
Torino, Milano, Trieste, Italy
Eindhoven, The Netherlands
Lisbon, Portugal

June 2020

Contents

John P. Eason and Lorenz T. Biegler

1 Model reduction in chemical process optimization

Abstract: Chemical processes are often described by heterogeneous models that range from algebraic equations for lumped parameter systems to black-box models for PDE systems. The integration, solution, and optimization of this ensemble of process models is often difficult and computationally expensive. As a result, reduction in the form of reduced-order models and data-driven surrogate models is widely applied in chemical processes. This chapter reviews the development and application of reduced models (RMs) in this area, as well as their integration to process optimization. Special attention is given to the construction of reduced models that provide suitable representations of their detailed counterparts, and a novel trust region filter algorithm with reduced models is described that ensures convergence to the optimum with truth models. Two case studies on CO_2 capture are described and optimized with this trust region filter method. These results demonstrate the effectiveness and wide applicability of the trust region approach with reduced models.

Keywords: Model reduction, trust region methods, POD, equation-oriented modeling, glass box, black box, nonlinear programming, NLP filter methods

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

1.1 Introduction

Chemical processes incorporate advanced technologies that need to be modeled, integrated, and optimized. To address these needs, state-of-the-art nonlinear optimization algorithms can now solve models with millions of decision variables and constraints. Correspondingly, the computational cost of solving discrete optimization problems has been reduced by *several orders of magnitude* [14]. Moreover, these algorithmic advances have been realized through software modeling frameworks that link optimization models to efficient nonlinear programming (NLP) and mixed-integer NLP solvers. On the other hand, these advances are enabled through modeling frameworks that require optimization models to be formulated as well-posed problems with exact first and second derivatives.

Despite these advances, *multiscale* processes still need effective problem formulation and modeling environments. At the process optimization level, multiscale in-

John P. Eason, Exenity, LLC, Pittsburgh, PA, USA
Lorenz T. Biegler, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

https://doi.org/10.1515/9783110499001-001

tegration is required to model complex transport and fluid flow phenomena. For instance, optimization models for advanced power generation processes, such as in Figure 1.4, comprise a heterogeneous modeling environment with algebraic equation (AE) models, such as heat exchangers, compressors, and expanders, as well as large, nonlinear partial differential AE (PDAE) models. These include multiphase reactor models such as fluidized beds, combustors, and gasifiers. Because of the substantial complexity of the associated model solvers, computational costs for multiscale process optimization are prohibitive. While equation-oriented flowsheet models may take only a few CPU seconds to solve, a PDAE combustion or gasification model alone may require many CPU hours or even days [48]. We denote these prohibitive models as *truth models*, which require model reduction. These models often follow a "bottom up" modeling approach, where DAEs or PDAEs derive from fundamental physical laws. Models at this higher fidelity can include transport behavior and detailed reaction kinetics, which require computationally costly simulations.

The process flowsheet in Figure 1.4 shows a detailed boiler model, pumps, compressors, turbines, heat exchangers, and mixing and splitting junctions, and the resulting model comprises equations that connect process units with process streams, conservation laws (mass, momentum, energy) within each unit, constitutive equations that describe physical phenomena, including transport behavior, equilibrium and reaction kinetics, and physical properties for materials and mixtures. This ensemble of PDAEs/DAEs/AEs within a chemical process is typically broader than many PDAE models in other domains, where model reduction can proceed in a more structured manner.

Model reduction in chemical process simulation and optimization can be effected in a number of ways. These include:

- Simplifying assumptions of physics-based models by removing terms in PDAEs that reflect negligible behavior in time and length scales. Often, these "shortcut" models can only be defined (and validated) over limited domains of applicability.
- Time scale reduction, where dynamic behaviors that are either too fast or too slow in the range of interest are eliminated [56].
- Data-driven input–output models which are generally unstructured, require few assumptions, and lead to general-purpose applications [63].

Process optimization with reduced models poses a special challenge as most reduced models are interpolative, while optimization requires extrapolation. Generally, we expect extrapolative capabilities to be captured by physics-based truth models, as they are based on fundamental phenomena and comprise constituent models that have been validated from many domains. To develop and preserve these capabilities, interpolative reduced models must be reconstructed and recalibrated with truth models, in order to remain consistent over the convergence path of the optimization solver.

To develop an integrated optimization approach we expect that an RM-based strategy is allowed to evaluate and compare information from the truth models to cap-

ture relevant multiscale phenomena such as complex fluid flow, particle mechanics, and dynamic operation within the process optimization. Moreover, general strategies can be applied to create RMs through a number of physics-based or data-driven model reductions. To develop this strategy, this chapter considers the properties needed for the RM-based optimization framework to converge to the optimum of the original system models, as well as the construction of RMs that balance model accuracy with computational cost during the optimization.

The next section briefly reviews developments in model reduction that include proper orthogonal decomposition (POD) and data-driven models. Section 1.3 then presents trust algorithms based on reduced models; these depend on whether gradients are available from the truth models or not. Section 1.4 then presents two case studies that describe the performance of these methods on large-scale process optimization problems. Most importantly, both methods work directly with the RM, and they also guarantee convergence to the optimum of the truth model. Finally, conclusions and future work are discussed in Section 1.5.

1.2 Model reduction for simulation

Model reduction is a broad topic with contributions from many different research communities. There has always been a balance between model fidelity and computational tractability since the earliest use of computers in chemical engineering. For instance, for vapor-liquid equilibrium, which is the basic building block of all process models, reduced physical property models are often constructed through simplifying thermodynamic assumptions. In early studies [8, 17, 50] these reduced models proved very effective to accelerate calculations without sacrificing much accuracy. While computing hardware has improved substantially since that time, these early works show how model reduction can be used to solve problems that otherwise may be intractable.

Beyond the use of simplifying fundamental assumptions with "shortcut" models, general model reduction strategies can be partitioned into two categories: model order reduction and data-driven model reduction.

We classify model order reduction methods as projection-based techniques applied to an accessible state-space description of the truth model, with an explicit projection applied to reduce the state-space dimension [11]. For fully accessible (i. e., equation-based) state-space truth models, *system-theoretic* model order reduction exploits the specific dynamic system structure and includes balanced truncation and rational interpolation based on Gramians and transfer functions. These usually apply to linear systems, although they have been extended to bilinear systems and quadratic-in-state systems. Moreover, they are widely applied in circuit theory, signal processing, structural mechanics, and linear, optimal control. A comprehensive review of system-theoretic methods can be found in [9].

When the truth model is a large-scale system of DAEs, system-theoretic methods can take advantage of that structure. In model-based reduction for chemical engineering, the truth model is also exploited to guide the projection steps [56], but system-theoretic model order reduction is seldom applied to these models. This is mainly because of the high model nonlinearity and limited accessibility of the chemical process model equations, often embedded within "gray-box" procedures. To handle these truth models, snapshot-based projection methods, such as reduced basis or POD, are applied, with sampled snapshot solutions over the parameter domain and space (or time) domains. Among these, POD is the most generally applicable as it relies only on snapshots of the underlying simulation code. POD has been demonstrated effectively in many areas including fluid dynamics, structural dynamics, thermal modeling, and atmospheric modeling. As a result, it is frequently applied for model-based reduction of large, nonlinear truth models in chemical engineering.

1.2.1 Proper orthogonal decomposition

POD, also known as Karhunen–Loève decomposition, can reduce large spatially distributed models to much smaller models. POD models are formulated by projecting the PDAE system onto a set of basis functions, which are themselves generated from the numerical solution of the original equations. Applications are numerous, with examples including [12, 22, 24, 28, 42, 44, 47, 53, 57, 60, 62, 71, 73, 21]. In addition, many studies report the use of POD for optimization. However, the basis functions used in POD are typically determined from a finite set of simulations of the full-scale PDAE system. This greatly reduces the system size, but the accuracy of the POD approximation is inherently local in nature. Therefore optimization may have a tendency to extrapolate far from data or otherwise exploit approximation errors to find artificially improved solutions.

Nevertheless, several studies report successful use of model order reduction in optimization and control. Examples in chemical processes include optimization of diffusion–reaction processes [5], transport-reaction processes [10], chemical vapor deposition [66], and thermal processing of foods [6].

As detailed in [67], POD models are constructed from Galerkin projections of the PDAE model onto a set of basis functions. These basis functions are often generated empirically from numerical solutions of the truth model, through the *method of snapshots*. The aim is to find a low-dimensional basis that can capture most information of the spatial distribution. To do so, one first gathers snapshot sets which consist of spatial solutions of the original PDAE system at several time instants as determined by numerical simulation. Let the snapshot matrix be given as

$$Z = \{z(\xi, t_1), \ldots, z(\xi, t_{N_t})\}, \tag{1.1}$$

where each snapshot $z(\xi, t_j)$ is a column vector representing the (discretized) spatial profile at time t_i. There are N_t snapshots and N_ξ spatial discretization nodes.

After gathering a set of snapshots, the singular value decomposition of the snapshot matrix Z is given as

$$Z = UDV^T = \sum_i \sigma_i u_i v_i^T. \tag{1.2}$$

The first M vectors $\{u_i\}_{i=1}^M$, where $M \leq N_\xi$, of the orthogonal matrix U represent the desired set of POD basis functions (or basis vectors in the case of discretized spatial dimensions). From this point we refer to these basis functions as $\phi_i(\xi)$, since each one describes the behavior in the spatial dimensions.

To determine the number of basis vectors M, the projection error can be approximated as

$$\varepsilon^{\mathrm{POD}}(M) = \sum_{i=M+1}^{N_\xi} \sigma_i^2. \tag{1.3}$$

Then, the interrelation between accuracy and dimension of the POD-based reduced-order models can be balanced by a predetermined threshold. The *error bound* λ is defined as

$$\lambda(M) = 1 - \frac{\sum_{i=1}^M \sigma_i^2}{\sum_{i=1}^{N_\xi} \sigma_i^2}. \tag{1.4}$$

M is then chosen such that $\lambda(M) \leq \lambda^*$ for a desired threshold λ^* [64]. Typically, M can be chosen rather small compared to N_ξ while still keeping λ close to zero (typically $< 10^{-3}$).

After computing the POD basis set, a reduced-order model is derived by projecting the PDAEs of the system onto the corresponding POD subspace. This means that we seek an approximation of the form

$$z(\xi, t) \approx z_{\mathrm{POD}}(\xi, t) = \sum_{i=1}^M a_i(t)\phi_i(\xi). \tag{1.5}$$

To demonstrate how the Galerkin approach is applied to determine the coefficients $a_i(t)$, consider a PDE in the following form:

$$\frac{\partial z}{\partial t} = f\left(z, \frac{\partial z}{\partial \xi}\right). \tag{1.6}$$

Using the POD basis functions as the weighted basis functions for the Galerkin projection, we obtain the system

$$\frac{da_i}{dt} = \int f\left(\sum_{j=1}^M a_j(t)\phi_j(\xi), \sum_{j=1}^M a_j(t)\frac{d\phi_j}{d\xi}\right)\phi_i(\xi)\, d\xi, \quad i = 1 \ldots M, \tag{1.7}$$

leading to a set of M ordinary differential equations (ODEs). If the spatially discretized system were directly solved with the method of lines, it would consist of N_ξ ODEs. Since M is normally much less than N_ξ, POD can create a much smaller model that still maintains reasonable accuracy.

1.2.2 Data-driven reduction

Data-driven reduction methods have been successfully applied to truth models where projections and basis functions cannot be generated from the model equations, and only input/output responses are available from a black-box code. This black box may be sampled, and regression/interpolation approaches can be used to fit the sampled data. The resulting surrogate model replaces the truth model for simulation, optimization, or other analysis. There is considerable flexibility in the functional form and fitting methods used for surrogate construction, and this flexibility can be used to customize an approach suitable for a particular problem. Simpson et al. [63] provide a review of the field, which outlines several important steps and existing surrogate modeling frameworks. The main steps of surrogate model construction include experimental design, model selection, and model fitting. Several established methodologies suggest combinations of choices for each of these three steps. For example, response surface methodology, typically used in optimization settings, uses central composite designs in combination with quadratic models constructed with least-squares regression. The central composite design helps determine curvature information for the quadratic models. A more complete description can be found in Myers and Montgomery [59]. In some ways, response surface methodology is a predecessor to the trust region-based methods that will be discussed in Section 1.3. Other surrogate modeling approaches include Gaussian process regression (including kriging) and artificial neural networks. These methods often perform better with space-filling or sequential experimental designs [45, 36].

Recent work also examines the role of model complexity in surrogate modeling. When simpler functional forms are preferred, best-subset techniques combined with integer programming can be used to fit models [29, 72]. Moreover, recent developments in machine learning have led to a wealth of new methods for model reduction [58, 65].

An example that demonstrates many concepts from data-driven model reduction may be found in [48]. That work proposes a model reduction method for distributed parameter systems based on principal component analysis and neural networks. The reduced model is designed to represent the spatially distributed states z of the system as functions of the inputs w, including boundary conditions, equipment parameters, operating conditions, and input stream information. Similar to POD, this PCA approach seeks to represent the states in terms of a set of empirically determined basis functions. First a set of snapshots is determined by running the truth model at

various input conditions $W = \{w_1 \dots u_{n_n}\}$, giving the snapshot set

$$Z = \{z(\xi, w_1), \dots, z(\xi, w_N)\}, \tag{1.8}$$

where each snapshot $z(\xi, w_i)$ is a column vector representing the spatial profile at input point w_i. There are n_n snapshots and N_ξ spatial discretization nodes. The basis functions are obtained in the same manner as discussed with equations (1.2), (1.3), and (1.4). After obtaining the reduced basis set $\phi_i(\xi)$ (the principal components), the reduced model is expressed as

$$z(\xi, w) \approx z_{\mathrm{PCA}}(\xi, w) = \sum_{i=1}^{M} a_i(w)\phi_i(\xi). \tag{1.9}$$

Whereas POD determines the coefficients a_i through Galerkin projection of the truth model equations onto the basis set, the PCA-RM approach of [48] uses neural networks. In other words, each function $a_i(w)$ is the result of training a neural network to capture the nonlinear relation between the input variables and the states represented with the principal components. This PCA-RM approach was applied to a CFD model of an entrained flow gasifier, embedded within an open-equation advanced power plant model [49]. The truth model was implemented in Fluent and takes around 20 CPU hours to solve. With both high computational cost and the use of commercial tools that may be difficult to use for custom analysis and simulations, this problem has both motivating features for the use of reduced models. There were three input variables for the truth model, including the water concentration in the slurry, oxygen to coal feed ratio, and the ratio of coal injected at the first feed stage. As described in [49], the resulting PCA-based RM had very good accuracy, as validated by leave-one-out cross-validation.

1.3 Process optimization using reduced models

Many process engineering applications on reduced modeling involve optimization formulations. One of the challenges in this field is the size of the problems created if the system is fully discretized before optimization. In addition, optimization routines are not easily customized to handle a particular problem as simulation. Here, a reduction in problem size can greatly speed solutions to enable real time application.

However, despite significant effort in building reduced models, it is known that using an RM in optimization can lead to inaccurate answers. Small errors introduced by the RM approximation can propagate through to large errors in the optimum solution. This is worsened by the optimization's tendency to exploit error to artificially improve the objective function, and hence optimization may terminate in regions of

poor RM accuracy. The RM can be refined sequentially during optimization, using information from the truth model to improve inaccuracies. However, these iterative improvements offer no convergence guarantees to the optimum of the high-fidelity optimization problem, even if it does converge at a point where the RM matches the truth model.

The nonconvergence behavior can be observed through a toy problem in [15, 16], shown as follows:

$$\begin{aligned} \min \quad & f(x) = \left(x^{(1)}\right)^2 + \left(x^{(2)}\right)^2 \\ \text{s.t.} \quad & t(x) = x^{(2)} - \left(x^{(1)}\right)^3 - \left(x^{(1)}\right)^2 - 1 = 0, \end{aligned} \tag{1.10}$$

where we denote the cubic function $t(x)$ as the truth model. As shown in Figure 1.1, the problem (1.10) has a global minimum, $(x^*) = (0,1), f(x^*) = 1$, and a local maximum, $(x^*) = (-1,1), f(x^*) = 2$. Now consider the corresponding RM-based problem given by

$$\begin{aligned} \min \quad & \hat{f}(x) = \left(x^{(1)}\right)^2 + \left(x^{(2)}\right)^2 \\ \text{s.t.} \quad & r(x) = x^{(2)} - x^{(1)} - b = 0, \end{aligned} \tag{1.11}$$

where we denote the linear function $r(x)$ as the RM with an adjustable constant b. It is straightforward to show that the solution to (1.11) is given by $x^* = (-b/2, b/2)$, and $f(x^*) = b^2/2$. Moreover, as shown by the steps in Figure 1.1(a), the RM-based algorithm proceeds at iteration k by choosing b_k so that $r(x_k) = t(x_k) = 0$. Then (1.11) is solved to obtain the next iterate x_{k+1}. However, this approach does not guarantee convergence to the optimum for the truth model. For instance, if we start from $(0,1)$, the global minimum solution of (1.10), Figure 1.1(a) shows that the iterates x_k actually move away from this solution and eventually converge to a nonoptimal point where $t(\bar{x}) = r(\bar{x})$, $b = 2$, and $\bar{x} = (-1,1)$. For this example, it can be seen that this point is actually a *local maximum* of (1.10). This behavior arises because optimality conditions rely on derivative information, not simple matching in function values.

The most common approach to "safe" optimization with reduced models is to use a trust region method. Instead of approximating a black-box function over the entire domain of decision variables, a reduced model is constructed to locally approximate over this trust region. Assuming sufficient data are available, smaller domains can lead to lower absolute error in the reduced model and the choice of functional form becomes less critical with the restricted approximation. Trust region methods exploit this feature by adapting the trust region radius during the optimization process.

Most trust region algorithms adopt the following outline. As a general example assume that the goal is to solve the following optimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0. \end{aligned} \tag{1.12}$$

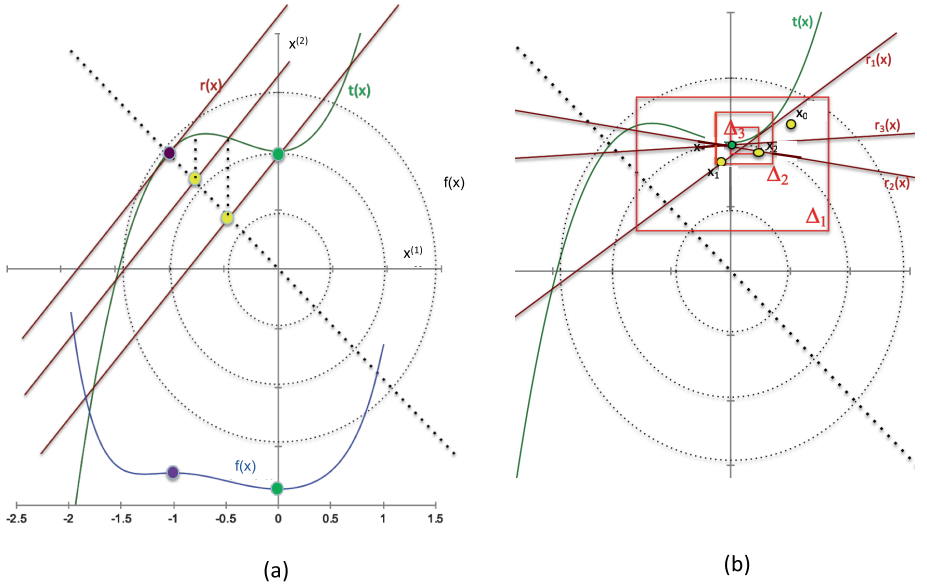


Figure 1.1: Characteristics of toy example. (a) Convergence failure. (b) Convergence with trust region method.

The initial point is denoted as x_0. At each iteration, a trust region method will first *construct an approximation* of (1.12) that is valid on the trust region at iteration k. In other words, identify $\hat{f}$ and $\hat{g}$ such that

$$\hat{f}(x) \approx f(x) \quad \text{and} \quad \hat{g}(x) \approx g(x) \quad \text{for } x \in B(x_k, \Delta_k),$$

where $B(x_k, \Delta_k) = \{x : \|x - x_k\| \leq \Delta_k\}$ is the trust region at iteration k. In classical trust region methods for nonlinear optimization, a quadratic approximation of the objective function is constructed, while the constraint functions are linearized. However, alternative forms may be used and the characteristics of the optimization algorithm change depending on the type of approximation (type of RM) and the nature of accuracy required for the approximation.

The second step is to *solve the trust region subproblem*. This means that the reduced models are used to optimize the function within the trust region, where sufficient accuracy is assumed. For our example problem (1.12), the trust region subproblem is

$$\begin{aligned} \min_{x} \quad & \hat{f}(x) \\ \text{s.t.} \quad & \hat{g}(x) \leq 0, \\ & \|x - x_k\| \leq \Delta_k. \end{aligned} \tag{1.13}$$

The trust region constraint helps prevent the algorithm from extrapolating into regions where the RM is not accurate, and provides a globalization mechanism to make

sure the algorithm converges. On the other hand, algorithms vary on how the trust region subproblem is solved, and even the type of subproblem considered (constrained vs. unconstrained, convex or nonconvex).

The final step is to *evaluate the solution* proposed by the trust region subproblem, denoted as $\bar{x}$. Using recourse to the truth model, the solution can be evaluated in terms of accuracy. Depending on the improvement at $\bar{x}$, the algorithm determines that either $x_{k+1} = \bar{x}$, in which case we say that the step was accepted, or $x_{k+1} = x_k$, in which case we say that the step was rejected. The algorithm also determines the trust region radius for the following iteration Δ_{k+1}. There is significant freedom in algorithm design to handle this last step, with various ways to decide on when to accept or reject the step and how to update the trust region radius.

Alexandrov et al. [3] applied the trust region concept to general reduced models in engineering. They considered the task of unconstrained minimization using arbitrary approximation functions in place of the actual function. Reduced models are constructed so that the function and gradient values of the reduced model match those of the truth model at the center of the trust region. The trust region subproblem was the minimization of the reduced model subject to the trust region constraint. Standard methods for unconstrained trust region methods were used to evaluate the step, including the ratio test [26]. Convergence was proved to the optimum with the truth model. These concepts were extended to classes of constrained optimization problems in the DAKOTA package [43]. Moreover, early work in [4, 39] demonstrated the use of POD reduced models with a trust region method. This algorithm was shown to be convergent under the assumption that the gradients of the POD model are sufficiently accurate, although this condition may be difficult to verify in practice.

To illustrate the RM-based trust region approach, we revisit problem (1.10) but consider the NLP associated with the following RM:

$$\begin{aligned} \min \quad & \hat{f}(x) = \left(x^{(1)}\right)^2 + \left(x^{(2)}\right)^2 \\ \text{s.t.} \quad & r(x) = x^{(2)} - ax^{(1)} - b, \end{aligned} \tag{1.14}$$

where the RM has adjustable constants a and b. The corresponding trust region problem is given by

$$\min_s \quad \left(x_k^{(1)} + s\right)^2 + \left(a_k\left(x_k^{(1)} + s\right) + b_k\right)^2 \tag{1.15}$$

$$\text{s.t.} \quad \|s\|_\infty \leq \Delta_k, \tag{1.16}$$

and the progress of the trust region algorithm is sketched in Figure 1.1(b). Using $\Delta_0 = 0.8$, the trust region algorithm converges to a tight tolerance after 20 iterations [16].

For the case where gradients are not available from the truth model, Conn et al. [27] discuss sufficient accuracy conditions on the reduced model to guarantee convergence. This condition, called the κ-fully linear property, can be verified for data-driven

reduced models (e. g., polynomial interpolation). The κ-fully linear property (see (1.20) below) dictates how the differences between reduced and truth models must scale directly with the trust region radius. In this way, shrinking the trust region allows first-order optimality to be guaranteed in the limit. These derivative-free trust region ideas were extended by March and Wilcox [55] to consider the use of multifidelity models. In that study, the reduced model was a coarse discretization of the PDE system. Wild, Regis, and Shoemaker also use the framework of κ-fully linear models to develop an algorithm with radial basis functions as RMs [70]. March and Wilcox [55] use constrained trust region subproblems with reduced models, with globalization managed with the use of a merit function. More recent extensions of these optimization strategies and applications for PDAE systems are reviewed in [61].

While the RM-based trust region methods guarantee convergence to the truth model-based optimization problem, the RM itself needs to be updated frequently, potentially for each trust region subproblem. To address this limitation, recent work on construction of RMs with embedded parameters (e. g., decision variables) appear to be particularly promising [11]. In particular, the idexempirical interpolation method (EIM) and the discrete EIM (DEIM) develop RMs that contain an interpolant for parameter values [7, 25]. This allows fast updates of the RM as part of the optimization process, with much less evaluation of the truth models.

Finally, for the optimization of multiscale chemical processes, Caballero and Grossmann [23] use kriging models to represent unit operations for process optimization and trust regions were used to shrink the domain of the kriging models, though convergence to local optima was not proved. Agarwal et al. [1] consider the optimization of periodic adsorption processes with POD-based reduced models. For this system they analyze and present convergence results in [2] for constrained subproblems when derivatives of the truth models are known. For the related simulated moving bed (SMB) process with linear isotherms, Li et al. [52] develop and apply system-theoretic model order reduction methods to accelerate the computation of the cyclic steady states and optimize the SMB system. In a related study, these authors develop and demonstrate an efficient trust region method with surrogate (POD as well as coarse discretization) models to optimize the SMB process [51]. Biegler et al. [16] use a penalty function to solve inequality-constrained problems when the derivatives are unavailable and suggest stopping criteria based on the reduced model errors. Moreover, Bremer et al. [20] apply a POD-DEIM method to a dynamic, two-dimensional reactor model for CO_2 methanation. In their dynamic studies they demonstrate that the resulting RM is accurate and accelerates the solution of the truth model by over an order of magnitude.

More recently, reduced models have also been used in global optimization/global search using "gray-box" models [13, 18, 19]. In this broader class of problems, simplified models stand in for challenging or computationally expensive modeling elements. However, dimensionality of these cases is restricted and asymptotic behavior is ignored because of tight budgets on function calls.

In the next section, we summarize our work on a trust region method that provides rigorous convergence guarantees to the truth model-based optimum. By extending concepts from classical optimization theory to RM-based optimization, an algorithm is developed to automatically manage RMs and prevent inaccurate solutions. Building off the κ-fully linear theory for RMs from [27] and classical optimization theory, the proposed algorithm is extensible to a wide range of RM-based optimization problems for chemical processes.

1.3.1 A trust region filter approach for RM-based optimization

In order to address process optimization problems with heterogeneous models illustrated in Figure 1.4, we consider a slight extension of Problem (1.12). Let $t(w)$ represent a high-fidelity truth model, which will be approximated with reduced-order models $r_k(w)$. We also refer to the truth model $y = t(w)$ as a *black-box* model, which is embedded within a larger optimization problem as follows:

$$\begin{aligned} \min_{w,y,z} \quad & f(w,y,z) \\ \text{s. t.} \quad & h(w,y,z) = 0, \\ & g(w,y,z) \leq 0, \\ & y = t(w). \end{aligned} \tag{1.17}$$

Here $w \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$, and the functions f, h, g are assumed to be twice differentiable on the domain $\mathbb{R}^{m+n+p}$ and form the equation-oriented process model. We refer to these functions as the *glass-box* model, where accurate derivatives are assumed to be cheaply available, e. g., through the use of automatic differentiation. The high-fidelity truth model is shown as a map $t(w) : \mathbb{R}^m \to \mathbb{R}^p$ taking input variables w; $y \in \mathbb{R}^p$ represents a lifted variable that represents the output of this black-box model. This lifting isolates the truth model from the glass-box model equations and allows its replacement with a (glass-box) reduced model. The remaining decision variables are represented by z.

In chemical processes, the black-box $y = t(w)$ often represents a complex PDE-based unit operation, while the remaining glass-box constraints h and g represent the rest of the process. This allows a model to be multiscale in that a detailed model of transport and reactions can be coupled with process-level equation-oriented models. For simplicity, we introduce an aggregated variable vector $x^T = [w^T, y^T, z^T]$.

To allow for convergence of the trust region algorithm, we also assume that $t(w)$ is twice continuously differentiable. In comparison to the generic formulation in (1.12), this form is chosen so that the only approximation task is the replacement of $t(w)$ with a reduced model. The remaining functions, including objective function and constraints, remain unaltered in the trust region subproblem (1.18). With $t(w)$ replaced by a glass-box reduced model $r_k(w)$ and a trust region constraint added to confine x to a

domain where the reduced model is presumed to be accurate, the trust region radius Δ_k and trust region center x_k are then updated by the algorithm to guarantee convergence:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s. t.} \quad & h(x) = 0, \\ & g(x) \leq 0, \\ & y = r_k(w), \\ & \|x - x_k\| \leq \Delta_k. \end{aligned} \tag{1.18}$$

The trust region filter (TRF) method for solving optimization problems with embedded RMs is presented in [37]. This algorithm combines filter methods for constrained NLP with model-based trust region methods developed in [27]. The TRF framework controls errors in the reduced model while balancing improvement in feasibility and optimality. The algorithm is based on the sequential quadratic programming filter method in Fletcher et al. [40] and its extension to inexact Jacobians by Walther and Biegler [68]. Borrowing concepts from multiobjective optimization, the filter examines trade-offs in constraint violation and objective function improvement.

We now briefly outline the algorithm, discuss convergence, and demonstrate the ability to solve practical problems. At each iteration, the TRF method constructs or obtains an RM with some specification on desired accuracy. Then, the RM is checked for compatibility with the rest of the model. Even though problem (1.17) may be feasible, the trust region subproblem (1.18) may be infeasible due to inaccuracies of the RM $r_k(w)$. For the convergence proof, compatibility is enforced by a slightly stricter condition, and requires that a feasible point lie within the ball $B(x_k, \Delta_k \min[1, \kappa_\mu \Delta_k^\mu])$, where κ_μ and μ are tuning parameters. After verifying compatibility, a criticality check is performed to determine whether the algorithm may be near convergence.

Once the trust region subproblem (1.18) is solved, the solution, x_k, is used to check the accuracy of the RM. The truth model is evaluated at the proposed solution, $t(w_k)$, and this value and the objective value $f(x_k)$ determine whether to accept or reject the proposed step using the filter method [41]. For the purposes of RM-based optimization of form (1.17), constraint violations are equivalent to the inaccuracy of the RM.

Depending on the accuracy condition on the reduced model, the algorithm can take several forms. Agarwal and Biegler [2] consider the case where the function and gradient values of the reduced model must agree with the truth model, i. e., $t(w_k) = r_k(w_k)$ and $\nabla t(w_k) = \nabla r_k(w_k)$. This requires accurate gradient information from the truth model. In fact, to construct these models for *any* RM, $\tilde{r}(w)$, one can define [3]

$$r_k(w) = \tilde{r}(w) + \big(t(w_k) - \tilde{r}(w_k)\big) + \big(\nabla t(w_k) - \nabla \tilde{r}(w_k)\big)^T (w - w_k), \tag{1.19}$$

i. e., through zero-order and first-order corrections, as shown in (1.27) and (1.28), respectively.

On the other hand, when the truth model gradients are unavailable, one can use the κ-fully linear property instead, as developed in [37] and defined as follows.

A model $r_k(w)$ is said to be κ-fully linear on $B(w_k, \Delta_k)$ for constants κ_f and κ_g if, for all $w \in B(w_k, \Delta_k)$,

$$\left\| r_k(w) - t(w) \right\| \leq \kappa_f \Delta_k^2 \quad \text{and} \quad \left\| \nabla r_k(w) - \nabla t(w) \right\| \leq \kappa_g \Delta_k. \tag{1.20}$$

The constants $\kappa_f > 0$ and $\kappa_g > 0$ can be derived from the particular form of $r_k(w)$ and the data used to construct $r_k(w)$ if applicable. We assume they are uniformly bounded, but exact values of these constants are not needed in the algorithm.

This property requires that the accuracy of the RM needs to behave in a similar manner to a first-order Taylor expansion. This accuracy condition is easily verified when using interpolation models, such as polynomials [27] or radial basis functions [69]. Of course, if $r_k(w)$ is given by (1.19), the κ-fully linear property applies directly.

1.3.2 Summary of the trust region filter algorithm

A detailed description of the TRF algorithm along with the convergence analysis and properties can be found in [37]. A summary of the key steps is listed below.

1. *Initialization*: Choose the initial point x_0 and trust region size Δ_0, as well as constants that guide the expansion and contraction of the trust region and the initial filter. Evaluate $t(w_0)$ and then calculate $\theta(x_0) = \|y_0 - t(w_0)\|$ and set $k = 0$.
2. *RM construction*: If $\nabla t(w_k)$ is available, calculate an RM $r_k(w)$ using (1.19). Else, generate $r_k(w)$ that is κ-fully linear on Δ_k.
3. *Compatibility check*: Solve a *compatibility problem* to determine if the trust region problem (1.18) is feasible. If so, go to Step 4. Otherwise, add (θ_k, f_k) to the filter and go to Step 7.
4. *Criticality check*: At x_k compute a criticality measure for problem (1.18) without the trust region constraint, and check whether the Karush–Kuhn–Tucker (KKT) conditions hold within tolerance. If this condition holds and the RM was constructed from (1.19), then STOP. Because $\nabla t(w_k) = \nabla r(w_k)$ then, the KKT conditions of problem (1.17) hold as well. If $\nabla t(w_k)$ is unavailable and the criticality check holds, reassign $\Delta_k := \omega \Delta_k$ and go to Step 2. Else, continue to Step 5.
5. *Trust region step*: Solve the subproblem (1.18) to compute a step s_k.
6. *Filter:* Evaluate $\theta(x_k + s_k)$ and determine whether it is sufficiently decreased from $\theta(x_k)$. Follow the filter steps described in [37] to manage the trust region size, update the filter and accept $x_{k+1} = x_k + s_k$, or reject the step and set $x_{k+1} = x_k$. Return to Step 2. (A detailed description of all of the steps in the filter procedure is presented in [37].)
7. *Restoration phase:* As described in detail in [37], compute x_{k+1}, Δ_{k+1}, and r_{k+1} that is compatible to (1.18) and is acceptable to the filter. Set $k = k + 1$ and return to Step 2.

The global convergence analysis of the above algorithm requires standard assumptions: κ-fully linear RMs, smoothness of the underlying functions, and regularity of limit points for problems (1.18) and (1.17). The convergence proof is based on the TRF analysis of Fletcher et al. [40]. With slight modifications [37] it extends to (1.18) to show existence of a subsequence x_{k_j} for which $x_{k_j} \to x^*$, where x^* is a first-order KKT point of (1.17). This global analysis also shows that feasibility and criticality measures limit to zero at x^*. Full details of the convergence proof can be found in [37].

1.4 RM-based process optimization case studies

This section presents two challenging case studies with complex truth models, the first with available truth model gradients, and the second without. These cases demonstrate the effectiveness of the TRF algorithm for RM-based process optimization. The first case study considers the optimization of a pressure swing adsorption process for carbon capture using POD reduced-order models, where we assume gradients are available from the truth model. The second case study considers optimization of an oxycombustion power plant, in which the boiler is modeled with a one-/three-dimensional hybrid zonal approach, leading to a PDE-based truth model, which is reduced using data-driven methods. Here the truth model does not provide gradient information. Additional studies that demonstrate trust region methods for RM-based optimization can be found in [16, 49, 38].

1.4.1 Pressure swing adsorption

Pressure swing adsorption (PSA) technology is widely applied for separation and purification of gas mixtures. By feeding gas through a bed of solid adsorbent, a stream enriched in the less strongly adsorbed components is produced. This normally occurs at high pressure to favor more adsorption. When pressure is reduced, equilibrium is shifted and the adsorbed components will desorb, thus regenerating the solid adsorbent for reuse. The desorbed gas is enriched in the more strongly adsorbed components of the feed gas. By cycling through adsorption and desorption, the feed gas stream is separated into two purified streams.

This cyclic process of adsorption and desorption driven by changing pressure lends the technology its name. PSA normally occurs in one or more packed beds. Because feed gas is only fed during the pressurization step and the outlet gas composition changes for each step, it is common to have multiple beds in parallel to maintain continuous operation. Because PSA is often used in continuous processes, the beds are operated in cyclic steady state. For modeling purposes, cyclic steady-state operation requires imposition of a periodic boundary condition which enforces the

same state at the end of the cycle as the beginning. The conditions within each bed can be described by nonlinear PDAEs for complex fluid flow and nonlinear adsorption isotherms. The solution of these PDAEs is governed by steep adsorption fronts, which require a large number of discretization points. In addition to the large set of discretized equations from the PDAE, the optimization model for PSA includes cyclic steady-state conditions and stringent purity constraints. The scale of this optimization problem, whether considering design or operation, presents a significant challenge to state-of-the-art nonlinear programming algorithms. This motivates the use of model reduction to reduce the size of this system.

A two-bed four-step isothermal PSA process is shown in Figure 1.2. The feed mixture is 85 % N_2 and 15 % CO_2. The four steps of operation are pressurization, adsorption, depressurization (counter-current), and light reflux (or desorption). The two beds are operated as follows. First, bed 1 is pressurized with feed gas while bed 2 is depressurized, producing a stream rich in the strongly adsorbed component. Next, high-pressure feed gas is continually added to bed 1, and the heavy component continues to adsorb, producing a product rich in the weakly adsorbed component (light product). A fraction of the light product gas is fed to bed 2 at low pressure to purge and further desorb the accumulated heavy adsorbate. This is called the light reflux step. Next, the two beds interchange roles, with bed 1 first depressurizing and then receiving light reflux. Thus, with four steps, the system returns to the original state.

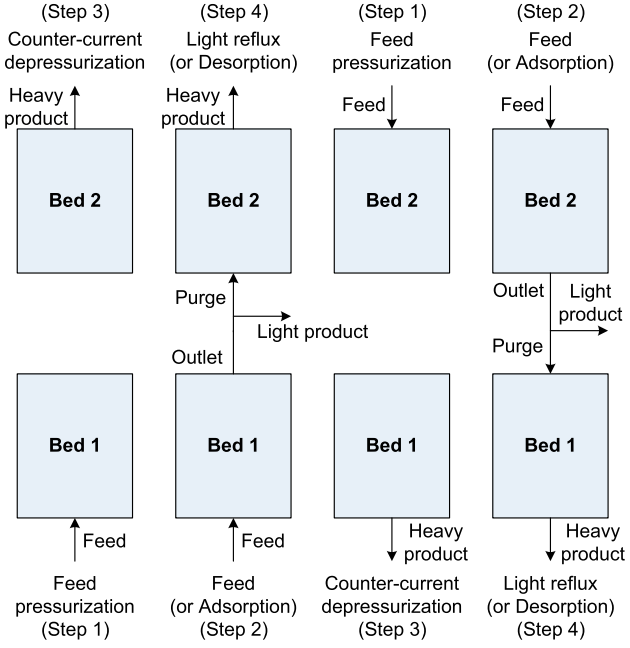


Figure 1.2: A two-bed four-step PSA cycle.

The mathematical model for the PSA process is presented in Table 1.1. The model assumes all gases to be ideal and radial variations in concentration are neglected in both gas and solid phases. In addition, the process is assumed to be isothermal with negligible pressure drop along the bed. Adsorption is modeled with the dual-site Langmuir isotherm and the linear driving force (LDF) expression. Zeolite 13X is chosen as the adsorbent. Model parameters can be found in [46]. The four steps of the process are enforced using boundary conditions on the feed flow rate and composition. This serves as the truth model for which exact gradients are available for construction of the reduced model.

Table 1.1: Model equations for isothermal PSA.

Component mass balance

$$\epsilon_b \frac{\partial y_i}{\partial t} + \frac{\partial (v y_i)}{\partial \xi} + \frac{RT}{P}(1-\epsilon_b)\rho_s \frac{\partial q_i}{\partial t} = 0 \quad i = \mathrm{CO_2} \tag{1.21}$$

Overall mass balance

$$\frac{\partial v}{\partial \xi} + \frac{RT}{P}(1-\epsilon_b)\rho_s \sum_i \frac{\partial q_i}{\partial t} = 0 \tag{1.22}$$

LDF equation

$$\frac{\partial q_i}{\partial t} = k_i (q_i^* - q_i) \quad i = 1, 2 \tag{1.23}$$

Dual-site Langmuir isotherm

$$q_i^* = \frac{q_{1i}^s b_{1i} y_i P}{1 + \sum_j b_{1j} y_j P} + \frac{q_{2i}^s b_{2i} y_i P}{1 + \sum_j b_{2j} y_j P} \quad i = \mathrm{CO_2}, \mathrm{N_2} \tag{1.24}$$

Cyclic steady state

$$z(t_0) = z(t_{\mathrm{cycle}}) \quad z : y_i, q_i \quad i = \mathrm{CO_2}, \mathrm{N_2} \tag{1.25}$$

k_i lumped mass transfer coefficient for i-th component (sec^{-1})

P total bed pressure (kPa)

q_i solid-phase concentration of i-th component ($\mathrm{gmol\,kg}^{-1}$)

q_i^* equilibrium solid-phase concentration of i-th component ($\mathrm{gmol\,kg}^{-1}$)

R universal gas constant ($\mathrm{J\,gmol}^{-1}\,\mathrm{K}^{-1}$)

T gas-phase temperature in the bed (K)

v gas superficial velocity ($\mathrm{m\,sec}^{-1}$)

y_i mole fraction of i-th component

Greek letters

ϵ_b bulk void fraction

ρ_s adsorbent density ($\mathrm{kg\,m}^{-3}$)

The model will be used in an optimization study to maximize CO_2 recovery subject to purity constraints. The key decision variables include the high pressure P_h at which the adsorption step takes place, the low pressure P_l for the depressurization and desorption steps, the duration t_p for pressurization and depressurization, the duration t_a for adsorption and desorption, and the feed velocity during adsorption u_a. These variables are aggregated into the vector $w = [P_h, P_l, t_p, t_a, u_a]$, which represents the inputs to the PSA model. Given particular values for w, the truth model can be simulated to cyclic steady state. Starting values for these variables are shown in Table 1.2. However, to make the optimization problem more tractable, the PSA truth model is replaced with reduced-order models.

Table 1.2: Initial guess for optimization problem (1.26).

Decision variable	Guessed value
Adsorption pressure (P_h)	150 kPa
Desorption pressure (P_l)	50 kPa
Pressurization step time (t_p)	50 sec
Adsorption step time (t_a)	150 sec
Adsorption feed flow (u_a)	20 cm/sec

Table 1.3: Comparison of truth model and RM based on the performance variables.

Performance variables	Truth model	RM
N_2 purity	92.57 %	92.51 %
N_2 recovery	80.21 %	80.71 %
CO_2 purity	37.76 %	38.29 %
CO_2 recovery	66.27 %	67.44 %

The RM for PSA is formed using POD. In fact, the reduced model is rebuilt several times during optimization as the decision variables change. We represent the sequence of decision variables as $\{w_k\}$, $k = 1, 2, \ldots$. Each time a POD model is built, the truth model is first simulated at the particular point w_k to gather a set of snapshots. To simulate the truth model, the spatial dimensions were discretized with 50 finite volumes and the resulting system of 1400 DAEs was simulated using MATLAB's *ode15s*. Then, the size of the POD basis is determined using a cutoff parameter $\lambda^* = 0.05$. The Galerkin projection is applied to obtain the RM, with equations shown in [2]. The RM has only 70 DAEs, which is a reduction of a factor of over 20 from the truth model. The RM equations are then discretized in time using Radau collocation on finite elements and the resulting equations are solved using IPOPT in AMPL. Table 1.3 compares the performance of the truth model and the RM for typical inputs. Further information on the model reduction, implementation of the TRF method, and its performance can be found in [2].

The RMs are used to maximize CO_2 recovery subject to a constraint on CO_2 purity. These variables are determined by a time average of the product stream compositions over the full-cycle time horizon. The optimization formulation is summarized as follows:

$$\begin{aligned} \max \quad & CO_2 \text{ recovery (time averaged)} \\ \text{s. t.} \quad & CO_2 \text{ purity} \geq 0.5 \text{ (time averaged)}, \\ & 101.32\,\text{kPa} \leq P_h \leq 300\,\text{kPa}, \\ & 40\,\text{kPa} \leq P_l \leq 101.32\,\text{kPa}, \\ & 35\,\text{sec} \leq t_p \leq 150\,\text{sec}, \\ & 50\,\text{sec} \leq t_a \leq 400\,\text{sec}, \\ & 10\,\text{cm/sec} \leq u_a \leq 30\,\text{cm/sec}, \\ & \text{PDAEs for PSA.} \end{aligned} \tag{1.26}$$

The bound on CO_2 purity is set at 50 %, which is determined by what is realistically attainable with a two-bed four-step cycle. Also, note that the pressure P_l is allowed to lie in the vacuum range, which aids desorption. To improve computational performance, the PDAE system in (1.26) is replaced with a sequence of RMs. The RMs are used in a trust region method to find the optimum of problem (1.26) considering the truth model.

The predicted CO_2 purity and recovery from the RM will often be different than those from the truth model. To ensure consistency and ultimately drive convergence to the truth model optimum, correction terms are added to the RM, as proposed by [3, 43]. The trust region method for the PSA problem used both zero- and first-order additive corrections (ZOC and FOC). The zero-order correction forms $r_k(w)$ that agrees with the truth model at the trust region center w_k:

$$r_k^{\text{zoc}}(w) := r_{\text{POD}}(w) + t(w_k) - r_{\text{POD}}(w_k), \tag{1.27}$$

so that the zero-order corrected outputs used for optimization are the same as predicted by the RM with an additive correction factor.

The first-order additive correction from (1.19) is given by:

$$r_k^{\text{foc}}(w) := r_{\text{POD}}(w) + \left(t(w_k) - r_{\text{POD}}(w_k)\right) + \left(\nabla t(w_k) - \nabla r_{\text{POD}}(w_k)\right)^T (w - w^k), \tag{1.28}$$

so that the corrected outputs r_k^{foc} used for optimization agree with the truth model in function values and gradients at the design point w_k. In addition to the simple Taylor expansion, the underlying POD RM provides a physics-based functional form to model how the outputs change further away from the design point w_k.

The zero- and first-order corrected RMs are used in a two-stage algorithm. The analysis in [2] shows that the trust region filter approach with first-order consistent RMs will converge to the optimum of (1.26). The POD models plus first-order correction satisfy this first-order consistency condition, but are relatively expensive to build because of the need to estimate gradient information. Therefore, the approach used to solve this problem will first attempt optimization using the zero-order correction

models. The zero-order correction comes at no additional cost since the point w_k is already evaluated with the truth model to gather snapshots to build POD models. The TRF approach is applied with zero-order correction models until no more progress is made towards the optimum. Then, the algorithm is reinitialized and first-order corrected models are used to guarantee convergence to a stationary point. The sensitivity information for first-order corrections is obtained using finite difference perturbations of the inputs w and simulating the truth model.

The two-phase TRF algorithm successfully solves problem (1.26) with 51 iterations, of which the first 35 use the zero-order correction. At each iteration, the discretized RM-based optimization problem is a nonlinear program with 52,247 variables. The CO_2 purity and recovery are plotted over the course of the iterations, as shown in Figure 1.3. As seen in (24) CO_2 purity is constrained to be above 50 %, and this constraint is clearly active at most steps of the algorithm. However, maximization of CO_2 recovery is more difficult for the algorithm. Progress is slow, but after the zero-order correction approach terminates the solution is still suboptimal. In the decision variable space, this final increase in recovery in the first-order correction phase is largely achieved by moving the pressurization step time from near its upper bound to being active at its lower bound. This indicates that the POD-based RMs were not accurately capturing derivative information, which led to suboptimal results. The final termination point was further validated to confirm that it is a local maximum.

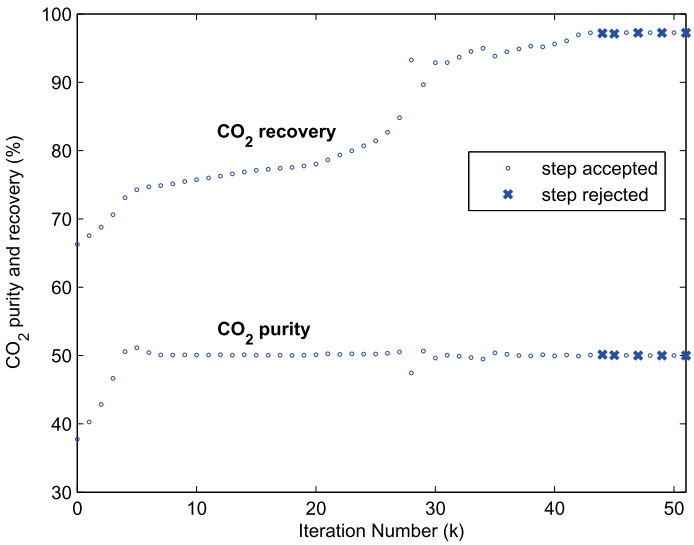


Figure 1.3: CO_2 purity and recovery for all the iterations of the filter-based algorithm.

The full results of this case study may be found in [2]. That work also compares the filter-based approach to a more conventional exact penalty approach. The benefit of

the filter mechanism is clearly shown by its flexibility to accept larger steps. In contrast to the TRF approach, the exact penalty method takes 92 iterations, each using the first-order correction. The filter's flexibility to accept steps that may reduce feasibility give it an advantage of faster convergence.

The PSA case study demonstrates the limitations of RM-based optimization while also demonstrating a solution in the form of first-order corrected reduced models. However, the expense of obtaining derivative estimates to build the first-order correction counteracts the benefits gained from the model reduction. As a result, the two-stage approach applies first-order corrections as necessary to ensure convergence of RM-based optimization.

1.4.2 Advanced power plant optimization

The second case study is also an application of CO_2 capture. Unlike the previous study, we now deal with an entire flowsheet with both glass-box and black-box models. While first and second derivatives are calculated cheaply from glass-box models, gradient information is not available from the black-box truth models. These need to be approximated by equation-oriented reduced models.

This case study deals with the optimization of the oxycombustion process, one of several approaches for carbon capture from power generation processes. In oxycombustion, a mixture of purified oxygen and recycle carbon dioxide is used to combust fuel in a boiler. This results in a flue gas that is primarily water and CO_2. The water is easily separated and the task of purifying the CO_2 is greatly simplified because most nitrogen was removed before combustion. Compared to conventional air-fired power plants, the oxycombustion process has several key differences. The introduction of new separation tasks, namely, air separation and CO_2 compression/purification, lower the overall power output of the plant due to the energy required to run them. In addition, the recycle loop further couples the temperature and composition of pre- and postcombustion gas streams. Finally, the oxy-fired boiler behaves differently than air-fired boilers, and it is necessary to consider the effect of these changes on the rest of the power plant design. To rigorously manage the interactions between these subsystems and reduce the cost of carbon capture, a comprehensive optimization approach is proposed.

In order to optimize the oxycombustion process, the first step is to consider the level of model fidelity required. For process optimization, steady-state models are usually sufficient. At the process level, heat exchanger, pumps, compressors, and turbines are modeled as AEs. A framework for building these models as well as application to the oxycombustion separation systems was presented in [33, 32]. However, modeling the boiler presents special challenges. A full-scale CFD simulation including reactions and transport behavior in three dimensions can take several weeks to solve. Instead, we use a hybrid one-/three-dimensional zonal boiler model as described in [54] as the truth model. This boiler model uses a series of nine vertical zones to model reactions

and particle/gas flow, while integrating the radiation PDE in three dimensions with the discrete ordinate method on 21,888 finite elements. The resulting truth model can converge in about 1 CPU minute on a desktop computer. The model is custom built in C++ with specialized methods to help guarantee robustness when converging the simulation. As discussed in Section 1.2, the requirement for specialized simulation methods for the truth model suggests the use of data-driven model reduction. The hybrid one-/three-dimensional boiler model is approximated with RMs and used to solve a plant-wide optimization problem with the TRF method.

For the purposes of the optimization problem, the boiler model can be viewed as a function from $\mathbb{R}^m \rightarrow \mathbb{R}^p$, where the input and output variables are chosen to capture the interactions of the boiler with the rest of the power plant. These inputs and output variables are given below (corresponding to the w and y variables in the glass-box/black-box formulation (1.17)):

Boiler inputs:

1. primary air temperature,
2. secondary/over-fired air temperature,
3. average temperature of boiling water inside water wall tubes,
4. average secondary superheater steam temperature,
5. primary air component flow rates (O_2, N_2, H_2, CO, CO_2, H_2O, SO_2, H_2S, CH_4, Ar),
6. secondary air component flow rates (same components as primary air, but different compositions),
7. over-fired air total flow rate (same composition as secondary air).

Boiler outputs:

1. boiler enclosure water wall heat duty,
2. secondary superheater heat duty,
3. flue gas component flow rates (same components as primary air),
4. flue gas temperature.

The primary air is the gas stream into the boiler that carries the pulverized coal particles, whereas secondary and over-fired air streams are added into the boiler directly to aid in combustion. The temperature and composition of these streams have a strong impact on the combustion behavior in the boiler. In general, higher heat transfer through the boiler enclosure wall improves performance, although the flue gas temperature must remain bounded for material considerations. The compositions flowing into the boiler are indirectly coupled with the composition leaving the boiler through recycle, and the heat transfer behavior is also indirectly coupled with the average temperature of water in the water wall tubes through the steam cycle. The use of a rigorous boiler model helps capture these interactions accurately.

Figure 1.4 shows the general configuration of the steam and gas sides of the steam cycle. On the steam side, high-pressure water enters at the bottom of the boiler and steam exits the boiler after the secondary superheater. The turbines are divided into

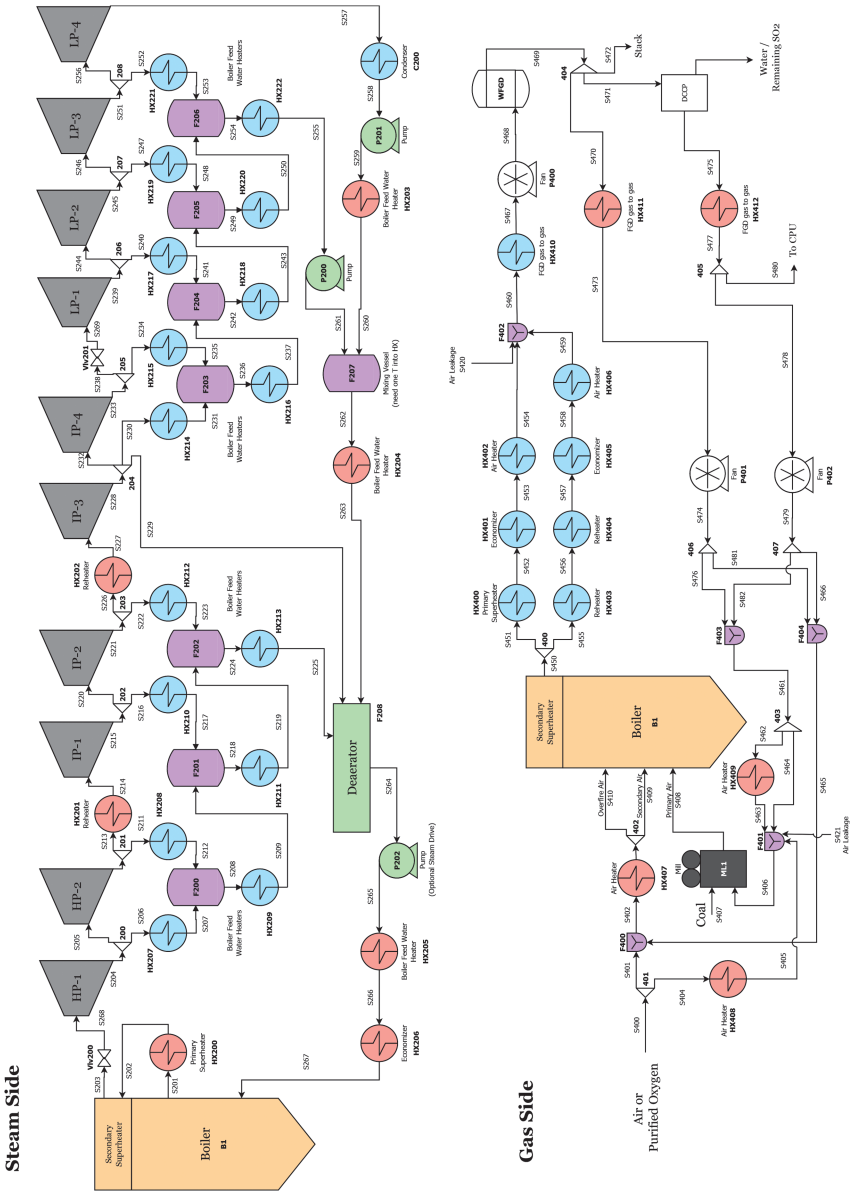


Figure 1.4: The water/steam and gas sides of the steam cycle. Heat exchangers are modeled as halves, where red units receive heat and blue units provide heat.

high-, intermediate-, and low-pressure sections (HP, IP, and LP, respectively). A superstructure of potential steam extraction sites between stage groups is considered as part of the optimization. The extracted steam may be sent to the boiler feedwater heaters. Heat exchangers are modeled with heat exchanger halves, specified either as a heater

(red) or cooler (blue). The heating and cooling duties are matched for these heat exchanger halves in order to form heat exchangers in the power plant, including the primary superheater, reheaters, and the economizer. The remaining heat exchanger halves are matched through the formulation of a pinch-based heat integration model. This heat integration model is developed using the Duran–Grossmann formulation [34]; see [33, 74] for details on the implementation in this equation-oriented flowsheet framework. On the gas side, the purified oxygen stream is split before being mixed with recycled flue gas. The split allows primary and secondary air streams to have different compositions. After combustion, the flue gas is split to two series of heat exchangers, and then sent to pollution controls. The direct contact cooler/polishing scrubber (DCCP) is used to remove much of the water from the flue gas. The ratio of flue gas sent for water cooling is also a key decision variable due to the role of water vapor in the radiation behavior in the boiler. Then, a fraction of the flue gas is sent for compression and purification while the rest is recycled to the boiler. Primary air flow rate is bounded below to ensure at least a 2:1 gas-to-coal ratio to ensure that the gas can carry the coal. For safety reasons, the primary air stream also has upper bounds on temperature and O_2 mole fraction, of 400 K and 35 %, respectively.

The TRF algorithm was used to maximize the thermal efficiency of a double reheat oxy-fired steam cycle. The optimization problem is summarized in (1.29). Both design and operational decisions could be modified for the boiler, but we consider the case where boiler geometry and burner configuration match an existing utility boiler (Pacificorp's Hunter 3 unit).

The objective of this study to maximize thermal efficiency with a fixed coal feed flow rate, with a small penalty for utility consumption:

$$
\begin{aligned}
\max \quad & \text{thermal efficiency} + \rho_w Q_w \\
\text{s. t.} \quad & \text{thermal efficiency} = \frac{\sum W_{\text{turbine}} - \sum W_{\text{pump}} - \sum W_{\text{fan}} - W_{\text{CPU}} - W_{\text{ASU}}}{\text{thermal input rate}}, \\
& \text{fixed thermal (fuel) input rate,} \\
& \text{steam turbine model,} \\
& \text{pump and fan models,} \\
& \text{Duran–Grossmann pinch-location heat integration equations,} \\
& \text{correlation-based fuel gas thermodynamics model,} \\
& \text{steam thermodynamics,} \\
& \text{hybrid boiler model,} \\
& \text{correlation model ASU,} \\
& \text{correlation model CPU,}
\end{aligned}
\tag{1.29}
$$

where ρ_w is a small penalty term for cooling water usage. The air separation unit (ASU) and carbon dioxide processing unit (CPU) are modeled with correlations derived from [33, 30, 31].

The truth boiler model is replaced with a sequence of reduced models $r_k(w)$. In contrast to the PSA study, this case study uses an improved version of the TRF algorithm that does not require first-order consistency for convergence. Instead, the RMs must satisfy the κ-fully linear property (1.20). Common choices such as interpolation or regression apply under mild assumptions (e. g., uniformly bounded second derivatives of $r_k(w)$). This condition provides great flexibility for a wide variety of reduced modeling approaches. For the boiler model, simple polynomial interpolation models constructed with well-poised sample sets provided good performance. These RMs are updated automatically, with recourse to the truth models, by the TRF method.

Optimization problem (1.29) was solved for two different scenarios. In Case A, the oxygen purity supplied by the ASU was fixed at 97 mol%, which means that the power requirement of the ASU is only dependent on the flow rate of oxygen supplied. In Case B, the oxygen purity was allowed to vary between 90 mol% and 98 mol%. This allows the optimizer to trade off the pre- and postcombustion separation tasks, while simultaneously considering the interactions with the detailed kinetics and radiation behavior in the boiler. In the lower oxygen environment, gasification reactions are more favored and the emissivity of the gas mixture changes. The optimization results for both cases are given in Table 1.4. In Case A, the optimum design found by the TRF algorithm is an oxy-fired power plant with a net power output of 437.4 MW and a net efficiency of 33.0 %. In Case B, the optimum solution has a net power output of 440.4 MW and an efficiency of 33.2 %.[1] Interestingly, in Case B the oxygen composition is pushed to its lower bound of 90 mol%. In both scenarios the optimizer pushes the steam temperatures leaving the secondary superheater and reheaters to their upper bounds of 835 K and 867 K, respectively, as expected when maximizing thermal efficiency. Similarly, the lower bound of 0.068 bar for the condenser operating pressures is active. The optimizer also pushes the temperature and oxygen content of the primary flue gas recycle streams (S408 in Figure 1.4) to their upper bounds of 400 K and 35 mol%. Another interesting conclusion lies in the recycle distribution of the flue gas. The temperature and composition of the flue gas (influenced by drying) has complex interactions with the detailed boiler model. By optimizing using the reduced boiler model, these interactions can be considered and additional efficiencies are identified. Additional information on the implementation and performance of the TRF algorithm for the oxycombustion case study can be found in [37, 35, 54].

1.5 Conclusions

Reduced models have been used in many domains in chemical engineering to provide computationally tractable simulation and optimization. Most model reduction tech-

1 Higher heating value basis.

Table 1.4: Power plant optimization results. Case A: fixed oxygen purity. Case B: variable oxygen purity.

	Case A	Case B
Work from turbines (MW)	568.2	570.3
HP	94.5	94.5
IP	267.3	267.7
LP	206.4	208.1
Pumping work (MW)	12.4	12.4
Fan work (MW)	3.6	3.7
Heat from boiler (MW)	520.6	531.3
Boiler walls	446.4	457.3
Secondary superheater	74.2	74
Heat from flge Gas (MW)	659.2	653.0
Primary superheater	201.0	220.5
Reheater (HX201)	168.6	168.7
Reheater (HX202)	146.6	148.4
Economizer	143.0	115.4
Heat rejected (MW)	620.9	623.3
Fuel heat rate (MW)	1325.5	1325.5
ASU power (MW)	71.2	65.6
CPU power (MW)	43.6	48.2
Net power (MW)	437.4	440.4
Thermal efficiency (HHV)	33.0 %	33.2 %
Flue gas recycle distribution		
Bypasses DCCP, to secondary rec.	29.7 %	31.3 %
To CPU after DCCP	28.0 %	29.9 %
To primary recycle after DCCP	23.6 %	25.3 %
To secondary recycle after DCCP	18.7 %	13.5 %

niques can be categorized as model-based or data-driven. Model-based methods can exploit known information about the structure of the truth model to build reduced models, whereas data-driven methods are often suitable when specialized software is used for the truth model.

When reduced models are used in optimization, the accuracy becomes an even greater concern. Optimal solutions are characterized by gradient information, which adds more demands on the reduced model accuracy. Trust region methods provide a systematic approach to manage the accuracy of reduced models in optimization. Through adaptive updating of reduced models, convergence can be guaranteed to solutions of the truth model-based problem. However, accuracy conditions must be enforced on the reduced models. While first-order consistency is straightforward to enforce, this may be computationally demanding in practice. Instead, the framework of κ-fully linear models provides more flexibility.

Our optimization strategy is agnostic to the type of reduced model as long as the κ-fully linear property holds. The verification that an RM is κ-fully linear is easy when a data-driven RM is used. On the other hand, if truth model gradients are not available, it is not clear that the κ-fully linear framework can be used for model-based reductions. Thus, while model reduction is already used in many domains of chemical engineering, challenges remain in building (certifiably) accurate and efficient RMs.

Future work will develop more efficient TRF methods. Recent advances of the TRF algorithm includes the addition of a sampling region, which ensures the accuracy of the RM, while the surrounding trust region globalizes the TRF algorithm, and need not shrink to zero upon convergence. As developed, analyzed, and demonstrated in [38], this TRF enhancement led to a 40 % decrease in computational effort. Further research will also be devoted to tailored algorithms that exploit the solution of multiple RMs within the process flowsheet as well as specialized decompositions for truth models.

Bibliography

[1] A. Agarwal, L. T. Biegler, and S. E. Zitney, Simulation and optimization of pressure swing adsorption systems using reduced-order modeling. *Ind. Eng. Chem. Res.*, **48** (5) (2009), 2327–2343.

[2] Anshul Agarwal and Lorenz T. Biegler, A trust-region framework for constrained optimization using reduced order modeling, *Optim. Eng.*, **14** (1) (2013), 3–35.

[3] Natalia M. Alexandrov, John E. Dennis Jr, Robert Michael Lewis, and Virginia Torczon, A trust-region framework for managing the use of approximation models in optimization, *Struct. Optim.*, **15** (1) (1998), 16–23.

[4] Eyal Arian, Marco Fahl, and Ekkehard W. Sachs, Trust-region proper orthogonal decomposition for flow control. Technical Report ICASE Report No. 2000-25, Institute for Computer Applications in Science and Engineering, 2000.

[5] Antonios Armaou and Panagiotis D. Christofides, Dynamic optimization of dissipative pde systems using nonlinear order reduction, *Chem. Eng. Sci.*, **57** (24) (2002), 5083–5114.

[6] Eva Balsa-Canto, Antonio A. Alonso, and Julio R. Banga, A novel, efficient and reliable method for thermal process design and optimization. Part I: theory, *J. Food Eng.*, **52** (3) (2002), 227–234.

[7] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci.*, I (2004), 339–667.

[8] A. Barrett and J. J. Walsh, Improved chemical process simulation using local thermodynamic approximations, *Comput. Chem. Eng.*, **3** (1-–4) (1979), 397–402.

[9] U. Baur, P. Benner, and L. Feng, Model order reduction for linear and nonlinear systems: a system-theoretic perspective, *Arch. Comput. Methods Eng.*, **21** (2014), 331–358.

[10] Eugene Bendersky and Panagiotis D. Christofides, Optimization of transport-reaction processes using nonlinear model reduction, *Chem. Eng. Sci.*, **55** (19) (2000), 4349–4366.

[11] Peter Benner, Serkan Gugercin, and Karen Willcox, A survey of model reduction methods for parametric systems, *SIAM Rev.*, **57** (4) (2015), 483–531.

[12] Michel Bergmann, Laurent Cordier, and Jean-Pierre Brancher, Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model, *Phys. Fluids*, **17** (9) (2005), 097101.
[13] B. Beykal, F. Boukouvala, C. A. Floudas, and E. N. Pistikopoulos, Optimal design of energy systems using constrained grey-box multi-objective optimization, *Comput. Chem. Eng.*, **116** (2018), 488–502.
[14] L. T. Biegler and I. E. Grossmann, Part I: Retrospective on optimization, *Comput. Chem. Eng.*, **28** (8) (2004), 1169–1192.
[15] Lorenz T. Biegler, Ignacio E. Grossmann, and Arthur W. Westerberg, A note on approximation techniques used for process optimization, *Comput. Chem. Eng.*, **9** (2) (1985), 201–206.
[16] Lorenz T. Biegler, Yidong Lang, and Weijie Lin, Multi-scale optimization for process systems engineering, *Comput. Chem. Eng.*, **60** (2014), 17–30.
[17] J. F. Boston and H. I. Britt, A radically different formulation and solution of the single-stage flash problem, *Comput. Chem. Eng.*, **2** (2-3) (1978), 109–122.
[18] F. Boukouvala and C. A. Floudas, Argonaut: algorithms for global optimization of constrained grey-box computational problems, *Optim. Lett.*, **11** (5) (2017), 895–913.
[19] Fani Boukouvala, M. M. Faruque Hasan, and Christodoulos A. Floudas, Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption, *J. Glob. Optim.*, (2015), 1–40.
[20] J. Bremer, P. Goyal, L. Feng, P. Benner, and K. Sundmacher, POD-DEIM for efficient reduction of a dynamic 2d catalytic reactor model, *Comput. Chem. Eng.*, **106** (2017), 777–784.
[21] Thomas A. Brenner, Raymond L. Fontenot, Paul G. A. Cizmas, Thomas J. O'Brien, and Ronald W. Breault, A reduced-order model for heat transfer in multiphase flow and practical aspects of the proper orthogonal decomposition, *Comput. Chem. Eng.*, **43** (2012), 68–80.
[22] Tan Bui-Thanh, Murali Damodaran, and Karen E. Willcox, Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, *AIAA J.*, **42** (8) (2004), 1505–1516.
[23] José A. Caballero and Ignacio E. Grossmann, An algorithm for the use of surrogate models in modular flowsheet optimization, *AIChE J.*, **54** (10) (2008), 2633–2650.
[24] Yanhua Cao, Jiang Zhu, Zhendong Luo, and I. M. Navon, Reduced-order modeling of the upper tropical Pacific Ocean model using proper orthogonal decomposition, *Comput. Math. Appl.*, **52** (8-9) (2006), 1373–1386.
[25] S. Chaturantabut and D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (2010), 2737–2764.
[26] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint, *Trust region methods*, SIAM, 2000.
[27] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia, 2009.
[28] M. Couplet, C. Basdevant, and P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, *J. Comput. Phys.*, **207** (1) (2005), 192–220.
[29] Alison Cozad, Nikolaos V. Sahinidis, and David C. Miller, Learning surrogate models for simulation-based optimization, *AIChE J.*, **60** (6) (2014), 2211–2227.
[30] A. W. Dowling, J. P. Eason, J. Ma, D. C. Miller, and L. T. Biegler, Coal oxycombustion power plant optimization using first principles and surrogate boiler models, *Energy Proc.*, **63** (2014), 352–361.
[31] A. W. Dowling, J. P. Eason, J. Ma, D. C. Miller, and L. T. Biegler, Equation-based design, integration, and optimization of oxycombustion power systems, in M. Martin (ed.), *Alternative Energy Sources and Technologies*, pp. 119–158, Springer, Switzerland, 2016.
[32] Alexander W. Dowling, Cheshta Balwani, Qianwen Gao, and Lorenz T. Biegler, Optimization of sub-ambient separation systems with embedded cubic equation of state thermodynamic models and complementarity constraints, *Comput. Chem. Eng.*, **81** (2015), 323–343.

[33] Alexander W. Dowling and Lorenz T. Biegler, A framework for efficient large scale equation-oriented flowsheet optimization, *Comput. Chem. Eng.*, **72** (2015), 3–20.
[34] Marco A. Duran and Ignacio E. Grossmann, Simultaneous optimization and heat integration of chemical processes, *AIChE J.*, **32** (1) (1986), 123–138.
[35] J. P. Eason and L. T. Biegler, Reduced model trust region methods for embedding complex simulations in optimization problems, in K. Gernaey, J. Huusom, and R. Gani (eds.), *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering, Computer Aided Chemical Engineering*, vol. 37, pp. 773–778, Elsevier, Oxford, UK, 2015.
[36] John Eason and Selen Cremaschi, Adaptive sequential sampling for surrogate model generation with artificial neural networks, *Comput. Chem. Eng.* (2014).
[37] John P. Eason and Lorenz T. Biegler, A trust region filter method for glass box/black box optimization, *AIChE J.*, **62** (9) (2016), 3124–3136.
[38] J. P. Eason and L. T. Biegler, Advanced trust region optimization strategies for glass box / black box models, *AIChE J.*, **64** (11) (2018), 3934–3943.
[39] M. Fahl and E. W. Sachs, Reduced-order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition, in *Large-Scale PDE-Constrained Optimization*, pp. 268–280, Springer Verlag, Heidelberg, Germany, 2003.
[40] Roger Fletcher, Nicholas I. M. Gould, Sven Leyffer, Philippe L. Toint, and Andreas Wächter, Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming, *SIAM J. Optim.*, **13** (3) (2002), 635–659.
[41] Roger Fletcher and Sven Leyffer, Nonlinear programming without a penalty function, *Math. Program.*, **91** (2) (2002), 239–269.
[42] B. Galletti, C. H. Bruneau, Luca Zannetti, and Angelo Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, *J. Fluid Mech.*, **503** (2004), 161–170.
[43] Anthony A. Giunta and Michael S. Eldred, Implementation of a trust region model management strategy in the Dakota optimization toolkit, in *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA*, 2000.
[44] Max D. Gunzburger, Janet S. Peterson, and John N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, *Comput. Methods Appl. Mech. Eng.*, **196** (4–6) (2007), 1030–1047.
[45] Jack P. C. Kleijnen, Kriging metamodeling in simulation: a review, *Eur. J. Oper. Res.*, **192** (3) (2009), 707–716.
[46] Daeho Ko, Ranjani Siriwardane, and Lorenz T. Biegler, Optimization of pressure swing adsorption and fractionated vacuum pressure swing adsorption processes for CO2 capture, *Ind. Eng. Chem. Res.*, **44** (21) (2005), 8084–8094.
[47] Karl Kunisch and Stefan Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM J. Numer. Anal.*, **40** (2) (2002), 492–515.
[48] Yi-dong Lang, Adam Malacina, Lorenz T. Biegler, Sorin Munteanu, Jens I. Madsen, and Stephen E. Zitney, Reduced order model based on principal component analysis for process simulation and optimization, *Energy Fuels*, **23** (3) (2009), 1695–1706.
[49] Yidong Lang, Stephen E. Zitney, and Lorenz T. Biegler, Optimization of IGCC processes with reduced order CFD models, *Comput. Chem. Eng.*, **35** (9) (2011), 1705–1717.
[50] M. E. Leesley and G. Heyen, The dynamic approximation method of handling vapor-liquid equilibrium data in computer calculations for chemical processes, *Comput. Chem. Eng.*, **1** (2) (1977), 103–108.
[51] S. Li, L. Feng, P. Benner, and A. Seidel-Morgenstern, Using surrogate models for efficient optimization of simulated moving bed chromatography, *Comput. Chem. Eng.*, **67** (2014), 121–132.

[52] S. Li, Y. Yue, L. Feng, P. Benner, and A. Seidel-Morgenstern, Model reduction for linear simulated moving bed chromatography systems using Krylov-subspace methods, *AIChE J.*, **60** (11) (2014), 3773–3783.
[53] Hung V. Ly and Hien T. Tran, Modeling and control of physical processes using proper orthogonal decomposition, *Math. Comput. Model.*, **33** (1–3) (2001), 223–236.
[54] Jinliang Ma, John P. Eason, Alexander W. Dowling, Lorenz T. Biegler, and David C. Miller, Development of a first-principles hybrid boiler model for oxy-combustion power generation system, *Int. J. Greenh. Gas Control*, **46** (2016), 136–157.
[55] Andrew March and Karen Willcox, Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives, *AIAA J.*, **50** (5) (2012), 1079–1089.
[56] W. Marquardt, Nonlinear model reduction for optimization based control of transient chemical processes, in J. B. Rawlings, B. A. Ogunnaike, and J. W. Eaton (eds.), *Chemical Process Control VI*, vol. 326, pp. 12–42, 2002.
[57] Jeff Moehlis, T. R. Smith, Philip Holmes, and H. Faisst, Models for turbulent plane Couette flow using the proper orthogonal decomposition, *Phys. Fluids*, **14** (7) (2002), 2493–2507.
[58] K. Murphy, *Machine Learning*, MIT Press, Cambridge, MA, 2012.
[59] Raymond H. Myers, Douglas C. Montgomery, G. Geoffrey Vining, Connie M. Borror, and Scott M. Kowalski, Response surface methodology: a retrospective and literature survey, *J. Qual. Technol.*, **36** (1) (2004), 53.
[60] H. M. Park and D. H. Cho, The use of the Karhunen–Loeve decomposition for the modeling of distributed parameter systems, *Chem. Eng. Sci.*, **51** (1) (1996), 81–98.
[61] B. Peherstorfer, K. Willcox, and M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, *SIAM Rev.*, **60** (3) (2018), 550–591.
[62] S. Y. Shvartsman, C. Theodoropoulos, R. Rico-Martınez, I. G. Kevrekidis, E. S. Titi, and T. J. Mountziaris, Order reduction for nonlinear dynamic models of distributed reacting systems, *J. Process Control*, **10** (2-3) (2000), 177–184.
[63] Timothy W. Simpson, J. D. Poplinski, Patrick N. Koch, and Janet K. Allen, Metamodels for computer-based engineering design: survey and recommendations, *Eng. Comput.*, **17** (2) (2001), 129–150.
[64] Lawrence Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, *Q. Appl. Math.*, **45** (3) (1987), 561–571.
[65] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, Projection-based model reduction: formulations for physics-based machine learning. *Comput. Fluids*, 2018, 10.1016/j.compfluid.2018.07.021.
[66] Artemis Theodoropoulou, Raymond A. Adomaitis, and Evanghelos Zafiriou, Model reduction for optimization of rapid thermal chemical vapor deposition systems, *IEEE Trans. Semicond. Manuf.*, **11** (1) (1998), 85–98.
[67] Stefan Volkwein and Michael Hinze, Model order reduction by proper orthogonal decomposition, in *Handbook on Model Order Reduction*, DeGruyter, 2019.
[68] Andrea Walther and Lorenz Biegler, On an inexact trust-region SQP-filter method for constrained nonlinear optimization, *Comput. Optim. Appl.*, (2014), 1–26.
[69] Stefan M. Wild, *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*. PhD thesis, Cornell University, 2009.
[70] Stefan M. Wild, Rommel G. Regis, and Christine A. Shoemaker, ORBIT: optimization by radial basis function interpolation in trust-regions, *SIAM J. Sci. Comput.*, **30** (6) (2008), 3197–3219.
[71] Karen Willcox and Jaime Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA J.*, **40** (11) (2002), 2323–2330.
[72] Zachary T. Wilson and Nikolaos V. Sahinidis, The alamo approach to machine learning, *Comput. Chem. Eng.*, **106** (2017), 785–795.

[73] T. Yuan, P. G. Cizmas, and T. O'Brien, A reduced-order model for a bubbling fluidized bed based on proper orthogonal decomposition, *Comput. Chem. Eng.*, **30** (2) (2005), 243–259.
[74] Dehao Zhu, J. P. Eason, and L. T. Biegler, Energy-efficient CO2 liquefaction for oxy-combustion power plant with ASU-CPU integration enhanced by cascaded cryogenic energy utilization and waste heat recovery, *Int. J. Greenh. Gas Control*, **61** (2017), 124–137.

B. Lohmann, T. Bechtold, P. Eberhard, J. Fehr, D. J. Rixen,
M. Cruz Varona, C. Lerch, C. D. Yuan, E. B. Rudnyi, B. Fröhlich,
P. Holzwarth, D. Grunert, C. H. Meyer, and J. B. Rutzmoser

2 Model order reduction in mechanical engineering

Abstract: This chapter describes several "success stories" of model order reduction (MOR) in mechanical engineering. First, the reader will be given an overview of specific model representations, MOR requirements, and reduction techniques relevant in the different fields of mechanical engineering. Then, four applications are presented: the reduction of a thermo-mechanical machining tool, the reduction of models of a car body and driver's seat, the reduction of an elastic crankshaft, and the reduction of a leaf spring model.

Keywords: Model order reduction, mechanical engineering, second-order systems, projection

MSC 2010: 65C05, 62M20, 93E11, 62F15, 86A22

2.1 Introduction

In the past 50 years, the number of applications of model reduction in mechanical engineering has increased massively, while the first applications in structural mechanics and acoustics ([94, 55, 56, 24], see also [18]) go back even further in time. *Modal methods* can be applied either to second-order models (common in multibody systems [48, 24, 1, 33, 84]) or to state-space representations. Corresponding methods were intensively developed from the 1960s onwards ([25, 73, 23, 20] and Chapter 4 of Volume 1 of *Model order reduction*) and are applied in other areas of engineering as well, for instance in the reduction of power systems [64] and for the purpose of control design, e. g., [71, 66, 58, 7]. With the advent of *balanced truncation* and of *Krylov subspace methods* ([78, 44], [42, 45], overviews in [4, 8, 10], and Chapters 2 and 3 of Volume 1 of *Model order reduction*) the approximation quality and the applicability to high- and very high-order linear systems improved significantly and opened numerous fields of applications. *Proper orthogonal decomposition* (POD) methods based on snapshots

B. Lohmann, D. J. Rixen, M. Cruz Varona, C. Lerch, C. H. Meyer, J. B. Rutzmoser, Technical University of Munich, Munich, Germany, e-mails: lohmann@tum.de, rixen@tum.de
T. Bechtold, C. D. Yuan, E. B. Rudnyi, University of Rostock, Rostock, Germany, e-mail: tamara.bechtold@uni-rostock.de
P. Eberhard, J. Fehr, B. Fröhlich, P. Holzwarth, D. Grunert, University of Stuttgart, Stuttgart, Germany, e-mails: peter.eberhard@itm.uni-stuttgart.de, joerg.fehr@itm.uni-stuttgart.de

https://doi.org/10.1515/9783110499001-002

[63, 70, 60] and *hyper-reduction* techniques [22, 30, 31] were developed for the reduction of *nonlinear* models.

Today, the successful applications of model order reduction (MOR) in mechanical engineering deal with different system and problem classes from different physical domains, like:

- *structural and multibody dynamics*, modeled by linear or nonlinear differential equations [18, 17, 21, 67, 69, 35, 34, 53, 3, 31, 91, 90, 5, 104, 19, 49, 47, 95, 110];
- *fluid dynamics*, including fluid–structure interaction and aerodynamics [93, 26, 89, 88, 83, 2, 31];
- *thermo-mechanical, thermo-fluid, thermo-acoustic, and thermo-electrical systems* [68, 12, 14, 11, 97, 75, 54, 46].

An overview on *coupled* problems is [15]. It should be emphasized that this list of references is incomplete. In view of the huge number of publications on MOR applications, it seems rather impossible to give a complete overview. Besides the many scientific publications, there exist several collections of benchmark problems, like the "Oberwolfach Benchmark Collection," the "Niconet Benchmark Collection," and the "MOR-Wiki." These collections include system models from different domains and can serve for the validation process.

In Sections 2.2 to 2.5 of this chapter, four successful applications are presented as examples of MOR of industrial problems:

- The reduction of a coupled thermo-mechanical machining tool model applies Krylov subspace-based reduction to a first- or second-order formulation of the linear model.
- The reduction of industrial models of a car body and driver's seat applies a Craig–Bampton method and a Gramian-based method to second-order models of very high order.
- The error-controlled reduction of an elastic crankshaft applies a combination of methods to a second-order model.
- The reduction of a leaf spring model applies simulation-free projection and hyper-reduction to a nonlinear second-order model.

In the following, an overview of common representations for mechanical systems, modeling aspects, and basic properties will be given.

Typically, mechanical models originate either from direct discrete modeling, spatial discretization of partial differential equations, system identification, or a combination of those.

Direct discrete modeling uses discrete, linear, or nonlinear mass, damper, and spring elements to build up a model. This process normally leads to small- or medium-sized models.

Most models originate from spatial discretization of partial differential equations describing the laws of physics. For solids, a continuum mechanics approach – equi-

librium of forces, kinematic equations and the constitutive equations – describe the physics. Well-established methods like the finite element method (FEM), the finite difference method, the finite volume method, the boundary element method, and others are applied for spatial discretization. All procedures are based on the method of weighted residuals and differ only in the specific choices of test and weighting functions. The classical displacement-based FEM, for example, uses first- or higher-order polynomial test and weighting functions from identical function spaces, i. e., a Ritz–Galerkin projection from the continuous to the discrete space is performed. This typically leads to very large models comprising thousands up to millions of degrees of freedom.

Another way of modeling mechanical systems is a data-driven identification of their dynamics. This typically leads to rather small models.

All models have in common that the equilibrium of inertia forces, $\boldsymbol{M}\ddot{\boldsymbol{q}}(t)$, damping and internal restoring forces, $\widehat{\boldsymbol{f}}(\dot{\boldsymbol{q}}(t), \boldsymbol{q}(t))$, and external forces, $\widehat{\boldsymbol{F}}(t)$, determines the basic dynamics of mechanical systems:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \widehat{\boldsymbol{f}}\big(\dot{\boldsymbol{q}}(t), \boldsymbol{q}(t)\big) = \widehat{\boldsymbol{F}}(t), \quad \boldsymbol{q}(0) = \boldsymbol{q}_0,\ \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0, \tag{2.1}$$

with (generalized) displacements $\boldsymbol{q}(t) \in \mathbb{R}^N$, mass matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$, nonlinear damping and internal forces $\widehat{\boldsymbol{f}} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$, and external forces or loadings $\widehat{\boldsymbol{F}}(t) \in \mathbb{R}^N$.

External forces can also be considered explicitly as a space- and a time-dependent part, $\widehat{\boldsymbol{F}}(t) = \boldsymbol{B}\boldsymbol{F}(t)$:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \widehat{\boldsymbol{f}}\big(\dot{\boldsymbol{q}}(t), \boldsymbol{q}(t)\big) = \boldsymbol{B}\boldsymbol{F}(t), \quad \boldsymbol{q}(0) = \boldsymbol{q}_0,\ \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0. \tag{2.2}$$

This allows for a system-theoretic point of view where the input–output behavior is of importance: The input matrix $\boldsymbol{B} \in \mathbb{R}^{N\times p}$ contains weights and allocations to the degrees of freedom of the time-dependent forces, i. e., the input signals $\boldsymbol{F}(t) \in \mathbb{R}^p$ ($p \leq N$). The corresponding system outputs are given by equation (2.5).

Modeling of the dominating damping mechanisms is mostly not straightforward. Therefore, one frequently gets by with assuming simple linear viscous damping $\boldsymbol{D}\dot{\boldsymbol{q}}(t)$. Additionally excluding gyroscopic effects allows for writing $\widehat{\boldsymbol{f}}(\dot{\boldsymbol{q}}(t), \boldsymbol{q}(t)) = \boldsymbol{D}\dot{\boldsymbol{q}}(t) + \boldsymbol{f}(\boldsymbol{q}(t))$, such that

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{D}\dot{\boldsymbol{q}}(t) + \boldsymbol{f}\big(\boldsymbol{q}(t)\big) = \boldsymbol{B}\boldsymbol{F}(t), \quad \boldsymbol{q}(0) = \boldsymbol{q}_0,\ \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0, \tag{2.3}$$

with damping matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ and nonlinear internal forces $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$.

For sufficiently small displacements around an equilibrium position, used as initial configuration, only the linear part of the internal restoring forces, $\boldsymbol{f}(\boldsymbol{q}(t)) \approx \boldsymbol{K}\boldsymbol{q}(t)$, has to be considered. This leads to the well-known linear second-order representation

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{D}\dot{\boldsymbol{q}}(t) + \boldsymbol{K}\boldsymbol{q}(t) = \boldsymbol{B}\boldsymbol{F}(t), \quad \boldsymbol{q}(0) = \boldsymbol{q}_0,\ \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0, \tag{2.4}$$

with stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$.

Typically, displacements, velocities, or linear combinations, e. g., stresses, at specific nodes constitute the system outputs:

$$\boldsymbol{y}(t) = \boldsymbol{C_q}\boldsymbol{q}(t) + \boldsymbol{C_{\dot{q}}}\dot{\boldsymbol{q}}(t), \tag{2.5}$$

with output matrices $\boldsymbol{C_q} \in \mathbb{R}^{q\times N}$ and $\boldsymbol{C_{\dot{q}}} \in \mathbb{R}^{q\times N}$ considering displacements and velocities, respectively.

The transfer behavior – inputs to outputs – is commonly represented as

$$\boldsymbol{G}(s) = (\boldsymbol{C_q} + s\boldsymbol{C_{\dot{q}}})(s^2\boldsymbol{M} + s\boldsymbol{D} + \boldsymbol{K})^{-1}\boldsymbol{B} \tag{2.6}$$

in the frequency domain, such that

$$\boldsymbol{Y}(s) = \boldsymbol{G}(s)\boldsymbol{F}(s), \tag{2.7}$$

where $\boldsymbol{Y}(s)$ and $\boldsymbol{F}(s)$ are the Laplace-transformed outputs $\boldsymbol{y}(t)$ and inputs $\boldsymbol{F}(t)$, respectively.

While second-order representations are common in mechanics, state-space representations are often used in systems and control theory. Different implicit state-space representations exist, possessing desired properties for their system matrices. Here only, except for basis transformations, unique explicit representations will be given, while implicit ones are to be favored when, e. g., preservation of the matrix sparsity pattern is desired. The general nonlinear system representation (2.2) can be reformulated as

$$\begin{bmatrix}\dot{\boldsymbol{q}}(t)\\ \ddot{\boldsymbol{q}}(t)\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}}(t)\\ -\boldsymbol{M}^{-1}\widehat{\boldsymbol{f}}(\dot{\boldsymbol{q}}(t),\boldsymbol{q}(t))\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{-1}\boldsymbol{B}\end{bmatrix}\boldsymbol{F}(t), \quad \begin{bmatrix}\boldsymbol{q}(0)\\ \dot{\boldsymbol{q}}(0)\end{bmatrix} = \begin{bmatrix}\boldsymbol{q}_0\\ \dot{\boldsymbol{q}}_0\end{bmatrix} \tag{2.8}$$

and the linear system representation (2.4) as

$$\begin{bmatrix}\dot{\boldsymbol{q}}(t)\\ \ddot{\boldsymbol{q}}(t)\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{D}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}(t)\\ \dot{\boldsymbol{q}}(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{-1}\boldsymbol{B}\end{bmatrix}\boldsymbol{F}(t), \quad \begin{bmatrix}\boldsymbol{q}(0)\\ \dot{\boldsymbol{q}}(0)\end{bmatrix} = \begin{bmatrix}\boldsymbol{q}_0\\ \dot{\boldsymbol{q}}_0\end{bmatrix}, \tag{2.9}$$

together with the output equation (2.5) rewritten as

$$\boldsymbol{y}(t) = \begin{bmatrix}\boldsymbol{C_q} & \boldsymbol{C_{\dot{q}}}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}(t)\\ \dot{\boldsymbol{q}}(t)\end{bmatrix}. \tag{2.10}$$

Systems involving coupling of either different components, i. e., multibody systems, or of different physical domains, i. e., multiphysics systems, can show mass matrices $\boldsymbol{M}$ in second-order representations (or left-hand side matrices $\boldsymbol{E}$ in implicit state-space representations $\boldsymbol{E}\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$, $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$) with rank deficiencies due to the algebraic interface equations. Thus, the system dynamics are described by differential algebraic equations (DAEs). An example is given in Section 2.2.

Mass ($\boldsymbol{M}$) and stiffness ($\boldsymbol{K}$) matrices are symmetric and positive (semi-)definite for typical mechanical systems with appropriate boundary conditions suppressing rigid body modes. Implying those matrix properties directly results in the following system properties:

- *Passivity*: A mechanical system is passive for collocated inputs and velocity only outputs, i. e., $\boldsymbol{B} = \boldsymbol{C}_{\dot{\boldsymbol{q}}}^{\mathrm{T}}$, $\boldsymbol{C}_{\boldsymbol{q}} = \boldsymbol{0}$ [109].
- *Stability*: A mechanical system is always Lyapunov stable. It is also asymptotically stable for positive definite $\boldsymbol{D}$ [59].

Commonly, linear damping is realized via modal damping. A simple and popular choice is the special case of proportional or Rayleigh damping, where $\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$ with $\alpha, \beta \geq 0$, i. e., the damping matrix ($\boldsymbol{D}$) is computed as a linear combination of mass ($\boldsymbol{M}$) and stiffness ($\boldsymbol{K}$) matrices [41]. With this, $\boldsymbol{D}$ is symmetric and positive definite, the same as $\boldsymbol{M}$ and $\boldsymbol{K}$. The conjugate complex eigenvalue pairs $s_{i,i+1} = \sigma \pm \mathrm{i}\omega$ of undamped systems are all located on the imaginary axis. A mass proportional part, i. e., $\alpha \geq 0$, shifts all eigenvalues with equal amount to the left, $\Delta\sigma = \text{const.}$, while a stiffness proportional part, i. e., $\beta \geq 0$, shifts all eigenvalues to the left with an amount proportional to the frequency of the eigenvalue squared, $\Delta\sigma \propto \omega^2$.

In order to reduce the computational effort involved in numerically solving (2.4), a reduced-order model (ROM) that accurately approximates the behavior of the original full-order model (FOM) is aimed. This is usually achieved by *projective* MOR. To this end, the full displacements $\boldsymbol{q}(t) \in \mathbb{R}^N$ are first approximated by a linear combination of reduced displacements $\boldsymbol{q}_{\mathrm{r}}(t) \in \mathbb{R}^n$ via the ansatz $\boldsymbol{q}(t) = \boldsymbol{V}\,\boldsymbol{q}_{\mathrm{r}}(t) + \boldsymbol{e}(t)$, where $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ and $n \ll N$. Inserting this ansatz in (2.4), (2.5) yields an overdetermined system with the residual $\boldsymbol{\varepsilon}(t) \in \mathbb{R}^N$,

$$\begin{aligned} \boldsymbol{M}\,\boldsymbol{V}\,\ddot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{D}\,\boldsymbol{V}\,\dot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{K}\,\boldsymbol{V}\,\boldsymbol{q}_{\mathrm{r}}(t) &= \boldsymbol{B}\,\boldsymbol{F}(t) + \boldsymbol{\varepsilon}(t), \\ \boldsymbol{y}_{\mathrm{r}}(t) &= \boldsymbol{C}_{\boldsymbol{q}}\,\boldsymbol{V}\,\boldsymbol{q}_{\mathrm{r}}(t) + \boldsymbol{C}_{\dot{\boldsymbol{q}}}\,\boldsymbol{V}\,\dot{\boldsymbol{q}}_{\mathrm{r}}(t). \end{aligned} \tag{2.11}$$

To obtain a square system, we premultiply (2.11) by $\boldsymbol{W}^{\mathrm{T}} \in \mathbb{R}^{n\times N}$,

$$\begin{aligned} \boldsymbol{M}_{\mathrm{r}}\,\ddot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{D}_{\mathrm{r}}\,\dot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{K}_{\mathrm{r}}\,\boldsymbol{q}_{\mathrm{r}}(t) &= \boldsymbol{B}_{\mathrm{r}}\,\boldsymbol{F}(t), \quad \boldsymbol{q}_{\mathrm{r}}(0) = \boldsymbol{q}_{\mathrm{r},0},\ \dot{\boldsymbol{q}}_{\mathrm{r}}(0) = \dot{\boldsymbol{q}}_{\mathrm{r},0}, \\ \boldsymbol{y}_{\mathrm{r}}(t) &= \boldsymbol{C}_{\boldsymbol{q}_{\mathrm{r}}}\,\boldsymbol{q}_{\mathrm{r}}(t) + \boldsymbol{C}_{\dot{\boldsymbol{q}}_{\mathrm{r}}}\,\dot{\boldsymbol{q}}_{\mathrm{r}}(t), \end{aligned} \tag{2.12}$$

thus enforcing the *Petrov–Galerkin* condition $\boldsymbol{W}^{\mathrm{T}}\,\boldsymbol{\varepsilon}(t) = \boldsymbol{0}$, where the residual $\boldsymbol{\varepsilon}(t)$ vanishes. The reduced matrices are given by $\{\boldsymbol{M}_{\mathrm{r}}, \boldsymbol{D}_{\mathrm{r}}, \boldsymbol{K}_{\mathrm{r}}\} = \boldsymbol{W}^{\mathrm{T}}\{\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K}\}\boldsymbol{V}$, $\boldsymbol{B}_{\mathrm{r}} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{B}$, $\boldsymbol{C}_{\boldsymbol{q}_{\mathrm{r}}} = \boldsymbol{C}_{\boldsymbol{q}}\,\boldsymbol{V}$, and $\boldsymbol{C}_{\dot{\boldsymbol{q}}_{\mathrm{r}}} = \boldsymbol{C}_{\dot{\boldsymbol{q}}}\,\boldsymbol{V}$, and the initial conditions are $\{\boldsymbol{q}_{\mathrm{r}}(0), \dot{\boldsymbol{q}}_{\mathrm{r}}(0)\} = (\boldsymbol{W}^{\mathrm{T}}\boldsymbol{V})^{-1}\boldsymbol{W}^{\mathrm{T}}\{\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0\}$. Therefore, the main task of any projective MOR technique consists of finding suitable reduction bases $\boldsymbol{V}, \boldsymbol{W} \in \mathbb{R}^{N\times n}$ that span appropriate subspaces $\mathcal{V} = \mathrm{span}(\boldsymbol{V})$ and $\mathcal{W} = \mathrm{span}(\boldsymbol{W})$.

MOR in mechanical engineering typically aims at achieving the following goals:

1. *Good approximation*. The reduction technique should yield a ROM which captures the most dominant dynamics and well approximates the state vector or input–output behavior of the FOM either in the time or in the frequency domain. The approximation quality can, for instance, be measured pointwise in time by $\|\boldsymbol{y}(t) - \boldsymbol{y}_{\mathrm{r}}(t)\|_{(\cdot)}$ or $\|\boldsymbol{x}(t) - \boldsymbol{V}\,\boldsymbol{x}_{\mathrm{r}}(t)\|_{(\cdot)}$, or pointwise in frequency by $\|\boldsymbol{G}(\mathrm{i}\omega) - \boldsymbol{G}_{\mathrm{r}}(\mathrm{i}\omega)\|_{(\cdot)}$ using

suitable matrix and vector norms $(\cdot) = \{1, 2, \infty, \mathrm{Fro}, \ldots\}$. Another possibility is to use normwise error measures as $\|\boldsymbol{y}-\boldsymbol{y}_\mathrm{r}\|_{(*)}$ or $\|\boldsymbol{x}-\boldsymbol{V}\boldsymbol{x}_\mathrm{r}\|_{(*)}$ with $(*) = \{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_\infty, \ldots\}$ in the time domain, or $\|\boldsymbol{G}-\boldsymbol{G}_\mathrm{r}\|_{(*)}$ with $(*) = \{\mathcal{H}_2, \mathcal{H}_\infty, \mathrm{Hankel}\ldots\}$ in the frequency domain.

2. *Preservation of system properties*. Basic features of the original model (e. g., stability, passivity, second-order structure, port-Hamiltonian structure, etc.) should be preserved in the ROM. This requirement is achieved by applying special or adapted reduction methods tailored to address these demands.
3. *Numerical efficiency*. Model reduction pays off if the benefit of having multiple, cheap online evaluations (required, e. g., for design analysis, optimization, and control) outweighs the upfront offline cost needed for the computation of the reduced model. Thus, the reduction methods should be as numerically efficient and stable as possible. Expensive offline, and especially online, computations should be avoided. In addition, reduction approaches should preferably be applicable to large-scale models and industrial problems.

Depending on the application and the characteristic behavior of the FOM that should be approximated during the reduction, two categories can be distinguished to meet requirement 1:

(i) *Initial condition-state-based reduction*. This category is especially interesting for mechanical engineering, where the eigendynamics, i. e., the state dynamics described for different initial conditions, are particularly relevant. Modal reduction techniques, such as modal truncation [94], mainly focus on the approximation of the homogeneous problem, i. e., the eigendynamics of the underlying model. POD also falls under this category, since it is based on snapshots of the simulated state trajectory.

(ii) *Input-output-based reduction*. This category is especially interesting for control engineering or for problems where inputs and outputs are defined. In this regard, approaches such as balanced truncation [78, 44] and Krylov subspace methods [42, 10] exploit the information contained in the input and output matrices $\boldsymbol{B}$, $\boldsymbol{C}$ to obtain a reduced model which is tailored to approximate the input–output behavior.

To meet requirement 2, so-called *structure-preserving model reduction* is applied. For instance, if the principle of virtual work known from mechanical systems should be fulfilled, then the reduction should be performed by a *Galerkin projection* with $\boldsymbol{W} = \boldsymbol{V}$ rather than by a two-sided (oblique) Petrov–Galerkin projection. Note, however, that most second-order balancing approaches do not underlie a Galerkin projection. In any case, with $\boldsymbol{W} = \boldsymbol{V}$, the reduced matrices preserve the symmetry and definiteness properties of the original matrices. Furthermore, this choice leads to the foremost aim of preserving crucial properties such as the stability, passivity, and structure of the original FOM. Note that second-order models could also be reduced by first reformulating them into the state-space/first-order representation (with $\tilde{N} = 2N$), and

then by reducing the first-order models. However, unless special care is taken during the reduction of the reformulated state-space model, it can be difficult to gain back a structure-preserving, second-order ROM out of the first-order ROM [102, 103, 85, 21]. Consequently, MOR for mechanical systems is often performed directly on the second-order FOM by applying *second-order reduction techniques* suited to meet the mentioned requirements and needs. Hereby, either (i) classic modal-based procedures (such as second-order modal truncation) or (ii) *adapted* input–output-based approaches (e. g., second-order Arnoldi [SOAR] [6, 9, 102], SO-IRKA [112], or second-order balanced truncation [76, 108, 17, 16]) can be applied. Note, however, that some of these reduction techniques still use the first-order representation to compute, e. g., the Gramians or the optimal interpolation data, in order to build afterwards second-order reduction bases $\boldsymbol{V}, \boldsymbol{W}$ to project the second-order model.

It is also worth mentioning that in many engineering applications multiphysics problems arise, yielding *coupled domain models*. Similarly, in structural dynamics and industrial applications, the mechanical systems often consist of several separable components or substructures. In this context, the concept of *substructuring* and *component mode synthesis* (CMS) [48, 56, 24, 1] plays a key role, allowing to partition a large structural model into multiple substructures that are individually reduced, and then compatibly coupled along the component interfaces after reduction. Typical CMS techniques are Guyan condensation [48], the Craig–Bampton method [24], and their derivatives [56, 1]. Craig–Bampton, e. g., represents the combination of (i) a modal-based and (ii) an input–output-based procedure, since (i) "component eigenmodes" are combined with so-called (static/dynamic) (ii) "constraint modes" obtained for a unit displacement applied to the interface degrees of freedom. In addition to the classic CMS techniques, system-level *interface reduction* approaches that reduce the number of interface degrees of freedom are also very common [51]. Finally, note that input–output-based reduction approaches such as balanced truncation and Krylov subspace methods can naturally be applied instead of the modal-based "component eigenmodes." This has been done, e. g., in [33, 84, 51], where (static/dynamic) (i) "constraint modes" have been combined with (ii) "input–output-based component modes."

2.2 Model order reduction of a thermo-mechanical machine tool model aimed for position control

For modern machine tools, the accuracy, stability, and repeatability are the key performance factors. Thermally induced displacement errors at the tool center point (TCP) are among the main causes of work piece defects. The heat generated by motors, the friction-caused heat, the influence of the environment temperature, etc., are the fac-

tors which lead to unwanted thermal expansion of the machine tool [29, 77, 82] and require a real-time compensation (position control). In the past few years, the collaborative research center Transregio 96 [43] contributed a great deal to the optimal thermo-energetic design of machine tools. Numerical simulation via, e. g., the FEM is a common tool to compute the thermal expansion. However, the accurate numerical model is too large to be co-simulated with the control circuitry.

In the following, the successful application of Krylov subspace-based MOR from [39, 6, 102] is presented for creating a reduced-order thermo-mechanical machine tool model. Our case study is an academic model, displayed in Figure 2.1.

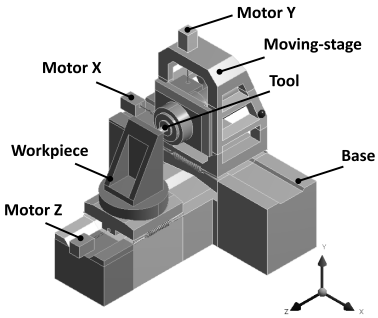


Figure 2.1: Structure of the machine tool model.

For demonstrating purpose, we observe a single stage movement in the positive x-direction and a single heat generation in the spindle of the tool. The heat generation is due to electrical drive only, whereas the friction-caused heat is neglected. Furthermore, ideal thermal contacts and temperature-independent material thermal parameters (volumetric heat capacity and heat conductivity) are assumed. The mechanical parts of the model are connected by linear springs, instead of using more realistic ball bearings, in which the frictional and thermal effects should be considered [65]. These assumptions lead to a linear thermo-mechanical model, which can be reduced by Krylov subspace-based MOR. Please note that as long as the above linearity conditions are fulfilled, this methodology can be applied to more realistic models as well. The primary goal of this section is to demonstrate that the position control can be implemented based on such an ROM. Further applications of MOR to machine tool models can be found in [32, 96, 81, 68].

2.2.1 Coupled domain thermo-mechanical models

In the thermo-mechanical model, both thermal and mechanical domains are taken into account during the simulation. The thermal and mechanical domains can be cou-

pled via the input vector or via the system matrices (strong coupling). As the thermal expansion is a transient process, which requires real-time compensation, the coupling via the system matrices is recommended. Such a strongly coupled linearized machine tool model has the following form:

$$\boldsymbol{\Sigma}_{N+k}: \begin{cases} \underbrace{\begin{bmatrix} \boldsymbol{M_q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\boldsymbol{M}} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{T}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{D_q} & \mathbf{0} \\ \boldsymbol{D_{Tq}} & \boldsymbol{D_T} \end{bmatrix}}_{\boldsymbol{D}} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{T}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{K_q} & \boldsymbol{K_{qT}} \\ \mathbf{0} & \boldsymbol{K_T} \end{bmatrix}}_{\boldsymbol{K}} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T} \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \\ \boldsymbol{y} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T} \end{bmatrix}, \end{cases} \tag{2.13}$$

where $\boldsymbol{M_q}, \boldsymbol{D_q}, \boldsymbol{K_q} \in \mathbb{R}^{N\times N}$ are the mass matrix, the structural damping matrix, and the stiffness matrix, respectively; $\boldsymbol{D_T}, \boldsymbol{K_T} \in \mathbb{R}^{k\times k}$ are the specific heat matrix and the conductivity matrix; $\boldsymbol{D_{Tq}} \in \mathbb{R}^{k\times N}$ is the thermo-elastic damping matrix and $\boldsymbol{K_{qT}} \in \mathbb{R}^{N\times k}$ is the thermo-elastic stiffness matrix. Matrices $\boldsymbol{D_{Tq}}$ and $\boldsymbol{K_{qT}}$ couple the thermal and mechanical domains. The state vector contains two parts, i. e., the structural displacement vector $\boldsymbol{q}(t) \in \mathbb{R}^N$ and the temperature vector $\boldsymbol{T}(t) \in \mathbb{R}^k$. In this model a single force input u_1 is applied onto the moving stage in the positive x-direction and a single heat generation input u_2 is assumed in the spindle of the tool. The output matrix $\boldsymbol{C} \in \mathbb{R}^{2\times(N+k)}$ defines the observed outputs, displacement of the moving stage at the selected node in the positive x-direction and the temperature at the TCP. The machine tool is fixed at the bottom ground and the bottom temperature is set to 20 °C. The heat exchange between system and environment is modeled by the convection boundary condition $\boldsymbol{\phi_T} = h(\boldsymbol{T} - \boldsymbol{T}_{\text{ambient}})$, where $\boldsymbol{\phi_T}$ is a heat flux and h is the heat transfer coefficient, which is set to 5 $\frac{\text{W}}{\text{m}^2}$. The convection is considered over the whole surface of the machine tool model and the ambient temperature is set to 20 °C. The Rayleigh damping is considered, that is, the damping matrix $\boldsymbol{D_q}$ is proportional to the mass matrix $\boldsymbol{M_q}$ and the stiffness matrix $\boldsymbol{K_q}$ as follows: $\boldsymbol{D_q} = \alpha\boldsymbol{M_q} + \beta\boldsymbol{K_q}$. In this case, based on experience, parameters $\alpha = 0$ and $\beta = 1.5915 \cdot 10^{-4}$ are chosen to cause the damping ratio of 1 % at a frequency of 20 Hz.

The advantage of model (2.13) is that all physical effects are included. However, as will be shown in Section 2.2.2, MOR of this model leads to unnecessarily large ROMs, whose time integration within the controller loop might be prohibitive. On the other hand, as the time scale of structural dynamics (≈ 1 s) is much smaller than that of the heat transfer ($\approx 10^5$ s), one might think of separating physical domains and integrating them at the system level with different time steps, as suggested in [15].

Therefore, the so-called quasi-static approximation of structural mechanics is used, in which the mass matrix $\boldsymbol{M_q}$ and the structural damping matrix $\boldsymbol{D_q}$ from equa-

tion (2.13) are ignored. The resulting model has the following form:

$$\boldsymbol{\Sigma}_{N+k\text{-Quasi}}: \begin{cases} \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{D_T} \end{bmatrix}}_{\boldsymbol{D}} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{T}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{K_q} & \boldsymbol{K_{qT}} \\ \mathbf{0} & \boldsymbol{K_T} \end{bmatrix}}_{\boldsymbol{K}} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T} \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \\ \boldsymbol{y} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T} \end{bmatrix}, \end{cases} \tag{2.14}$$

and accounts for thermal expansion of the mechanical structure and the transient heat transfer. Furthermore, the thermo-elastic damping $\boldsymbol{D_{Tq}}$, which describes the influence of structural dynamics upon the temperature field, can be neglected as well, because the dissipated heat is negligible compared to the motor-generated heat. Please note that the reduction of the DAE system (2.14) with Krylov subspace-based MOR is numerically well conditioned, since those methods solely impose that the matrix $\boldsymbol{K} + s_0\boldsymbol{D}$ is regular for some value s_0 of the Laplace variable [6].

At the system level, (2.14) has to be combined with the purely mechanical model:

$$\Sigma_{\mathrm{N}}: \begin{cases} \boldsymbol{M_q}\ddot{\boldsymbol{q}} + \boldsymbol{D_q}\dot{\boldsymbol{q}} + \boldsymbol{K_q}\boldsymbol{q} = \boldsymbol{B_q}u_1, \\ y = \boldsymbol{C_q}\boldsymbol{q}, \end{cases} \tag{2.15}$$

that is, with its reduced counterpart, to reflect the dynamical behavior of the structure within a control loop.

2.2.2 Application of Krylov subspace-based MOR

The main issue in reducing coupled domain thermo-mechanical machine tool models lies in the fact that both physical domains have very different time constants, as the time scale of the structural part is much smaller than that of the heat transfer. In the following, the Krylov subspace-based MOR is applied, which matches the moments (coefficients of the Taylor series) of the transfer functions of the full and reduced models. For the reduction of model (2.13), the SOAR algorithm from [6] and [102] is used. For the first-order system (2.14) and for the proportionally damped second-order system (2.15), the first-order block Arnoldi algorithm from [39] is implemented (for the application to (2.15), see [28]).

Two inputs (mechanical force applied at a selected surface of the moving stage and heat source in spindle) and two outputs (displacement at selected node of the moving stage and the temperature at TCP) are defined, which leads to the following 2×2 transfer function matrix:

$$\boldsymbol{G}(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}, \tag{2.16}$$

where $G_{11}(s) = \frac{Y_{\text{displ}}(s)}{U_{\text{force}}(s)}$, $G_{12}(s) = \frac{Y_{\text{temp}}(s)}{U_{\text{force}}(s)}$, $G_{21}(s) = \frac{Y_{\text{displ}}(s)}{U_{\text{heat}}(s)}$, $G_{22}(s) = \frac{Y_{\text{temp}}(s)}{U_{\text{heat}}(s)}$, and $Y_{\text{displ}}(s)$ and $Y_{\text{temp}}(s)$ are the Laplace transforms of the displacement and temperature outputs, respectively, defined by matrix $\boldsymbol{C}$ in (2.13); $U_{\text{force}}(s)$ and $U_{\text{heat}}(s)$ are the Laplace transforms of the force and heat inputs, respectively, defined as $u_1(t)$ and $u_2(t)$ in (2.13).

The goal is to investigate the convergence behavior of each reduced model for different model sizes and compare them with respect to their applicability for position control. The relative error between the full and the reduced model transfer functions at a specific frequency f is defined as

$$\epsilon(f) = \left\|\boldsymbol{G}(\mathrm{i}2\pi f) - \boldsymbol{G}_{\mathrm{r}}(\mathrm{i}2\pi f)\right\|_2 / \left\|\boldsymbol{G}(\mathrm{i}2\pi f)\right\|_2. \tag{2.17}$$

For the harmonic simulations of the full-scale model, we use the FEM simulator ANSYS Academic Research, Rel. 18.0, whereas the reduced models are created with model reduction inside ANSYS [98].

Firstly, the purely mechanical proportionally damped model (2.15) with the single force input (force applied at a selected surface of the moving stage in the positive x-direction in Figure 2.1) and a single output (displacement at selected node of the moving stage) is investigated.

Figure 2.2 (left) shows an excellent match between the transfer function $G_{11}(s)$ of the full model with 51,378 degrees of freedom and the reduced-order 100 model. The expansion point 0 Hz is chosen for the Arnoldi-based reduction and for matching the frequency range between 1 Hz and 1000 Hz, which is the frequency range of interest for the particular application. Note that other frequency ranges of interest can be matched by employing different expansion points [42]. From Figure 2.2 (right), it is observed that for matching the transfer function with maximal relative error of 1 % over the whole frequency range of interest, a moderate number of 30 Arnoldi vectors – which represent the orthonormal basis of the Krylov subspace $\mathcal{K}(-\boldsymbol{K}_q^{-1}\boldsymbol{M}_q, -\boldsymbol{K}_q^{-1}\boldsymbol{B}_q)$ – is required if no other expansion points are to be employed.

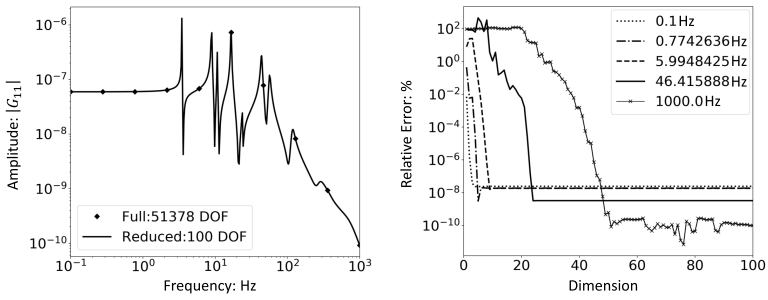


Figure 2.2: Transfer function $G_{11}(s)$ of purely mechanical full-scale model (2.15) and corresponding reduced-order 100 model, gained by Arnoldi-based reduction with expansion point 0 Hz (left). Relative error between the transfer functions of full-scale model (2.15) and corresponding reduced models of different sizes at different frequencies (right).

The coupled domain model (2.13) suffers from different time constants of structural dynamics and heat transfer, as the ROM has to simultaneously match the frequency ranges from 0 Hz to 10 Hz (thermal expansion effects occur here) and from 1 Hz to 1000 Hz (quick changes in dynamics occur here).

Figure 2.3 (top) shows an excellent match of the transfer functions $G_{11}(s)$ and $G_{12}(s)$ of the thermo-mechanical model (2.13) exposed solely to a single unit force input. Figure 2.3 (bottom) shows an excellent match of the transfer functions $G_{21}(s)$ and $G_{22}(s)$ of the thermo-mechanical model (2.13) exposed solely to a single unit heat input; $G_{21}(s)$ describes the thermal expansion. In this case we used the unit input for computing the transfer function, as the model is linear. At the system level, an arbitrary input can be used.

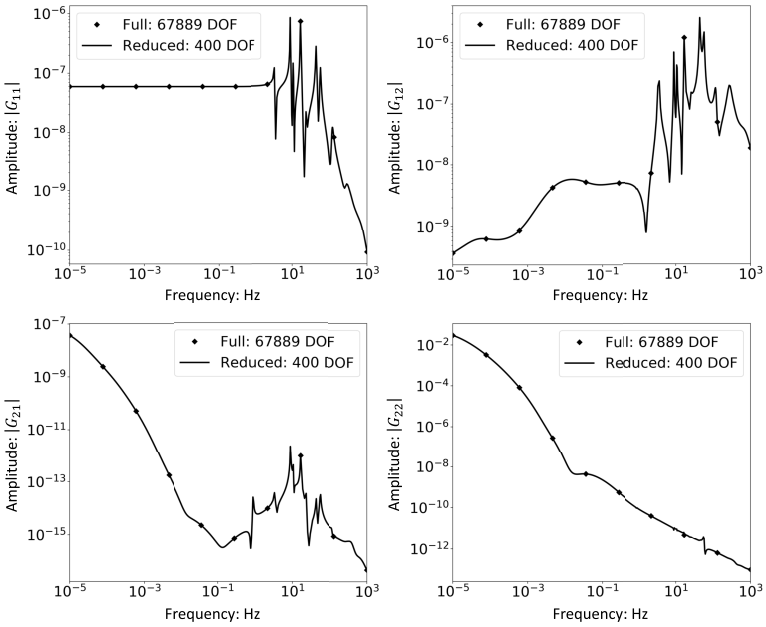


Figure 2.3: Transfer functions $G_{11}(s)$ (top-left), $G_{12}(s)$ (top-right) with single force input and $G_{21}(s)$ (bottom-left), $G_{22}(s)$ (bottom-right) with single heat input of strongly coupled full-scale model (2.13) and corresponding reduced models of order 400, gained by the SOAR algorithm with expansion points 0 Hz, 10 Hz, 100 Hz, and 1000 Hz.

In order to match the full required frequency range from 0 Hz to 1000 Hz, with maximal relative error of 1 %, 300, respectively, 250 Arnoldi vectors and four expansion points (0 Hz, 10 Hz, 100 Hz, and 1000 Hz) are required (Figure 2.4). This high number of 250 vectors is the main disadvantage of reducing the strongly coupled model (2.13). Note that more modern methods allow for an automatic choice of the expansion points and

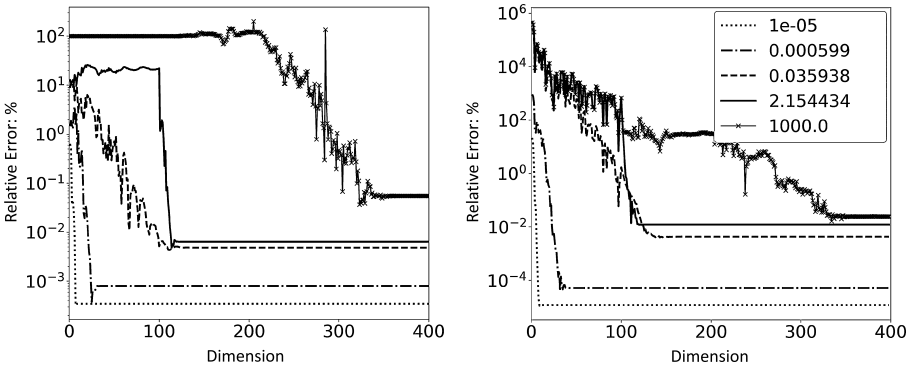


Figure 2.4: Relative error (2.17) between the transfer functions of full and reduced model at different frequencies for different sizes of reduced models with only force input (left) and with only heat input (right) in strongly coupled thermo-mechanical model (2.13).

the corresponding number of moments. In this work, however, we adopted the practical engineering approach of "trial and error," since our goal was to apply the classical MOR methodology to our coupled physics, realistic industrial application rather than to apply or improve adaptive reduction procedures. Future research should include more modern methods based on adaptive approaches such as, e. g., IRKA, TSIA, and CUREd SPARK from [45, 27, 113, 85, 111].

It is expected that the reduction of the coupled model with quasi-static approximation of the mechanical domain (2.14) leads to reduced models with lower order than the reduction of (2.13). Furthermore, (2.14) can be reduced with the first-order Krylov subspace algorithm from [39]. Figure 2.5 shows an excellent match of the transfer functions $G_{21}(s)$ and $G_{22}(s)$ of the full-scale and reduced-order 100 thermo-mechanical model with quasi-static approximation of the mechanical domain (2.14). It is exposed to a single heat input and $G_{21}(s)$ describes the thermal expansion, which should be accounted for within the position control. In this case, however, there is no need to cover the frequency range until 1000 Hz, as the structural dynamics are not included. Rather, the frequency range of interest from 0 Hz to 10 Hz should be approximated. For this, three expansion points (chosen by "trial and error") are still needed: 0 Hz, 0.1 Hz, and 10 Hz.

Figure 2.6 shows that for matching the transfer function $\boldsymbol{G}(s)$ of the fully coupled model with quasi-static approximation of the mechanical domain and the single heat input, with maximal relative error of 1 %, within the frequency range of interest, solely 50 Arnoldi vectors and three expansion points are needed, which outperforms the reduced models of system (2.13).

In conclusion, by introducing the quasi-static approximation of the mechanical domain, the accuracy of the reduced thermo-mechanical model could be improved and the dimension of the reduced model could be decreased. For the complete approximation of the frequency range of interest, this model has to be combined with

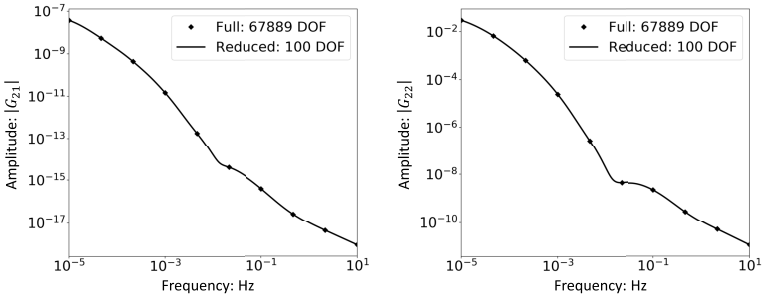


Figure 2.5: Transfer functions $G_{21}(s)$ (left) and $G_{22}(s)$ (right) for single heat input of strongly coupled full-scale model with quasi-static approximation of mechanical domain (2.14) and corresponding reduced-order 100 models, gained by Arnoldi-based reduction with expansion points 0 Hz, 0.1 Hz, and 10 Hz.

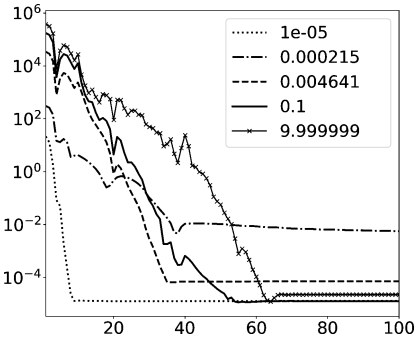


Figure 2.6: Relative error (2.17) between the transfer functions of full and reduced model at different frequencies for different sizes of reduced model with only heat input in strongly coupled thermo-mechanical model with quasi-static approximation of the mechanical domain (2.14).

the (reduced) purely mechanical model at the system level, as will be shown in the next section.

2.2.3 Position control scenario based on reduced-order model

For position control of machine tools usually a cascade controller structure is applied [40]. Our goal is to parameterize such a controller based on an ROM. Figure 2.7 shows the ANSYS Twin Builder simulation setup.

In Figure 2.8 (left) the impact of the position control to the moving stage, when taking into account only structural dynamics of the machine, is displayed and Figure 2.8 (right) shows the gain-parameter optimization of the controller, based on the reduced-order purely mechanical model (2.15) with dimension 30. Note that timing

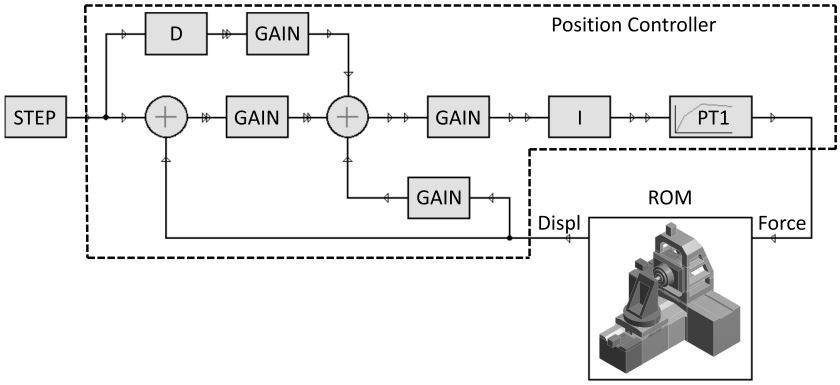


Figure 2.7: Co-simulation of the cascade controller and the reduced-order model (ROM) of the machine tool in ANSYS Twin Builder.

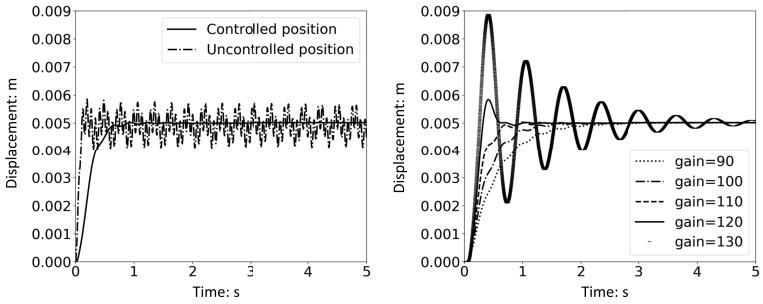


Figure 2.8: Output node displacement of the reduced purely mechanical model (2.15) with and without controller displayed in Figure 2.7 (left) and optimization of the controller gain based on the reduced-order numerical model (right).

for five simulations with different controller gain values amounts to solely 24 s on an Intel(R) Core(TM) i7 @ 2.50GHz RAM 8 GB.

In order to take into account thermal expansion, the reduced strongly coupled thermo-mechanical model can be applied within the control loop as schematically shown in Figure 2.9 (left).

The main disadvantage of this approach is that, due to the fact that structural dynamics and heat transfer have very different time constants and, hence, a large frequency range has to be matched, the size of the reduced model is relatively large for a system-level simulation (order 250 up to 300, as shown in the previous section). A remedy is to apply the quasi-static approximation of the mechanical domain, as described in equation (2.14). Based on the reduced-order quasi-statically approximated coupled model, it is suggested to use the structure for the positioning controller shown in Figure 2.9 (right). The advantage over the controller from Figure 2.9 (right) is that both

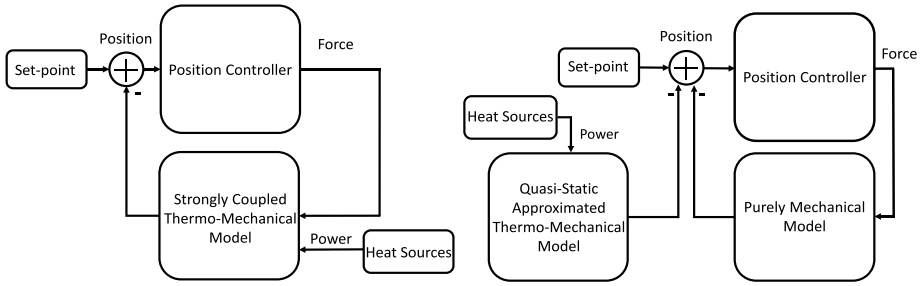


Figure 2.9: Position controller based on the reduced strongly coupled thermo-mechanical model (left) and on the reduced thermo-mechanical model with quasi-static approximation of the mechanical domain (right).

ROMs are of smaller size and can also be integrated in time separately. Also in this setup, it is possible to parameterize the controller based on the ROMs.

2.3 Coupling of reduced elastic bodies in vehicle dynamics

2.3.1 Requirements on the model reduction software and model reduction method

One important aspect in the development of vehicles is to ensure a comfortable ride feeling for the driver. In this context, it is necessary to investigate the effect of external disturbances, e. g., from potholes, on the driver in transient simulations over a long period of time [72]. Therefore, models of a driver's seat and a car body are used as an industrial automotive example in this contribution. The driver's seat and the car body, the coupling nodes, and the input and output nodes are shown in Figure 2.10.

In industrial applications, the elastic components are usually modeled with the FEM, where complex geometries require fine spatial discretizations to obtain meaningful models. For the driver's seat this leads to a model with $1.43 \cdot 10^5$ degrees of freedom and for the car body to a model with $1.99 \cdot 10^6$ degrees of freedom. The model properties are summarized in Table 2.1. The discretization and model generation are usually performed with commercial FEM software packages. Obviously, one challenge in MOR for industrial models is to access the model data, as the system matrices or geometry information from proprietary data structures, for further usage in the actual MOR. In this application, the MOR toolbox *Morembs* is used. It contains interfaces to many commercial FEM software packages and allows a user-friendly import of the model data. In a further step various MOR algorithms for second-order systems can be applied and the reduced models can be exported for further use in an industrial simulation chain. The

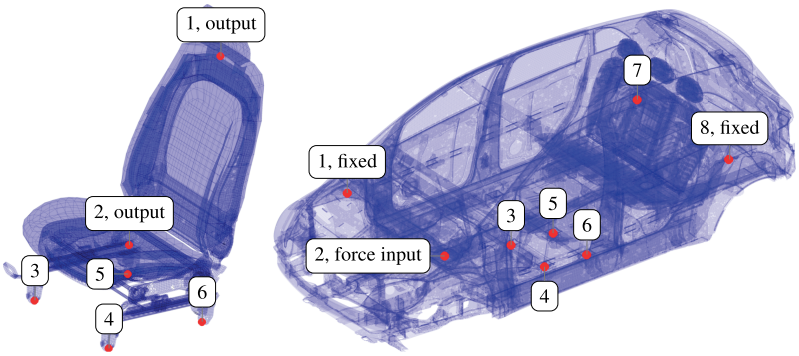


Figure 2.10: Finite element mesh of the driver's seat and the car body with inputs, outputs, and interface nodes, kindly provided by Daimler AG.

Table 2.1: Model properties of the driver's seat and the car body.

	Number of elements	Number of degrees of freedom	Number of inputs
Driver's seat	23,968	142,941	24
Car body	511,156	1,993,167	36

typical MOR work flow with *Morembs* consisting of import, reduction, and export is illustrated in Figure 2.11. An exemplary MATLAB code for import, reduction, and export with *Morembs* is given in Figure 2.12. The toolbox is freely available for academic institutions. Further information can be found in [38] and in Chapter 13 of Volume 3 of *Model order reduction*.

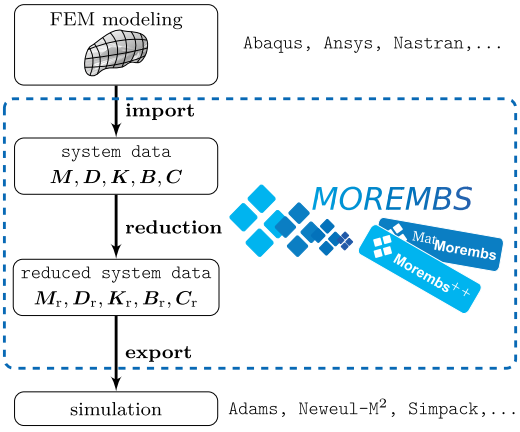


Figure 2.11: Import, reduction, and export work flow with MOR toolbox *Morembs*.

```
% start the toolbox
setupMatMorembs;
% path to data from FEM software
dataDir = 'FE/VehicleStabilizer';
% set the interaction nodes
Interactions(1).node = 13;
Interactions(1).dof = 1:6;
% import the system matrices from the FEM data
sysdata = importFEdata('workdir',dataDir,'id','VehicleStabilizer',...
    'input',Interactions);
% reduce the system with a Craig-Bampton approach to a reduced order of 10
redSysdata = MOR('sysdata',sysdata,'redmethod','Craig-Bampton','nred',10);
% export the reduced system for further use in multibody software Neweul-M^2
ElasticBody = RedExport('sysdata',redSysdata,'savedir',pwd,...
    'filename','redbody','target','neweulm2');
```

Figure 2.12: Exemplary MATLAB code for import, reduction, and export with *Morembs*.

Another challenge is that the reduced component models should be reusable in different mounting situations or product variants. This means that a MOR method must be able to perform the reduction on component level first and to deliver reduced models which can be combined afterwards with other reduced models, e. g., from a database.

One well-established method meeting these requirements is the Craig–Bampton method as suggested in [24]. This method uses a splitting into an internal part and a boundary part of the elastic body. Static constraint modes are then combined with fixed interface eigenmodes of the internal part of the elastic body. One feasible modern method meeting these requirements as well is the so-called CMS-Gram method as presented in [52]. The method uses a splitting into an internal part and a boundary part, too. Then, a modified input matrix is formulated to allow the application of input–output-based MOR methods for the internal dynamics. In this case, the internal dynamics are reduced using a two-step approach. The first step uses *moment matching* to reduce the large-scale model to a medium-size model. The second reduction to a small system size is done using *frequency-weighted balanced truncation*. Both MOR steps are directly applied to the second-order system described by equation (2.4) to obtain a reduced second-order system. First, this obviously simplifies the physical interpretation of the ROM. Second, in industrial applications it is often even necessary to preserve the second-order structure, for example if the ROMs shall be used in a commercial simulation environment. In the following, both approaches will be compared.

2.3.2 Comparison of the approximation quality of the noncoupled and the coupled system

First, the driver's seat as a single component is investigated. The Frobenius norm of the transfer function $\|\boldsymbol{G}(f)\|_{\mathrm{Fro}}$ of the original model and of two reduced models of order

$n_{\text{seat}} = 50$ are shown in Figure 2.13. One reduced model is obtained using the Craig–Bampton approach while the second reduced model is obtained using the CMS-Gram method. It can be seen that all transfer functions are in good agreement. The corresponding Frobenius norm relative error $\epsilon(f) = \|\boldsymbol{G}(f) - \boldsymbol{G}_{\text{r}}(f)\|_{\text{Fro}} / \|\boldsymbol{G}(f)\|_{\text{Fro}}$ of the reduced models is shown in Figure 2.13 for the frequency range of interest from 0 Hz to 60 Hz. Both approaches show a good approximation quality in the lower frequency range. However, the error of the Craig–Bampton reduced model exceeds 10 % around 60 Hz, which does not fulfill the accuracy requirements of the ride simulation. In contrast, the approximation error of the CMS-Gram reduced model is about one to two magnitudes smaller and, therefore, shows a satisfying approximation quality over the entire frequency range.

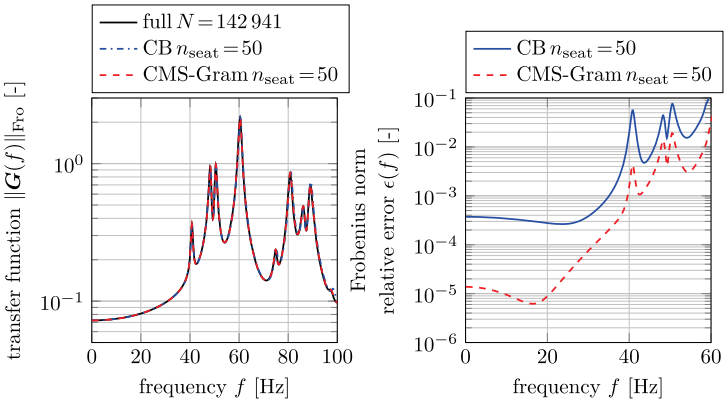


Figure 2.13: Frobenius norm of the transfer function of the driver's seat for the full model, a Craig–Bampton reduced model, and a CMS-Gram reduced model with $n_{\text{seat}} = 50$ and corresponding Frobenius norm relative errors $\epsilon(f)$.

Another aspect is the improvement of the approximation quality for an increasing order of the reduced model. For the Craig–Bampton model this can be done by adding more eigenmodes to the basis. However, identifying additional eigenmodes which are important for the transfer behavior is not always a trivial task, especially for industrial models. As for the CMS-Gram approach, the approximation can be simply improved by adding more approximated eigenvectors of the Gramian controllability matrix to the basis.

Figure 2.14 shows the Frobenius norm maximum error

$$\epsilon_{\max} = \max_{f \in [0,60]\,\text{Hz}} \left(\left\| \boldsymbol{G}(f) - \boldsymbol{G}_{\text{r}}(f) \right\|_{\text{Fro}} \right) \tag{2.18}$$

between 0 Hz and 60 Hz for an increasing order n_{seat} of the reduced system. The error for the Craig–Bampton method decreases from $\epsilon_{\max} = 2.46 \cdot 10^{-1}$ at $n_{\text{seat}} = 50$ to

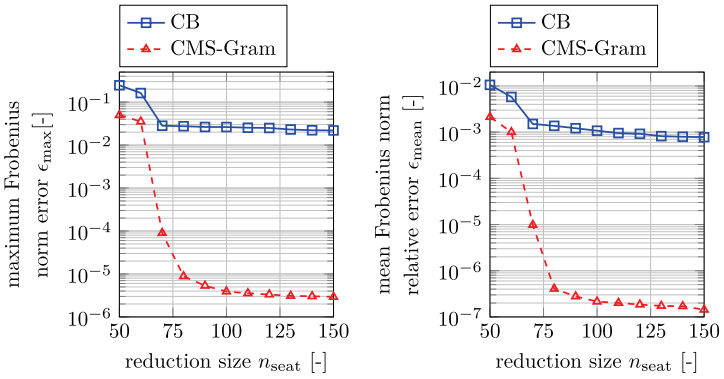


Figure 2.14: Maximum Frobenius norm error $\epsilon_{\max}$ and mean Frobenius norm relative error ϵ_{mean} for a Craig–Bampton reduction and a CMS-Gram reduction from 0 Hz to 60 Hz.

$\epsilon_{\max} = 2.19 \cdot 10^{-2}$ at $n_{\text{seat}} = 150$. The convergence of the input–output error is therefore rather slow, which is typical for modal-like MOR methods. However, the CMS-Gram method shows a very rapid convergence, leading to an error which is several magnitudes smaller, showing its superiority compared to the Craig–Bampton method. A similar behavior can be observed when investigating the mean error

$$\epsilon_{\text{mean}} = \frac{1}{60\,\text{Hz}} \int_0^{60\,\text{Hz}} \epsilon(f)\, df \tag{2.19}$$

from 0 Hz to 60 Hz, as shown in Figure 2.14. It can be seen that the CMS-Gram method delivers reduced models with much faster decreasing mean errors in the frequency range of interest compared to the Craig–Bampton method.

Next, the complete system where the driver's seat is mounted to the car body is investigated. The transfer function of the connected, damped system is depicted in Figure 2.15. It can be seen that there are more eigenfrequencies from 0 Hz to 60 Hz compared to the single model of the driver's seat, making a good approximation in this frequency range more difficult. The reduced coupled system is derived by a kinematic coupling of the reduced model of the driver's seat to the reduced model of the car body. The relative error $\epsilon(f)$ for the coupled system is shown in Figure 2.16 for a reduced order of $n_{\text{seat}} = 50$ for the driver's seat and a reduced order of $n_{\text{car}} = 200$ for the car body. Since both methods are interface-compatible, both methods deliver satisfying coupled ROMs, as well. However, the error of the reduced model obtained by the CMS-Gram method is about two to five magnitudes smaller compared to that of the Craig–Bampton reduced model. So it can be seen again that the CMS-Gram method delivers more accurate reduced models. A convergence analysis for increasing orders of the reduced models shows that the approximation error of the coupled CMS-Gram models also decreases faster.

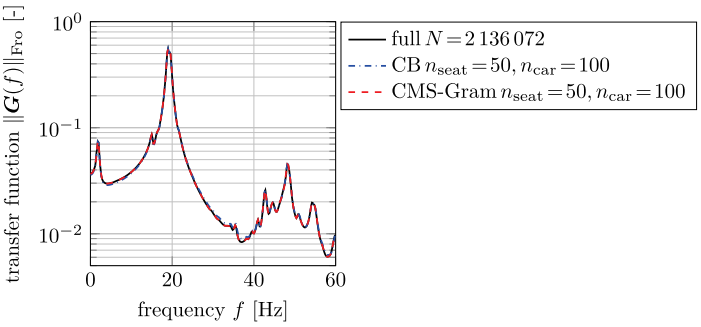


Figure 2.15: Frobenius norm of the transfer function of the coupled system for the full model, a Craig–Bampton reduced model, and a CMS-Gram reduced model reduced model with $n_{\text{seat}} = 50$ and $n_{\text{car}} = 200$.

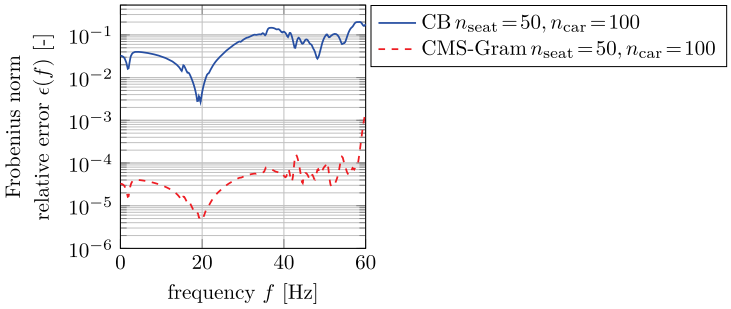


Figure 2.16: Frobenius norm relative error $\epsilon(f)$ of the coupled system for a Craig–Bampton reduced model and a CMS-Gram reduced model with $n_{\text{seat}} = 50$ and $n_{\text{car}} = 200$.

2.3.3 Concluding remarks

Two MOR methods, the modal-based Craig–Bampton method and the CMS-Gram method, are compared in this section in the context of an industrial application. Both methods deliver interface-compatible ROMs of single components. This allows the user to combine different ROMs to conduct, e. g., product variant studies. However, it is also shown that the CMS-Gram method yields ROMs with smaller relative errors, confirming the superiority of nonmodal-based MOR methods. Using nonmodal-based MOR methods as the CMS-Gram method allows the user therefore either to benefit from smaller approximation errors or to use reduced models of a smaller order with a similar approximation quality. Therefore, it became clear that it is worth considering modern input–output-based MOR methods such as frequency-weighted balanced truncation for industrial applications.

2.4 Error-controlled model order reduction of an elastic crank drive

A small-scaled, four-stroke internal combustion engine with two pistons in V configuration is depicted in Figure 2.17. The system consists of pistons, the crankshaft, and piston rods connecting each piston to the crankshaft. This mechanical model of the combustion engine is part of a multiphysics system since the gas force acting on the pistons results from a chemical reaction. In the model of this section, the gas force is approximated by an analytical function. As mentioned by [84], the elastic effects which superimpose the overall rigid body movements have a significant influence on the behavior of the crank drive.

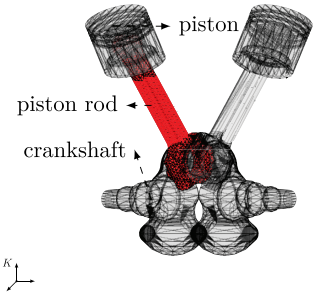


Figure 2.17: Flexible multibody system of a crank drive as major moving parts of a combustion engine.

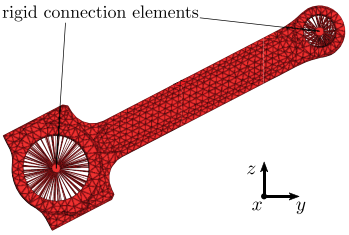


Figure 2.18: Elastic piston rod of the crank drive. Rigid connection elements (RBE2) are used to create an interface reduction.

Very often the mechanical parts of the crank drive are modeled as a flexible multibody system (FMBS) with the floating frame of reference formulation. FMBS consist of flexible and rigid bodies which are coupled by joints and coupling elements. Advantages of FMBSs are their inherently modular fashion and the description of the flexible motion with respect to the reference frame, which allows a linear description of the elasticity. One single flexible body is described with a nonlinear second-order ordinary differential equation (ODE),

$$\begin{bmatrix} \boldsymbol{M}_{\mathrm{f}}(\boldsymbol{q}) & \boldsymbol{M}_{\mathrm{ef}}^{\mathrm{T}}(\boldsymbol{q}) \\ \boldsymbol{M}_{\mathrm{ef}}(\boldsymbol{q}) & \boldsymbol{M}_{\mathrm{e}} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}(t) \\ \ddot{\boldsymbol{q}}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{e}}\boldsymbol{q}(t) + \boldsymbol{D}_{\mathrm{e}}\dot{\boldsymbol{q}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_{\mathrm{f}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{h}_{\mathrm{e}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix}, \tag{2.20}$$

which can be split into two parts: The part belonging to the motion of the floating frame, quantities with the subscript f, and the often high-dimensional flexible part, quantities with the subscript e, describing the linear elastic motion with respect to the reference frame (see, e. g., [33]). The subparts of the equation of motion are ex-

plained, e. g., in [33]. The linear elastic part usually stems from a finite element description of continua. Since it is very high-dimensional, it needs to be reduced by MOR. The procedure is made in a modular fashion. First, only the linear elastic part of (2.20),

$$\boldsymbol{M}_\mathrm{e}\ddot{\boldsymbol{q}}(t) + \boldsymbol{D}_\mathrm{e}\dot{\boldsymbol{q}}(t) + \boldsymbol{K}_\mathrm{e}\boldsymbol{q}(t) = \boldsymbol{B}_\mathrm{e}\boldsymbol{u}_\mathrm{e}(t), \tag{2.21}$$

$$\boldsymbol{y}_\mathrm{e}(t) = \boldsymbol{C}_\mathrm{e}\boldsymbol{q}(t) \tag{2.22}$$

is considered as a second-order, time-invariant multiple-input multiple-output (MIMO) system. All reaction, applied, and coupling forces – the latter especially need to be taken into account due to nonlinear rigid body motion – acting on the elastic body are considered as inputs $\boldsymbol{B}_\mathrm{e}\boldsymbol{u}_\mathrm{e}(t)$ and outputs $\boldsymbol{y}_\mathrm{e}(t) = \boldsymbol{C}_\mathrm{e}\boldsymbol{q}(t)$ to the elastic body. Using the Laplace transformation, the transfer matrix of the system $\boldsymbol{G}(s) = \boldsymbol{C}_\mathrm{e}(s^2\boldsymbol{M}_\mathrm{e} + s\boldsymbol{D}_\mathrm{e} + \boldsymbol{K}_\mathrm{e})^{-1}\boldsymbol{B}_\mathrm{e}$ is obtained.

For this second-order system, an appropriate subspace is generated by second-order structure-preserving reduction techniques, e. g., by a Galerkin ansatz $\boldsymbol{q}(t) \approx \boldsymbol{V}\boldsymbol{q}_\mathrm{r}(t)$, where $\boldsymbol{q}_\mathrm{r} \in \mathbb{R}^n$, $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ and $n \ll N$. The reduced second-order MIMO system consisting of the matrices $\{\boldsymbol{M}_\mathrm{er}, \boldsymbol{D}_\mathrm{er}, \boldsymbol{K}_\mathrm{er}\} = \boldsymbol{V}^\mathrm{T}\{\boldsymbol{M}_\mathrm{e}, \boldsymbol{D}_\mathrm{e}, \boldsymbol{K}_\mathrm{e}\}\boldsymbol{V}$, $\boldsymbol{B}_\mathrm{er} = \boldsymbol{V}^\mathrm{T}\boldsymbol{B}_\mathrm{e}$, and $\boldsymbol{C}_\mathrm{er} = \boldsymbol{C}_\mathrm{e}\boldsymbol{V}$ is never simulated. Instead, the calculated ansatz space $\mathcal{V} = \mathrm{span}(\boldsymbol{V})$ is used to calculate the reduced nonlinear equations of motion for one body,

$$\begin{bmatrix} \boldsymbol{M}_\mathrm{fr}(\boldsymbol{q}_\mathrm{r}) & \boldsymbol{M}_\mathrm{efr}^\mathrm{T}(\boldsymbol{q}_\mathrm{r}) \\ \boldsymbol{M}_\mathrm{efr}(\boldsymbol{q}_\mathrm{r}) & \boldsymbol{V}^\mathrm{T}\boldsymbol{M}_\mathrm{e}\boldsymbol{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \ddot{\boldsymbol{q}}_\mathrm{r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{V}^\mathrm{T}\boldsymbol{K}_\mathrm{e}\boldsymbol{V}\boldsymbol{q}_\mathrm{r} + \boldsymbol{V}^\mathrm{T}\boldsymbol{D}_\mathrm{e}\boldsymbol{V}\dot{\boldsymbol{q}}_\mathrm{r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_\mathrm{fr}(\boldsymbol{q}_\mathrm{r}, \dot{\boldsymbol{q}}_\mathrm{r}) \\ \boldsymbol{h}_\mathrm{er}(\boldsymbol{q}_\mathrm{r}, \dot{\boldsymbol{q}}_\mathrm{r}) \end{bmatrix}. \tag{2.23}$$

It is worth mentioning that by this MOR procedure a perfect hyper-reduction is achieved, i. e., the nonlinear terms in (2.23) only depend on the reduced quantities. All quantities labeled $[\cdot](\boldsymbol{q}_\mathrm{r}, \dot{\boldsymbol{q}}_\mathrm{r})$ depend linearly or quadratically on the reduced elastic quantities. Different from many other nonlinear MOR procedures, no additional hyper-reduction scheme (see Section 2.5 of this chapter of this volume and Chapter 5 of Volume 2 of *Model order reduction*) like the discrete empirical interpolation method from [22] is necessary.

The quantity parts of (2.23), e. g., $\boldsymbol{h}_\mathrm{fr}(\boldsymbol{q}_\mathrm{r}, \dot{\boldsymbol{q}}_\mathrm{r})$ or $\boldsymbol{h}_\mathrm{er}(\boldsymbol{q}_\mathrm{r}, \dot{\boldsymbol{q}}_\mathrm{r})$, are called mass invariants and can be calculated prior to the simulation (see, e. g., [105]). Therefore, no back projection into the original high-dimensional space is necessary during the simulation, which is one benefit of EMBS simulations.

The described procedure is applied per body, which allows for a modular setup. The single reduced or rigid bodies are later coupled using a minimal coordinate approach or Lagrange multipliers (see, e. g., [105]).

As shown in Figure 2.17 for the crank drive example, only the piston rod is considered flexible. In other settings (see, e. g., [84]), other parts of the crank drive, e. g., the crankshaft are analyzed. As mentioned in [84], elastic effects and the chosen reduction method are important for realistic simulations for a crank drive system. Due

to the fact that the input/output behavior is essential, special care needs to be taken to approximate the interfaces in a correct way in the modeling and reduction process. Nevertheless, we know that every approximation introduces an error. Therefore, we are especially concerned to find measures for the evaluation of the error.

In the next section, a short explanation about the utilized reduction methods is given. Afterward, the error in the frequency and time domains is analyzed. Furthermore, we will also mention some possible error estimators and error bounds. We will finish the section with an outlook.

2.4.1 Used MOR methods

Since the piston rod undergoes only small deformations, it is modeled as a linear system. Several linear reduction techniques can be used to approximate the elastic coordinates of the piston rod, e. g.:

- Krylov method, i. e., matching the moments of the transfer function at defined frequencies up to defined orders [42];
- Craig–Bampton, i. e., static interface constraint modes in combination with fixed interface normal modes;
- CMS-Gram, i. e., frequency-weighted balanced truncation for the internal dynamics together with static correction modes [52];
- POD-Gram, i. e., balanced truncation with POD approximated frequency-weighted Gramian matrices [36].

All these methods have in common that they can be tuned for specific loading scenarios, e. g., by specifying frequencies in the Krylov method. In the previous section some information about the Craig–Bampton and CMS-Gram methods is given. Furthermore, the same work flow is used. For MOR of EMBS a correct consideration of boundary conditions and static correctness are essential steps; therefore, CMS-based or interpolation-based methods are favored. For systems with many connection points, e. g., gearboxes, the Krylov reduction or the CMS-based reduction turns out to be challenging. Therefore, an interface reduction is necessary very often in a first step. Here a model-based interface reduction is performed by inserting rigid connection elements (RBE2) at the two bearings of the piston rod (Figure 2.18). The slave nodes (nodes at the bearing seats) and the master node (the central node) behave like a rigid body.

Due to the boundary conditions, the connecting points at the piston rod can only move in the *yz*-plane. Therefore, only the inertia forces in these directions are considered as excitations in the CMS-Gram method and collocated outputs $\boldsymbol{C}_\mathrm{e} = \boldsymbol{B}_\mathrm{e}^\mathrm{T}$ are considered. The six static modes of the interface node between crankshaft and piston rod correspond to the rigid body modes. For the CMS-Gram approach, the frequency range of interest is $\mathcal{I}_f = [0\,\mathrm{kHz}, 8\,\mathrm{kHz}]$.

2.4.1.1 Results in the frequency domain

We are interested in the elastic movements; therefore, we only analyze the nonrigid body movement, which is the elastic dynamics of the system. This neglects the error due to coupling of the rigid body movement and internal dynamics. In a first step the separation between interface and internal nodes is conducted by a CMS-based approach, and later the Gramian matrix is used to approximate only the internal dynamics, which belongs to the internal nodes. In Figure 2.19, the transfer function of the unreduced elastic dynamic of the piston rod is plotted. The error system $\boldsymbol{G}_\mathrm{e} = \boldsymbol{G} - \bar{\boldsymbol{G}}_\mathrm{r}$ with the reduced second-order system $\bar{\boldsymbol{G}}_\mathrm{r}(s) = \boldsymbol{C}_{\mathrm{er}}(s^2\boldsymbol{M}_{\mathrm{er}} + s\boldsymbol{D}_{\mathrm{er}} + \boldsymbol{K}_{\mathrm{er}})^{-1}\boldsymbol{B}_{\mathrm{er}}$ is used to evaluate different reduction methods. As in Section 2.3, the relative error

$$\epsilon_\mathrm{r}(s = \mathrm{i}\omega) = \|\boldsymbol{G}(\omega) - \bar{\boldsymbol{G}}_\mathrm{r}(\omega)\|_\mathrm{Fro} / \|\boldsymbol{G}(\omega)\|_\mathrm{Fro} \tag{2.24}$$

over the angular frequency ω measured in the Frobenius norm of models reduced with different reduction methods is plotted in Figure 2.20. The interesting frequency range between 0 kHz and 20 kHz is equidistantly sampled with 500 points.

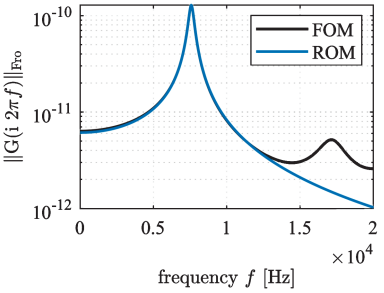


Figure 2.19: Norm of the transfer function of the unreduced ($N = 7332$) internal dynamic (black) and with CMS-Gram ($n = 6$) reduced internal dynamic (blue).

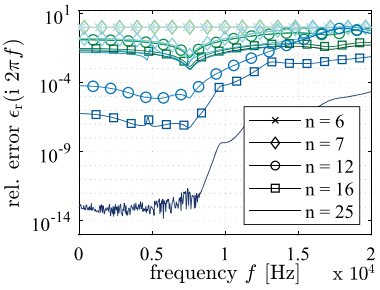


Figure 2.20: Different relative errors for different reduction methods and different reduction levels. Blue: CMS-Gram; green: classical Craig–Bampton. Lighter colors label models with smaller reduction size.

It can be seen that the relative error for the CMS-Gram-based reductions is smaller than the error of the Craig–Bampton reductions.

If we compare the CMS-based approach with reduction not based on a separation between interface and internal nodes, e. g., modal or POD-Gram reduction, a far worse approximation of the elastic/internal dynamics is achieved even with higher dimensions. The eigenvectors of the free system have a very low correlation to the internal dynamics. The generalized inertia forces do not excite these eigenmodes. Therefore, they are unimportant for the problem at hand. The POD-based reduction is not tuned

to approximate the internal dynamics. Therefore, no fair comparison is possible and we decided not to show these results.

Typically, the error is measured in specific norms, e. g., the $\mathcal{H}_\infty$- or the $\mathcal{H}_2$-norm [85]. For Gramian-based reductions, an error bound based on the sum of neglected Hankel singular values (HSVs) is available.

In Figure 2.21, the relative error in the $\mathcal{H}_2$-norm of the internal dynamics is plotted. For a reduction size of 6, only static modes are used; therefore, the CMS-Gram and the Craig–Bampton approach behave exactly the same. Already with one more reduction mode, the CMS-Gram approach shows better results. After 12 and 16 modes, we again see a big improvement (Figure 2.21). The Craig–Bampton reduction has a step-like decay, some eigenmodes of the bounded system have only a minor influence on the approximation quality. Therefore, after we achieved a reduction size of 14 no further improvements were made with the Craig–Bampton approach. In comparison to the Craig–Bampton reduction, the CMS-Gram-based reduction explicitly considers the generalized inertia forces, and therefore has a more steady and rapid decay of the error and the HSVs. A more elaborate description of this behavior is given by [51] for a slightly different crank drive example.

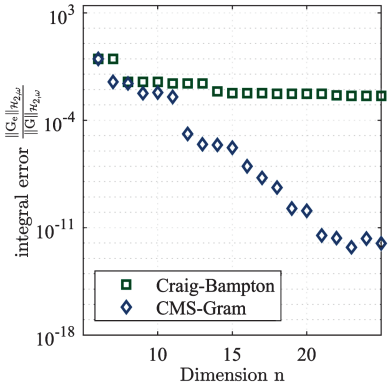


Figure 2.21: $\mathcal{H}_2$-error of the internal dynamics.

We saw that Gramian-based approximation in combination with static condensations is favorable. However, for a large-scale system, a direct calculation of the Gramian matrices is not possible. Two of the many approaches to calculate Gramian matrix-based reduction spaces are (i) the POD-based approximation of the frequency-weighted Gramian matrices and (ii) a two-step approach in which a Krylov-based approach calculates an intermediate size model in a first step. The reduction quality of the POD approach depends on the location of the frequency snapshots and the Krylov-based reduction depends on the location of the expansion points. The intermediate models

need to guarantee an acceptable approximation quality; however, due to their large size we cannot calculate the $\mathcal{H}_\infty$-norm, the $\mathcal{H}_2$-norm, or ϵ_r.

2.4.1.2 Error estimators in the frequency domain

As mentioned in [33] and [36], error estimators can be used to gradually select good positions of the frequency snapshots – respectively expansion points – at the most suitable position and estimate the error. Different error estimators for a given frequency range $[\omega_{\min},\ \omega_{\max}]$ are available to replace the time needed to evaluate the original system in calculating the relative error ϵ_r defined in (2.24).

One possibility introduced by [42] is the replacement with a second reduced system $\check{\boldsymbol{G}}$. The approximation error is then estimated by the relative error between the two reduced-order systems (see, e. g., [42]):

$$\epsilon_{\mathrm{Grimme}}(\omega) = \left\|\check{\boldsymbol{G}}(\omega) - \bar{\boldsymbol{G}}(\omega)\right\|_{\mathrm{Fro}} / \left\|\check{\boldsymbol{G}}(\omega)\right\|_{\mathrm{Fro}}. \tag{2.25}$$

A second error estimator was introduced by [13]: Instead of using a second ROM, the ROM from the previous iteration $\bar{\boldsymbol{G}}_{k-1}$ is used:

$$\epsilon_{\mathrm{BRK}}(f) = \left\|\bar{\boldsymbol{G}}_k(\mathrm{i}2\pi f) - \bar{\boldsymbol{G}}_{k-1}(\mathrm{i}2\pi f)\right\|_{\mathrm{Fro}} / \left\|\bar{\boldsymbol{G}}_k(\mathrm{i}2\pi f)\right\|_{\mathrm{Fro}}. \tag{2.26}$$

In the provable error estimator of [62], the error is separated into two complementary subspaces $\mathcal{S}_\Phi$ and $\mathcal{S}_\perp$ of the eigenmodes in the interesting frequency range. This error estimator has a provable error bound for lossless systems in addition to an extension to second-order systems (see, e. g., [33]):

$$\epsilon_{\mathrm{KFD}}(\omega) = V_2(R, \eta, \xi_{p_{\max}}) \left\|\boldsymbol{K}_c \boldsymbol{B}_\mathrm{e} - \boldsymbol{K}_\omega(\omega) \boldsymbol{V} \mathbb{L}_\mathrm{r}^{-1}(\mathrm{i}\omega) \boldsymbol{B}_{\mathrm{e\,r}}\right\|. \tag{2.27}$$

The error can be calculated efficiently by an offline-online decomposition. The term $\boldsymbol{K}_c$ is independent of ω and $\boldsymbol{V}$, the term $\boldsymbol{K}_\omega(\omega)$ is independent of the reduction matrix $\boldsymbol{V}$, the term $\mathbb{L}_\mathrm{r}^{-1}$ is based on reduced quantities and is further explained in [33], and the term $V_2(R, \eta, \xi_{p_{\max}})$ is a maximum bound of the magnification function of a second-order elementary vibrating system well known in linear vibrations theory (see, e. g., [79]).

In Figure 2.22, the different error estimators are plotted. These error estimators could, e. g., be used in a greedy-based selection process to add new expansion points for a Krylov subspace used for reduction or in a POD-like approximation of the second-order frequency-weighted Gramian matrices [36]. For this example, the error estimator ϵ_{KFD} has a slightly different behavior from the other error estimators but both share the same form.

Mechanical systems are not very sensitive to the location of the expansion points as other systems may be, especially in this example, where there is not much dynamic

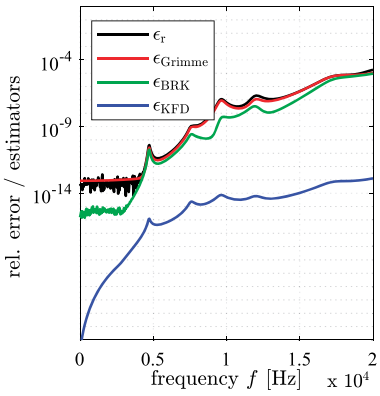


Figure 2.22: Comparison of the three error estimators and the relative error.

in the interesting frequency range. Therefore, all of the error estimators are helpful in automating the reduction process. Even if the absolute values of the error estimators are not consistent with the real error, the frequency location of maximum values and the convergence behavior of the error estimators are consistent with the real error.

2.4.1.3 Results in the time domain

In the time domain, we compare the well-established Craig–Bampton method with a mediocre basis size of 25 to the other reduction methods with basis size of only 7 (Figure 2.23). The Craig–Bampton model of size 25 is considered as a converged ROM. For the results an explicit first-order Runge–Kutta solver with a fifth-order automatic time step size control for stiff ODEs (MATLAB ode15s) is used. Modal reduction shows a phase shift and a larger error at the time of highest deformation. Despite having no phase shift, the Craig–Bampton method with only seven basis vectors shows bad conformance with the reference solution at the zoom-in view due to the small basis size. Only CMS-Gram accomplishes almost no error compared to the reference solution with a basis size of only 7.

2.4.1.4 Error estimators in the time domain

Error bounds based on the residual

$$\boldsymbol{R}_{\mathrm{m}}(t) = \boldsymbol{M}_{\mathrm{e}}\boldsymbol{V}\ddot{\boldsymbol{q}}_{\mathrm{r}} + \boldsymbol{D}_{\mathrm{e}}\boldsymbol{V}\dot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{K}_{\mathrm{e}}\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}}(t) - \boldsymbol{B}_{\mathrm{e}}\boldsymbol{u}_{\mathrm{e}}(t) \tag{2.28}$$

between the reduced and the original model can be used to deliver a posteriori error bounds in the time domain, which account for the current excitation. The general, a posteriori error estimator $\tilde{\Delta}$ of [99] for second-order mechanical systems is used in

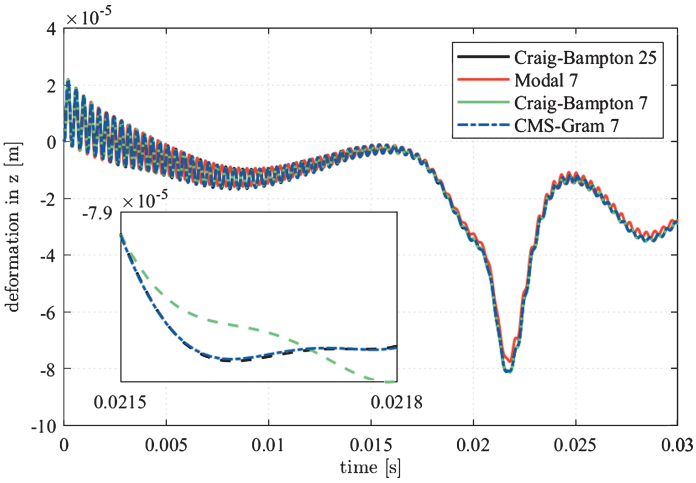


Figure 2.23: Deformation of one node in the *z*-direction for various reduction methods. The zoom-in view shows the time of the largest deformation.

the following. It has been analyzed in [37]. We apply this error estimator to the single, clamped piston rod excited with an approximated gas force. The true error is compared to the error estimator in Figure 2.24 for different model reduction methods.

It is of no surprise that modal reduction with 100 modes produces the smallest error. Looking at reductions with smaller basis size, Krylov 6 seems to have the best results. While all other methods have a practically indistinguishable error bound, the error estimator of Krylov 6 is two orders lower. This phenomenon still needs to be investigated.

2.4.2 Outlook

For EMBS simulations, MOR is one essential step to create fast-to-calculate but still convincing simulation models. If there are many inputs to the system, the interface reduction process often plays a far more critical role than the used MOR schemes. Error estimators in the frequency domain are helpful in automating the reduction process. Error estimation in the time domain with a priori error bounds for coupled multibody systems is a nontrivial task since coupling terms influence the behavior of a single part. New strategies such as to rewrite the multibody system as DAEs to consider all reaction forces as inputs need to be developed and rigorously tested.

With the CMS-Gram methods, a method is at hand which combines the benefit of static correctness, with an error-based, Gramian matrix-based approximation of the internal dynamics. Furthermore, the generalized inertia forces are considered in the reduction process. Good results are achieved in the frequency as well as in the time domain.

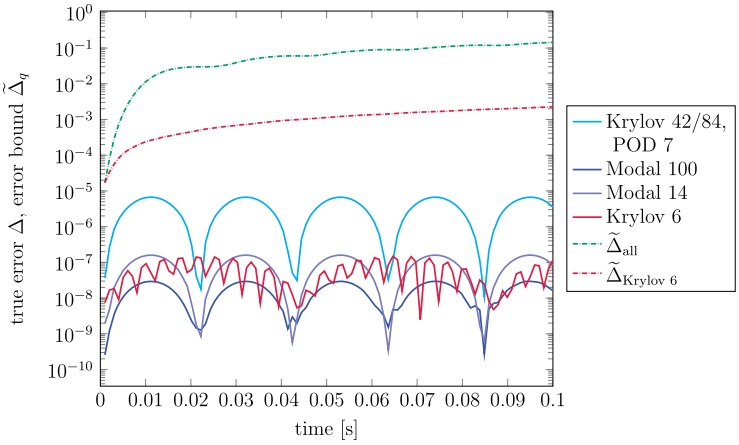


Figure 2.24: True error $\Delta(t) = \|\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}}(t) - \boldsymbol{q}(t)\|_2$ in solid lines and the error estimator $\tilde{\Delta}_q$ of the state in dashed lines of a clamped piston rod excited with an approximated gas force; $\tilde{\Delta}_{\text{all}}$ represents the error estimator of all methods except Krylov 6.

2.5 Nonlinear model order reduction for a leaf spring model

2.5.1 Nonlinear model reduction in structural dynamics

In engineering, one finds many applications where exact knowledge of the dynamic behavior of the structures during operation is essential. This is mainly due to the fact that the dynamics influence design goals such as fatigue, vibration comfort, or noise emission. In this section, we present the MOR of the leaf spring of a truck. This part undergoes large deformations in certain maneuvers like strong braking and exhibits hence geometrically nonlinear behavior. Since the transfer path of the excitation force goes through the spring, an accurate model of the nonlinear spring is crucial for the dynamics assessment of the otherwise linear chassis model.

The semi-discretized equations of motion from finite element models that are able to describe the dynamics of structures with large deflections are described by equation (2.2). To reduce the computational effort, one needs to perform two steps. First, a Galerkin projection is carried out in order to reduce the number of unknowns. Assuming a linear viscous damping matrix, this yields

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{V}\ddot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\dot{\boldsymbol{q}}_{\mathrm{r}}(t) + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}}(t)) = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{F}(t). \tag{2.29}$$

This approximation is similar to the projection methods used in linear model reduction for structural dynamics as described in Section 2.1. However, this reduction step alone does not reduce computation time significantly. The solution time for the leaf spring model, which will be shown below, can only be sped up 1.6 times by this

step, despite the fact that the solution vector has been reduced from 200,000 degrees of freedom to 100. The reason for this is the nonlinear restoring force term $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}})$, which is evaluated by an assembly of all element forces. Unlike in the reduction of linear systems, one cannot build a reduced matrix since the restoring forces are nonlinear. Hence, the evaluation of the nonlinear term is the new bottleneck of the solution process. The methods reducing the evaluation costs of this nonlinear term are called "hyper-reduction" methods. The application of a hyper-reduction is the second step that needs to be performed. Different hyper-reduction methods can be found in the literature. One prominent example is the discrete empirical interpolation method [22]. Other approaches are more suitable for nonlinear structural dynamics, such as the energy-conserving sampling and weighting (ECSW) method [31, 30] and polynomial expansion [50], which will be used here.

2.5.2 Basis generation

One approach to get a reduction basis is the so-called POD. It is based on the singular value decomposition $\boldsymbol{q}_\tau = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^{\mathrm{T}}$ of some training displacements $\boldsymbol{q}_\tau$, which usually come from results of a time integration of the full model. The reduction basis $\boldsymbol{V}$ is then built by stacking the first n left singular vectors into $\boldsymbol{V}$ so that $\boldsymbol{V}$ consists of the first n columns of $\boldsymbol{U}$.

Another approach generates the reduction matrix from two parts,

$$\boldsymbol{V} = [\boldsymbol{V}_{\mathrm{lin}} \quad \boldsymbol{V}_{\mathrm{nl}}], \tag{2.30}$$

where $\boldsymbol{V}_{\mathrm{lin}}$ contains some vibration modes $\boldsymbol{\phi}$ of the linearized system and $\boldsymbol{V}_{\mathrm{nl}}$ contains some modal derivatives $\boldsymbol{v}_{ij}$. The modal derivatives describe how the mode $\boldsymbol{\phi}_i$ changes if the system is perturbed in the direction of another mode $\boldsymbol{\phi}_j$. A slightly modified version of the modal derivatives are the static modal derivatives (SMD) [107, 57] that are calculated by solving

$$\boldsymbol{K}\boldsymbol{v}_{ij} = -\nabla_{\boldsymbol{\phi}_j}\boldsymbol{K}(\boldsymbol{q})\boldsymbol{\phi}_i, \tag{2.31}$$

which often perform better than the modal derivatives.

2.5.3 Hyper-reduction

2.5.3.1 Polynomial expansion

When linear materials are used, the internal force taking into account geometric nonlinear effects due to large deformations and rotations can be written using Einstein summation convention as

$$\boldsymbol{f}_i(\boldsymbol{q}) = \boldsymbol{K}^{(1)}_{ij}\boldsymbol{q}_j + \boldsymbol{K}^{(2)}_{ijk}\,\boldsymbol{q}_j\boldsymbol{q}_k + \boldsymbol{K}^{(3)}_{ijkl}\,\boldsymbol{q}_j\boldsymbol{q}_k\boldsymbol{q}_l. \tag{2.32}$$

Thus, the nonlinear restoring force vector can be described by three tensors $\boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)}$, and $\boldsymbol{K}^{(3)}$.

Applying a Galerkin projection with the reduction basis $\boldsymbol{V}$, one gets

$$(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}}))_i = \bar{\boldsymbol{K}}^{(1)}_{ij}\,\boldsymbol{q}_{\mathrm{r},j} + \bar{\boldsymbol{K}}^{(2)}_{ijk}\,\boldsymbol{q}_{\mathrm{r},j}\boldsymbol{q}_{\mathrm{r},k} + \bar{\boldsymbol{K}}^{(3)}_{ijkl}\,\boldsymbol{q}_{\mathrm{r},j}\boldsymbol{q}_{\mathrm{r},k}\boldsymbol{q}_{\mathrm{r},l}, \tag{2.33}$$

with

$$\begin{aligned}
\bar{\boldsymbol{K}}^{(1)}_{ij} &= (\boldsymbol{V}^{\mathrm{T}})_{ik}\,\boldsymbol{K}^{(1)}_{kl}\,\boldsymbol{V}_{lj},\\
\bar{\boldsymbol{K}}^{(2)}_{ijk} &= (\boldsymbol{V}^{\mathrm{T}})_{il}\,\boldsymbol{K}^{(2)}_{lmn}\,\boldsymbol{V}_{mj}\boldsymbol{V}_{nk},\\
\bar{\boldsymbol{K}}^{(3)}_{ijkl} &= (\boldsymbol{V}^{\mathrm{T}})_{im}\,\boldsymbol{K}^{(3)}_{mnop}\,\boldsymbol{V}_{nj}\boldsymbol{V}_{ok}\boldsymbol{V}_{pl}.
\end{aligned}$$

Therefore, one only needs to identify $\bar{\boldsymbol{K}}^{(1)}, \bar{\boldsymbol{K}}^{(2)}$, and $\bar{\boldsymbol{K}}^{(3)}$ to get a reduced model whose nonlinear term can be evaluated very fast. For the computation of these three tensors, several techniques exist which can be classified as intrusive and nonintrusive methods. Intrusive methods identify the tensors by computing the coefficients on element level [92, 106] requiring access to the element formulation inside the finite element code. Nonintrusive methods, however, do not require the element formulation and can hence be used with commercial finite element software, where the access to internal computations is limited. Some techniques identify the tensors by prescribing displacements and evaluating the resulting nonlinear forces [80, 61], others by prescribing forces and evaluating displacements [74]. The implicit condensation and expansion method [50] computes both polynomial tensors and the reduced basis in one step. In our case study, we use the nonintrusive identification as proposed in [101, 86, 87], which uses multiple evaluations of the reduced tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{q})$ at different displacements.

2.5.3.2 Energy-conserving sampling and weighting

Another approach is the ECSW method [31, 30], which is based on a reduced assembly of a subset of the elements and extrapolates their contribution to the full force vector:

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}_{\mathrm{r}}) = \sum_{e\in E} \boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}_e^{\mathrm{T}}\boldsymbol{f}_e(\boldsymbol{L}_e\boldsymbol{V}_e\boldsymbol{q}_{\mathrm{r}}) \approx \sum_{e\in\tilde{E}\subset E} \xi_e \boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}_e^{\mathrm{T}}\boldsymbol{f}_e(\boldsymbol{L}_e\boldsymbol{V}_e\boldsymbol{q}_{\mathrm{r}}). \tag{2.34}$$

The matrix $\boldsymbol{L}_e$ is a Boolean mapping matrix from the local degrees of freedom of element e to the global degrees of freedom and ξ_e are positive weights for extrapolation. The weights ξ_e and the element subset $\tilde{E}$ are chosen by using training displacements for which the virtual work of the restoring force in the direction of the reduction basis shall be retained in the hyper-reduced model. As in the POD, these training displacements are often computed from simulations of the high-dimensional model. Another approach to gain training displacements is presented in the next section, since performing full simulations is numerically expensive.

2.5.3.3 Nonlinear stochastic Krylov training sets for ECSW

An alternative approach to build training vectors for ECSW, which avoids the full simulations, is called nonlinear stochastic Krylov training sets (NSKTS), proposed in [100]. The idea of this method is to build a subspace whose vectors are able to approximate the nonlinear force vector $\boldsymbol{f}$. If the viscous damping term $\boldsymbol{D}$ is neglected, one finds

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{f}(\boldsymbol{q}(t)) = \boldsymbol{B}\boldsymbol{F}(t) \rightsquigarrow \boldsymbol{f} \in \operatorname{span}(\boldsymbol{B}, \boldsymbol{M}\ddot{\boldsymbol{q}}(t_1), \boldsymbol{M}\ddot{\boldsymbol{q}}(t_2), \ldots, \boldsymbol{M}\ddot{\boldsymbol{q}}(t_n)).$$

As the accelerations $\ddot{\boldsymbol{q}}(t_i)$ are unknown, an approximation for the subspace described above must be made. In the NSKTS, the subspace is approximated with the Krylov subspace

$$\boldsymbol{F}_{\text{kry}} = \operatorname{span}\{\boldsymbol{B}, \boldsymbol{M}\boldsymbol{K}^{-1}\boldsymbol{B}, (\boldsymbol{M}\boldsymbol{K}^{-1})^2\boldsymbol{B}, \ldots\} = \mathcal{K}(\boldsymbol{M}\boldsymbol{K}^{-1}, \boldsymbol{B}), \tag{2.35}$$

which is orthogonalized and normalized so that $\boldsymbol{F}_{\text{kry}}^{\text{T}}\boldsymbol{K}^{-1}\boldsymbol{F}_{\text{kry}} = \mathbf{I}$.

Then, some random vectors $\boldsymbol{f}_{\text{NSKTS}}^{\tau}$ living in this Krylov subspace are generated and applied as external force to the nonlinear static problem

$$\boldsymbol{f}(\boldsymbol{q}_{\tau}^{(k)}) = k\boldsymbol{f}_{\text{NSKTS}}^{\tau}, \quad k \in (0, 1].$$

This equation must be solved by a nonlinear solver such as Newton–Raphson. The training set is then built by saving the solution $\boldsymbol{q}_{\tau}^{(k)}$ for some load steps k for each external force $\boldsymbol{f}_{\text{NSKTS}}^{\tau}$ as training vector. This procedure reduces the computation cost for obtaining a set of training vectors by avoiding direct time integration of the full model. All steps can also be carried out in the reduced subspace spanned by $\boldsymbol{V}$. This has the advantage that the nonlinear static problems can be solved faster and the resulting training vectors live in the subspace of the displacements for the ROM [100]. There are no a priori error estimates for the NSKTS method and, therefore, the necessary number of training vectors is evaluated by varying its number and checking the convergence of the solution.

2.5.4 Case study: leaf spring

In the following, the results of the reduction of a truck chassis leaf spring are summarized. The case study is carried out with the Finite Element Package AMfe, developed by the Chair of Applied Mechanics at Technical University of Munich. The code is available at https://github.com/AppliedMechanics/AMfe. The full study is published in [101].

Figure 2.25 shows the leaf spring, which consists of two leafs (top and bottom), a central fixture that joins the leafs, and two rubber pads that keep the distance between the leafs at the ends. The front eye is fixed with the frame via a joint allowing

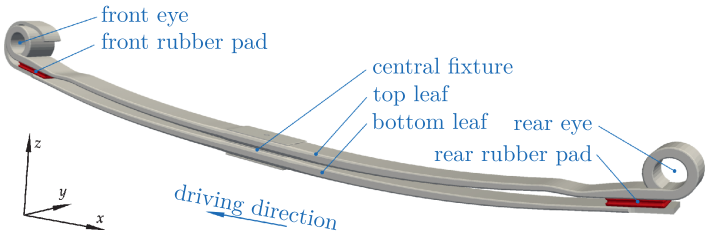


Figure 2.25: Leaf spring obtained from [101] consisting of two leafs, rubber pads, and central fixture.

rotations about the y-axis while the rear eye allows both rotations about the y-axis and a translation in the x-direction. The rubber pads are fixed with the top leaf and have a sliding contact with the lower leaf. The model is meshed with 85,762 linear elements (tetrahedrons and hexahedrons). A load case is applied that stems from a multibody simulation of a brake maneuver. Time-varying forces and loads are applied on the top of the central fixture.

We compare the simulation time and accuracies of a simulation with 1,500 time steps carried out with a generalized α scheme ($\rho_\infty = 0.8$). As accuracy measure we define the relative error

$$\mathrm{RE} = \frac{\sqrt{\sum_i \Delta \boldsymbol{q}(t_i)^\mathrm{T} \Delta \boldsymbol{q}(t_i)}}{\sqrt{\sum_i \boldsymbol{q}_\mathrm{ref}(t_i)^\mathrm{T} \boldsymbol{q}_\mathrm{ref}(t_i)}} \cdot 100\,\% \quad \text{with } \Delta \boldsymbol{q}(t_i) = \boldsymbol{q}(t_i) - \boldsymbol{q}_\mathrm{ref}(t_i). \tag{2.36}$$

We compare two kinds of errors: First, the error of the Galerkin projection, where the non-hyper-reduced but projected system is compared with the full solution (RE_f), and, second, the error of the hyper-reduction, where the hyper-reduced solution is compared to the non-hyper-reduced but projected solution (RE_{hr}).

Table 2.2 lists the relative errors and simulation times for different simulations. The wall time for a full simulation run with 216,499 degrees of freedom is 40,022 s ($\approx$ 11 h). As reduction basis, 100 basis vectors consisting of 40 vibration modes and 60 static modal derivatives are chosen. One can see that only a small speedup of 1.66 is gained while the error is quite small. To further speed up the simulation, hyper-reduction is needed.

Therefore, two hyper-reduction methods are considered. First, the ECSW is carried out with 160 nonlinear stochastic Krylov training sets. The hyper-reduced mesh is shown in Figure 2.26. Only 816 elements are evaluated while the relative error of the hyper-reduction is 0.13 %. The hyper-reduced model gains a speedup of 38.23 compared to the full simulation. Second, the polynomial expansion is carried out. The identification of the tensor coefficients needs 28,856 s ($\approx$ 8 h), but the hyper-reduced model has a wall time of just 113 s, which is a speedup of 354.18 compared to the full simulation. Since the model has a linear elastic material and the nonlinear strain measure is quadratic in $\boldsymbol{q}$, the cubic polynomial expansion of the internal restoring force

Table 2.2: Errors and speedup rates of the different reduction techniques for the leaf spring example obtained from [101].

Reduction method	DOFs	Elements	RE_f [%]	RE_{hr} [%]	t_w [s]	Speedup [–]
full	216,499	85,762	–	–	40,022	–
Modes & SMDs	100	85,762	0.68	–	24,127	1.66
Modes & SMDs + ECSW with NSKTS	100	816	0.78	0.13	1 047	38.23
Modes & SMDs + polynomial Exp.	100	–	0.68	0.00	113	354.18

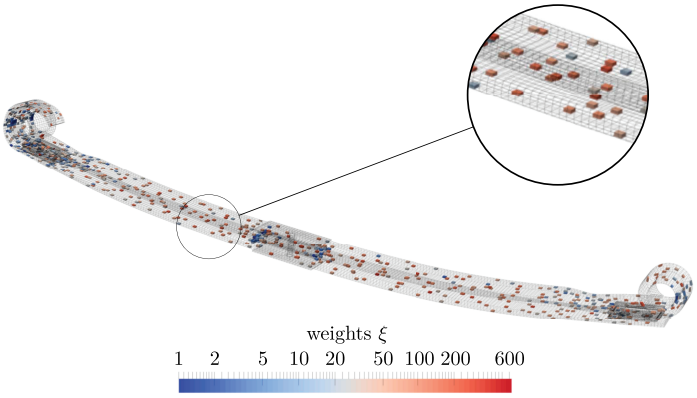


Figure 2.26: Hyper-reduced mesh obtained from [101]. The reduced mesh consists of 816 elements that have different weights ξ.

(2.32) is exact. Hence, the relative error of the polynomial expansion hyper-reduction method is zero in this case.

2.5.5 Conclusion

The geometric nonlinear leaf spring has been reduced by a reduction basis of dimension 100. The reduction basis has been computed by using the properties of the system without the need for results from full dynamic simulations. The combination of 40 vibration modes and 60 static derivatives leads to accurate results although the full model has 216,499 degrees of freedom. Then, two different hyper-reduction techniques, ECSW and polynomial expansion, are carried out. Nonlinear stochastic Krylov training sets are used for ECSW which avoid full simulation runs and lead to relative small reduction costs of about 73 minutes. The ECSW-reduced model leads to a speedup factor of 38.23. The polynomial expansion gives the best speedup, which is about 10 times higher than with ECSW, while the reduction time is much higher (about 8 h). Thus, one can conclude that polynomial expansion is best suited if offline costs

do not matter or if one needs to run many simulations or many time steps with the reduced model. However, the polynomial (cubic) expansion is only valid for models with linear elastic materials. Another issue is the computational cost and memory demand for the identification of the polynomial coefficients. Both depend highly on the dimension n of the reduction basis, since they increase with $\mathcal{O}(n^4)$. In our test case, the reduction basis of dimension 100 was suitable. For models requiring a larger reduced basis, hyper-reduction using the ECSW is a good option.

Author contributions

M. Cruz Varona, C. Lerch, and B. Lohmann (Technical University of Munich) wrote Section 2.1. C. D. Yuan, E. B. Rudnyi, and T. Bechtold (University of Rostock) wrote Section 2.2. B. Fröhlich, P. Holzwarth, and P. Eberhard (University of Stuttgart) wrote Section 2.3. D. Grunert and J. Fehr (University of Stuttgart) wrote Section 2.4. And C. H. Meyer, J. B. Rutzmoser, and D. J. Rixen (Technical University of Munich) wrote Section 2.5.

Bibliography

[1] O. P. Agrawal and A. A. Shabana, Dynamic analysis of multibody systems using component modes, *Comput. Struct.*, **21** (6) (1985), 1303–1312.

[2] D. Amsallem and C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, *AIAA J.*, **46** (7) (2008), 1803–1813.

[3] D. Amsallem, M. Zahr, Y. Choi, and C. Farhat, Design optimization using hyper-reduced-order models, *Struct. Multidiscip. Optim.*, **51** (4) (2015), 919–940.

[4] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.

[5] C. Bach, I. C. Salazar, L. Song, J. Fender, and F. Duddeck, Nonlinear model order reduction of explicit finite element simulations for crash analysis, in *14th Int. Conf. on Comput. Plast.*, 2017.

[6] Z. Bai and Y. Su, Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method, *SIAM J. Sci. Comput.*, **26** (5) (2005), 1692–1709.

[7] B. Bandyopadhyay, S. Janardhanan, and V. Sreeram et al., Sliding mode control design via reduced order model approach, *Int. J. Autom. Comput.*, **4** (4) (2007), 329–334.

[8] U. Baur, P. Benner, and L. Feng, Model order reduction for linear and nonlinear systems: a system-theoretic perspective, *Arch. Comput. Methods Eng.*, **21** (4) (2014), 331–358.

[9] C. A. Beattie and S. Gugercin, Krylov-based model reduction of second-order systems with proportional damping, in *44th IEEE Conference on Decision and Control*, pp. 2278–2283, IEEE, 2005.

[10] C. A. Beattie and S. Gugercin, Model reduction by rational interpolation, *Model Reduct. Algorithms: Theory Appl.*, **15** (2017), 297–334.

[11] T. Bechtold, D. Hohlfeld, E. B. Rudnyi, and J. G. Korvink, Moment-matching-based linear model order reduction for nonparametric and parametric electrothermal MEMS models, in *System-Level Modeling of MEMS*, pp. 211–235, 2013.

[12] T. Bechtold, E. B. Rudnyi, and J. G. Korvink, Automatic generation of compact electro-thermal models for semiconductor devices, *IEICE Trans. Electron.*, **86** (3) (2003), 459–465.

[13] T. Bechtold, E. B. Rudnyi, and J. G. Korvink, Error indicators for fully automatic extraction of heat-transfer macromodels for MEMS, *J. Micromech. Microeng.*, **15** (3) (2005), 430–440.

[14] T. Bechtold, E. B. Rudnyi, and J. G. Korvink, *Fast simulation of electro-thermal MEMS: efficient dynamic compact models*, Springer, 2006.

[15] P. Benner and L. Feng, Model order reduction for coupled problems, *Appl. Comput. Math.*, **14** (1) (2015), 3–22.

[16] P. Benner, P. Kürschner, and J. Saak, An improved numerical method for balanced truncation for symmetric second-order systems, *Math. Comput. Model. Dyn. Syst.*, **19** (6) (2013), 593–615.

[17] P. Benner and J. Saak, Efficient balancing-based MOR for large-scale second-order systems, *Math. Comput. Model. Dyn. Syst.*, **17** (2) (2011), 123–143.

[18] B. Besselink, U. Tabak, A. Lutowska, N. Van De Wouw, H. Nijmeijer, D. Rixen, M. Hochstenbach, and W. Schilders, A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control, *J. Sound Vib.*, **332** (19) (2013), 4403–4422.

[19] B. Blockmans, T. Tamarozzi, F. Naets, and W. Desmet, A nonlinear parametric model reduction method for efficient gear contact simulations, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1162–1191.

[20] D. Bonvin and D. Mellichamp, A unified derivation and critical review of modal approaches to model reduction, *Int. J. Control*, **35** (5) (1982), 829–848.

[21] Y. Chahlaoui, K. A. Gallivan, A. Vandendorpe, and P. Van Dooren, Model reduction of second-order systems, in *Dimension Reduction of Large-Scale Systems*, pp. 149–172, Springer, 2005.

[22] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[23] M. R. Chidambara, Two simple techniques for the simplification of large dynamic systems, in *Joint Automatic Control Conference*, vol. 7, pp. 669–674, 1969.

[24] R. Craig and M. Bampton, Coupling of substructures for dynamic analyses, *AIAA J.*, **6** (7) (1968), 1313–1319.

[25] E. Davison, A method for simplifying linear dynamic systems, *IEEE Trans. Autom. Control*, **11** (1) (1966), 93–101.

[26] S. Deparis and G. Rozza, Reduced basis method for multi-parameter-dependent steady Navier–Stokes equations: applications to natural convection in a cavity, *J. Comput. Phys.*, **228** (12) (2009), 4359–4378.

[27] P. V. Dooren, K. Gallivan, and P.-A. Absil, $\mathcal{H}_2$-optimal model reduction of MIMO systems, *Appl. Math. Lett.*, **21** (12) (2008), 1267–1273.

[28] R. Eid, B. Salimbahrami, B. Lohmann, E. Rudnyi, and J. Korvink, Parametric order reduction of proportionally damped second-order systems, *Sens. Mater.*, **19** (3) (2007), 149–164.

[29] M. Ess, S. Weikert, K. Wegener, and J. Mayr, *Dynamic loads and thermal errors on machine tools. Technical report, Institute of Machine Tools and Manufacturing*, ETH Zurich, 2012.

[30] C. Farhat, P. Avery, T. Chapman, and J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *Int. J. Numer. Methods Eng.*, **98** (9) (2014), 625–662.

[31] C. Farhat, T. Chapman, and P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1077–1110.

[32] H. Faßbender and A. Soppa, Machine tool simulation based on reduced order FE models, *Math. Comput. Simul.*, **82** (2011), 404–413.
[33] J. Fehr, Automated and Error-Controlled Model Reduction in Elastic Multibody Systems, in *Dissertation, Institut für Technische und Numerische Mechanik der Universität Stuttgart*, vol. 21, Shaker Verlag, Aachen, 2011.
[34] J. Fehr and P. Eberhard, Error-controlled model reduction in flexible multibody dynamics, *J. Comput. Nonlinear Dyn.*, **5** (3) (2010), 031005.
[35] J. Fehr and P. Eberhard, Simulation process of flexible multibody systems with non-modal model order reduction techniques, *Multibody Syst. Dyn.*, **25** (3) (2011), 313–334.
[36] J. Fehr, M. Fischer, B. Haasdonk, and P. Eberhard, Greedy-based approximation of frequency-weighted Gramian matrices for model reduction in multibody dynamics, *Z. Angew. Math. Mech.*, **93** (8) (2012), 501–519.
[37] J. Fehr, D. Grunert, A. Bhatt, and B. Haasdonk, A sensitivity study of error estimation in reduced elastic multibody systems, in *9th Vienna International Conference on Mathematical Modelling, IFAC-PapersOnLine*, vol. 51(2), pp. 202–207, 2018.
[38] J. Fehr, D. Grunert, P. Holzwarth, B. Fröhlich, N. Walker, and P. Eberhard, Morembs – a Model Order Reduction Package for Elastic Multibody Systems and Beyond, in *Reduced-Order Modeling (ROM) for Simulation and Optimization*, pp. 141–166, Springer, 2018.
[39] R. Freund, Krylov-subspace methods for reduced-order modelling in circuit simulation, *J. Comput. Appl. Math.*, **123** (2000), 395–421.
[40] O. Föllinger, *Regelungstechnik:Einführung in die Methoden und ihre Anwendung*, VDE Verlag GmbH, 2016.
[41] M. Géradin and D. J. Rixen, *Mechanical Vibrations: Theory and Application to Structural Dynamics*, John Wiley & Sons, 2014.
[42] E. Grimme, *Krylov projection methods for model reduction*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.
[43] K. Großmann et al., *Thermo-energetic Design of Machine Tools*, Springer, 2016.
[44] S. Gugercin and A. C. Antoulas, A survey of model reduction by balanced truncation and some new results, *Int. J. Control*, **77** (8) (2004), 748–766.
[45] S. Gugercin, A. C. Antoulas, and C. Beattie, $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems, *SIAM J. Matrix Anal. Appl.*, **30** (2) (2008), 609–638.
[46] P. Guha and M. Nabi, Optimal control of a nonlinear induction heating system using a proper orthogonal decomposition based reduced order model, *J. Process Control*, **22** (9) (2012), 1681–1687.
[47] P. Guha and M. Nabi, Reduced order modeling of a microgripper using SVD-Second-Order Krylov Method, *Int. J. Comput. Methods Eng. Sci. Mech.*, **16** (2) (2015), 65–70.
[48] R. J. Guyan, Reduction of stiffness and mass matrices, *AIAA J.*, **3** (2) (1965), 380.
[49] G. H. K. Heirman, F. Naets, and W. Desmet, A system-level model reduction technique for the efficient simulation of flexible multibody systems, *Int. J. Numer. Methods Eng.*, **85** (3) (2011), 330–354.
[50] J. Hollkamp and R. Gordon, Reduced-order models for nonlinear response prediction: Implicit condensation and expansion, *J. Sound Vib.*, **318** (4–5) (2008), 1139–1153.
[51] P. Holzwarth, Modellordnungsreduktion für substrukturierte mechanische Systeme (in German), in *Dissertation, Institut für Technische und Numerische Mechanik der Universität Stuttgart*, vol. 51, Shaker Verlag, Aachen, 2017.
[52] P. Holzwarth and P. Eberhard, SVD-based improvements for component mode synthesis in elastic multibody systems, *Eur. J. Mech. A, Solids*, **49** (2015), 408–418.
[53] P. Holzwarth, N. Walker, and P. Eberhard, Interface reduction of linear mechanical systems with a modular setup, *Multibody Syst. Dyn.*, (2018), 1–19.

[54] S. Hu, C. Yuan, A. Castagnotto, B. Lohmann, S. Bouhedma, D. Hohlfeld, and T. Bechtold, Stable reduced order modeling of piezoelectric energy harvesting modules using implicit Schur complement, *Microelectron. Reliab.*, **85** (2018), 148–155.

[55] W. Hurty, Vibrations of structural systems by component mode synthesis, *J. Eng. Mech. Div.*, **86** (4) (1960), 51–70.

[56] W. Hurty, Dynamic analysis of structural systems using component modes, *AIAA J.*, **3** (4) (1965), 678–685.

[57] S. R. Idelsohn and A. Cardona, A reduction method for nonlinear structural dynamic analysis, *Comput. Methods Appl. Mech. Eng.*, **49** (3) (1985), 253–279.

[58] S. Janardhanan, Model order reduction and controller design techniques, 2005. https://www.researchgate.net/publication/236166577_Model_Order_Reduction_and_Controller_Design_Techniques.

[59] T. Kailath, *Linear systems*, Prentice-Hall information and system sciences series, Prentice-Hall, 1980.

[60] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, *Nonlinear Dyn.*, **41** (1-3) (2005), 147–169.

[61] K. Kim, A. Radu, X. Wang, and M. Mignolet, Nonlinear reduced order modeling of isotropic and functionally graded plates, *Int. J. Non-Linear Mech.*, **49** (2013), 100–110.

[62] Y. Konkel, O. Farle, and R. Dyczij-Edlinger, Ein Fehlerschätzer für die Krylov-Unterraum basierte Ordnungsreduktion zeit-harmonischer Anregungsprobleme, in B. Lohmann and A. Kugi (eds.), *Tagungsband GMA-Fachausschuss 1.30 "Modellbildung, Identifikation und Simulation in der Automatisierungstechnik"*, pp. 139–149, VDI/VDE-GMA, Automation and Control Institute (ACIN), Vienna University of Technology, 2008.

[63] K. Kunisch and S. Volkwein, Control of the Burgers Equation by a reduced-order approach using Proper Orthogonal Decomposition, *J. Optim. Theory Appl.*, **102** (2) (1999), 345–371.

[64] A. Kuppurajulu and S. Elangovan, Simplified power system models for dynamic stability studies. *IEEE Trans. Power Appar. Syst.*, **PAS-90** (1) (1971) 11–23.

[65] A. S. Kushwaha, A. B. Wankhade, D. E. Mahajan, and D. K. Thakur, Analysis of the ball bearing considering the thermal (temperature) and friction effects, *Int. J. Eng. Res. Appl.*, (2012), 115–120.

[66] S. Lamba and S. Rao, On suboptimal control via the simplified model of Davison, *IEEE Trans. Autom. Control*, **19** (4) (1974), 448–450.

[67] N. Lang, J. Saak, and P. Benner, Model order reduction for systems with moving loads, *Automatisierungstechnik*, **62** (7) (2014), 512–522.

[68] N. Lang, J. Saak, and P. Benner, Model order reduction for thermo-elastic assembly group models, in *Thermo-energetic Design of Machine Tools*, pp. 85–93, Springer, 2015.

[69] M. Lehner and P. Eberhard, A two-step approach for model reduction in flexible multibody dynamics, *Multibody Syst. Dyn.*, **17** (2–3) (2007), 157–176.

[70] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, Proper orthogonal decomposition and its applications—Part I: Theory, *J. Sound Vib.*, **252** (3) (2002), 527–544.

[71] L. Litz, *Reduktion der Ordnung linearer Zustandsraummodelle mittels modaler Verfahren*, Hochschulverlag, Stuttgart, 1979.

[72] D. Maier, *Moderne Reduktionsverfahren in der EMKS-Simulation von gekoppelten elastischen Körpern* (in German). Diploma thesis 199, Institut für Technische und Numerische Mechanik, Universität Stuttgart, 2012.

[73] S. Marshall, An approximate method for reducing the order of a linear system, *Int. J. Control*, **10** (102) (1966), 642–643.

[74] M. McEwan, J. Wright, J. Cooper, and A. Leung, A finite element/modal technique for nonlinear plate and stiffened panel response prediction, in *Collection of Technical Papers – AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 5, pp. 3061–3070, 2001.

[75] M. Meindl, M. Cruz Varona, A. Castagnotto, F. Thomann, W. Polifke, and B. Lohmann, Model order reduction in thermoacoustic stability analysis, in *9th Vienna International Conference on Mathematical Modelling*, 2018.

[76] D. G. Meyer and S. Srinivasan, Balancing and model reduction for second-order form linear systems, *IEEE Trans. Autom. Control*, **41** (11) (1996), 1632–1644.

[77] N. Mian, S. Fletcher, A. Longstaff, and A. Myers, Efficient estimation by FEA of machine tool distortion due to environmental temperature perturbations, *Precis. Eng.*, **37** (2) (2013), 372–379.

[78] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Trans. Autom. Control*, **26** (1) (1981), 17–32.

[79] P. Müller and W. Schiehlen, *Linear Vibrations*, Martinus Nijhoff Publishers, Dordrecht, 1985.

[80] A. Muravyov and S. Rizzi, Determination of nonlinear stiffness with application to random vibration of geometrically nonlinear structures, *Comput. Struct.*, **81** (15) (2003), 1513–1523.

[81] A. Naumann, N. Lang, M. Partzsch, M. Beitelschmidt, P. Benner, A. Voigt, and J. Wensch, Computation of thermo-elastic deformations on machine tools, a study of numerical methods, *Prod. Eng.*, **10** (3) (2016), 253–263.

[82] S. Nestmann, Mittel und Methoden zur Verbesserung des thermischen Verhaltens von Werkzeugmaschinen. *Mechatronik: Optimierungspotenzial der Werkzeugmaschine nutzen. IWB Seminarberichte*, 83, 2006.

[83] V. Nguyen, M. Buffoni, K. Willcox, and B. Khoo, Model reduction for reacting flow applications, *Int. J. Comput. Fluid Dyn.*, **28** (3–4) (2014), 91–105.

[84] C. Nowakowski, P. Kürschner, P. Eberhard, and P. Benner, Model reduction of an elastic crankshaft for elastic multibody simulations, *Z. Angew. Math. Mech.*, **93** (2012), 198–216.

[85] H. K. F. Panzer, *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*, Dissertation, Technische Universität München, Verlag Dr. Hut, München, 2014.

[86] R. Perez, X. Wang, and M. Mignolet, Nonintrusive structural dynamic reduced order modeling for large deformations: Enhancements for complex structures, *J. Comput. Nonlinear Dyn.*, **9** (2014), 3.

[87] G. Phlipot, X. Wang, M. P. Mignolet, L. Demasi, and R. Cavallaro, Reduced order modeling for the nonlinear geometric response of some joined wings, in *55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference*, p. 0151, 2014.

[88] G. Pitton, A. Quaini, and G. Rozza, Computational reduction strategies for the detection of steady bifurcations in incompressible fluid-dynamics: Applications to coanda effect in cardiology, *J. Comput. Phys.*, **344** (2017), 534–557.

[89] G. Pitton and G. Rozza, On the application of reduced basis methods to bifurcation problems in incompressible fluid dynamics, *J. Sci. Comput.*, **73** (1) (2017), 157–177.

[90] I. Pontes Duff, P. Vuillemin, C. Poussot-Vassal, C. Seren, and C. Briat, Stability and performance analysis of large-scale aircraft vibration delay model using model reduction techniques, in *Proceedings of the EuroGNC'15*, 2015.

[91] C. Poussot-Vassal and C. Roos, Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models, *Control Eng. Pract.*, **20** (9) (2012), 919–930.

[92] A. Przekop, M. Azzouz, X. Guo, C. Mei, and L. Azrar, Finite element multiple-mode approach to nonlinear free vibrations of shallow shells, *AIAA J.*, **42** (11) (2004), 2373–2381.

[93] A. Quarteroni and G. Rozza, Numerical solution of parametrized Navier–Stokes equations by reduced basis methods, *Numer. Methods Partial Differ. Equ.*, **23** (4) (2007), 923–948.
[94] J. W. Strutt and B. Rayleigh, *The theory of sound*, vol. 2, Dover, 1945.
[95] D. Rixen, A dual Craig-Bampton method for dynamic substructuring, *J. Comput. Appl. Math.*, **168** (1-2) (2004), 383–391.
[96] L. Roncarati, Model order reduction and system simulation of a machine tool for real-time compensation of thermally induced deformations, in *ANSYS Conference & 32. CADFEM Users' Meeting*. Nürnberg, 2014.
[97] G. Rozza, D. B. P. Huynh, and A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Arch. Comput. Methods Eng.*, **15** (3) (2007), 1.
[98] E. Rudnyi, MOR for ANSYS, in *System-level modeling of MEMS*, Advanced Micro and Nanosystems, pp. 425–438, Wiley–VCH, 2013.
[99] T. Ruiner, J. Fehr, B. Haasdonk, and P. Eberhard, A-posteriori error estimation for second order mechanical systems, *Acta Mech. Sin.*, **28** (3) (2012), 854–862.
[100] J. Rutzmoser and D. Rixen, A lean and efficient snapshot generation technique for the hyper-reduction of nonlinear structural dynamics, *Comput. Methods Appl. Mech. Eng.*, **325** (2017), 330–349.
[101] J. B. Rutzmoser, *Model Order Reduction for Nonlinear Structural Dynamics*. Dissertation, Technische Universität München, München, 2018.
[102] B. Salimbahrami and B. Lohmann, Order reduction of large scale second-order systems using Krylov subspace methods, *Linear Algebra Appl.*, **415** (2-3) (2006), 385–405.
[103] S. B. Salimbahrami, *Structure preserving order reduction of large scale second order models*. PhD thesis, Technische Universität München, 2005.
[104] D. Scheffold, C. Bach, F. Duddeck, G. Müller, and M. Buchschmid, Vibration frequency optimization of jointed structures with contact nonlinearities using hyper-reduction, *IFAC-PapersOnLine*, **51** (2) (2018), 843–848.
[105] R. Seifried, *Dynamics of Underactuated Multibody Systems – Modeling, Control and Optimal Design*, vol. 205, Springer, Berlin, 2014.
[106] Y. Shi and C. Mei, A finite element time domain modal formulation for large amplitude free vibrations of beams and plates, *J. Sound Vib.*, **193** (2) (1996), 453–463.
[107] P. Slaats, J. De Jongh, and A. Sauren, Model reduction tools for nonlinear structural dynamics, *Comput. Struct.*, **54** (6) (1995), 1155–1171.
[108] T. Stykel and T. Reis, Balanced truncation model reduction of second-order systems, *Math. Comput. Model. Dyn. Syst.*, **14** (5) (2008), 391–406.
[109] A. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*, Springer International Publishing, 2017.
[110] S. N. Voormeeren, P. L. C. van der Valk, B. P. Nortier, D.-P. Molenaar, and D. J. Rixen, Accurate and efficient modeling of complex offshore wind turbine support structures using augmented superelements, *Wind Energy*, **17** (7) (2013), 1035–1054.
[111] T. Wolf, $\mathcal{H}_2$ *pseudo-optimal model order reduction*. Dissertation, Technische Universität München, 2014.
[112] S. A. Wyatt, *Issues in interpolatory model reduction: Inexact solves, second-order systems and DAEs*. PhD thesis, Virginia Tech, 2012.
[113] Y. Xu and T. Zeng, Optimal $\mathcal{H}_2$ model reduction for large scale MIMO systems via tangential interpolation, *Int. J. Numer. Anal. Model.*, **8** (1) (2011), 174–188.

Elke Deckers, Wim Desmet, Karl Meerbergen, and Frank Naets

3 Case studies of model order reduction for acoustics and vibrations

Abstract: This chapter presents several case studies to illustrate specific aspects in setting up reduced-order models of acoustic and vibration models in mechanical applications. Modal truncation approaches have been a proven workhorse for over half a century in civil and mechanical engineering, but, for many (recent) applications, these techniques are too limited. In mechanical engineering, model users are interested in a range of model applications: frequency and time domain, linear and nonlinear, single domain and multiphysics, etc. This broad range of applications makes it particularly challenging to devise appropriate reduced-order model schemes, as a scheme for one model use might be completely inadequate for other applications. Krylov methods for example have been a go-to technique in many domains, but face particular challenges in mechanical finite element models as the system's eigenvalues lie along the imaginary axis and the high frequencies are irrelevant for a given mesh size from a physical perspective. In the current chapter we explore these particularities for different types of mechanical models and simulation purposes, in order to surface several good practices and points of attention when applying model order reduction on these models. We bring together two different viewpoints: the application of model order reduction from a purely mathematical point of view and the physical interpretation of models and expected properties of reduced-order models based on physical arguments from the field of mechanics. While we touch upon a range of novel model order reduction techniques, we do not discuss parametric model order reduction as it is expected that the presented guidelines can be exploited in parametric problems without additional specific concerns.

Keywords: Model order reduction, acoustics and vibrations, finite element method, structure-preserving methods, nonlinear frequency dependency

MSC 2010: 65F50, 65F15, 65F30, 65Z05

3.1 Overview of mechanical vibration and acoustic applications and models

3.1.1 Introduction

The modeling of the vibrational and acoustic behavior of physical systems is far from trivial. In a general coupled vibro-acoustic system, in which a structure and acoustic

Elke Deckers, Wim Desmet, Frank Naets, Department of Mechanical Engineering, KU Leuven, Leuven, Belgium

Karl Meerbergen, Department of Computer Science, KU Leuven, Leuven, Belgium

https://doi.org/10.1515/9783110499001-003

cavities mutually interact with each other, the system behavior is typically determined by the coupled response of each of the components, often requiring an accurate representation of the interface. For acoustic simulations, the frequency range of interest typically runs up to 20 kHz. In real life, however, this range cannot be covered due to the limitations of the current computer-aided engineering tools. Moreover, there is a significantly different response in different frequency regions. In general, three different frequency regions can be identified, also visualized in Figure 3.1, which are typically problem-dependent:

Low-frequency range In the low frequency-range, the characteristic length of the problem under study is smaller than or in the same order of magnitude as the dominant physical wavelengths in the dynamic response. In this frequency range, the response of the system is determined by well-separated eigenvalues or modes and can be predicted by means of deterministic approaches. For vibro-acoustic problems, element-based techniques, such as the finite element method (FEM) [62] and the boundary element method [57] are most commonly applied. Element-based approaches divide the problem domain or its boundary into a large number of small elements. Inside these elements, the field variables are approximated using simple, often polynomial, functions. As wavelengths shorten with increasing frequency, the element size also needs to decrease to diminish the effect of interpolation and pollution errors [21, 34]. As a consequence, the number of degrees of freedom increases, as does the size of the system matrices, limiting the practical use of element-based approaches to lower-frequency applications.

High-frequency range When the characteristic length of the problem under study is much larger than the dominant physical wavelengths in the dynamic response, the considered problem is situated in the high-frequency range. Typically, the modal density and modal overlap are high and the system is very sensitive to small variations in for instance material properties and geometrical details. As small variabilities are inevitable in real-life applications, the response of one nominal system loses its meaning. As a result, the spatially averaged response of a number of realizations is of interest together with its variance. In this frequency range, statistical techniques are applied; for instance the statistical energy analysis (SEA) [32] is often used for vibro-acoustic analysis. The SEA divides the problem domain into a small number of subsystems in which a spatially averaged estimate of the energy level is obtained. SEA is computationally not demanding, but relies on a number of assumptions, such as for instance a high modal overlap and an energetic similarity of the different subsystems. Since these assumptions are only met above a certain frequency limit, the method is restricted to the high-frequency range.

Mid-frequency range In-between the low- and the high-frequency range, a frequency band exists for which it is stated that currently no mature and adequate prediction techniques are available. However, for many applications, this mid-frequency gap coincides with the frequency range where the human perception

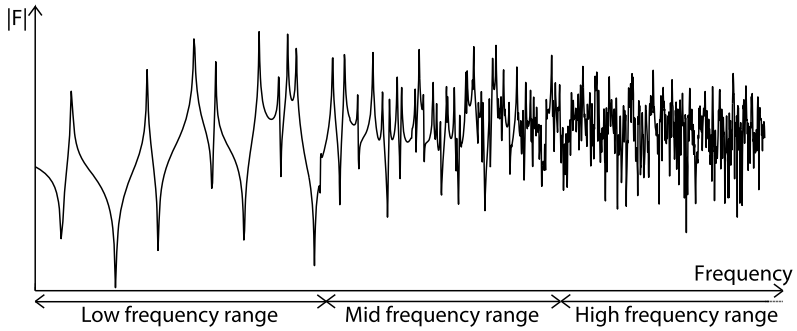


Figure 3.1: Typical frequency response function of weakly damped mechanical vibrational system [56].

and hearing is highly sensitive. Therefore, solutions are sought to bridge (part of) this gap. One approach is to increase the frequency range of the deterministic approaches by increasing the size of the resulting models. Here, amongst others, model order reduction can be an important enabler.

This chapter will focus on model order reduction techniques for vibrational and acoustic systems in the low- to mid-frequency range. Typically, these finite element models of vibrational and acoustic systems result in a system of second-order ordinary differential equations with large, but sparsely populated system matrices that allow for efficient solution algorithms. If no complex damping treatments are considered, the obtained models are linear and frequency-independent. Finite element models can easily be represented as linear state-space systems. The system matrices of a boundary element model are in general smaller (as only the boundary of the domain has to be discretized into elements resulting in a substantially lower amount of degrees of freedom) than their finite element counterpart. However, the boundary element matrices are fully populated and have a rather complex frequency dependence. For this reason, it is not straightforward to convert boundary element equations to a time-domain equivalent. This sometimes makes this approach inadequate for engineering applications.

Given the properties listed above, the maturity and the widespread industrial use of the FEM make it very accessible for practical problems. This is why only finite element models are considered in this chapter.

Besides component model analysis acceleration, reduced-order vibrational and acoustic models are a key enabler for many integrated simulation applications. In flexible multibody simulation, reduced-order vibrational component models are exploited in order to allow for the inclusion of coupled body flexibility in a mechanical system level context, as discussed in Chapter 2 of this volume. The analysis and design of new materials requires multiscale simulation where model reduction has the poten-

tial to allow to bridge these different scales in a fully coupled framework [61]. Overall, these reduced-order model (ROM) approaches can be considered as a key enabler for current and future applications where vibro-acoustic models need to be evaluated in a broader system-level performance context.

3.1.2 Mathematical models of vibrational, acoustic, and vibro-acoustic systems

3.1.2.1 Vibrational problem definition

By combining constitutive equations based on Hooke's law, which expresses that the relationship between stress and strain is linear, and momentum equations, the elastic wave equation is obtained [5]:

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = (\lambda_L + 2\mu_L)\overrightarrow{\nabla}(\overrightarrow{\nabla} \cdot \vec{u}) - \mu_L \overrightarrow{\nabla} \times \overrightarrow{\nabla} \times \vec{u} + \vec{F}_b, \tag{3.1}$$

in which $\vec{u}$ describes the solid displacement in the three spatial dimensions, $\vec{F}_b$ is the body force per unit volume acting on the solid, and λ_L and μ_L are the Lamé coefficients of the isotropic material.

For some commonly encountered geometries in vibrational problems, this elastic wave equation can be simplified. If the geometrical domain is much smaller in one direction than the other two, Kirchhoff–Love plate theory [51] can be applied, considering only bending due to transverse loads for plates subject to small deformations. It is explicitly assumed that straight cross-sections remain straight under deformation, including no shear effects. In this case, the equation of motion is given by

$$D\left(\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}\right) = -\rho \frac{\partial^2 u}{\partial t^2} + F_t, \tag{3.2}$$

where u is now only the (scalar) transverse displacement, x and y describe the in-plane location, F_t is the transverse load expressed as a force per unit area, ρ is the mass per unit area, and D is the plate bending stiffness, defined as

$$D = \frac{Eh^3}{12(1-\nu^2)}, \tag{3.3}$$

where h is the plate thickness, E is the plate material Young's modulus, and ν is its Poisson's ratio. As the Kirchhoff–Love plate equation (3.2) is a fourth-order differential equation, two boundary conditions have to be applied at each location on the boundary to have a well-posed problem. Commonly applied boundary conditions are free edges, clamped edges, and simply supported edges. Next to plates, often shells are applied. The difference with plates is that for shells also in-plane deformations and

stresses are modeled. These types of models are often used to represent thin-walled structures, increasing computational efficiency. Thin-walled structures are often encountered in vibro-acoustic applications. Shell structures are in general quite stiff in-plane, and more flexible out-of-plane. These types of structural models are typical for mechanical and civil engineering applications and are not often encountered in other disciplines. They moreover tend to couple strongly to the acoustic domain.

3.1.2.2 Acoustic problem definition

The constitutive equation in acoustics is the so-called acoustic wave equation:

$$\overrightarrow{\nabla}^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = -\rho_0 \frac{\partial q}{\partial t}, \tag{3.4}$$

which has to be solved in order to obtain the sound pressure $p(x, y, z, t)$ in a given system, being the pressure perturbation around the ambient reference state p_0. In this equation, $\overrightarrow{\nabla}^2$ is the Laplace operator which is defined as $\overrightarrow{\nabla}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$, ρ_0 is the ambient fluid density, q is the flow rate (m^3/s) of an acoustic volume source, and c is the speed of sound, defined as

$$c = \sqrt{\frac{\gamma p_0}{\rho_0}}, \tag{3.5}$$

where γ is an inherent property of the studied gas, being the ratio of the specific heat capacity for constant pressure and the specific heat capacity for constant volume.

The acoustic wave equation (3.4) assumes that the fluid behaves as an ideal gas, that pressure changes in acoustics are adiabatic, and that the fluid flow is inviscid. More details can be found in, e. g., [13].

Acoustics are often studied in the frequency domain. Assuming a time-harmonic $e^{\jmath\omega t}$-dependence of the dynamic quantities and excitations, the inhomogeneous Helmholtz equation is retrieved [43]:

$$\overrightarrow{\nabla}^2 p(\omega) + k^2 p(\omega) = -\jmath\rho_0 \omega q(\omega), \tag{3.6}$$

where

$$k = \frac{\omega}{c} = \frac{2\pi f}{c} \tag{3.7}$$

is the acoustic wavenumber at frequency f. The acoustic wavelength λ is defined in terms of the speed of sound and the frequency as

$$\lambda = \frac{c}{f}. \tag{3.8}$$

The Helmholtz equation (3.6) is a second-order differential equation, meaning that one boundary condition has to be specified at each point of the boundary in order to obtain a well-posed problem. Typically, for bounded acoustic problems, Neumann, Dirichlet, and Robin boundary conditions are applied.

3.1.2.3 Vibro-acoustic coupling

When strong vibro-acoustic coupling is considered, it is assumed that vibrations induce acoustic waves, but that the acoustic field in turn also excites structural vibrations. This is often the case for closed cavities and thin-walled structures, as often encountered in mechanical applications like car interior cavities. In a coupled vibro-acoustic system, the pressure field p in the acoustic domain and the elastic displacements $\vec{u}$ mutually affect each other on the wetted interface:

- the structural out-of-plane velocity $\dot{u}$ on the wetted structure is seen as an imposed velocity for the acoustic domain;
- the acoustic pressure field p on the wetted structure is considered as a distributed load on the structural domain.

3.1.3 Finite element modeling and discretization

3.1.3.1 General formulation

The modeling procedure of the FEM can be applied to any general set of (coupled) differential equations. In a first step, the problem domain is discretized into a large number of small elements which are interconnected by a network of n_{fe} nodes. Each field variable $v_i(\vec{r})$ at location $\vec{r}$ is approximated in each of the elements by a solution expansion $\hat{v}_i(\vec{r})$ in terms of n_{fe} (polynomial) shape functions N_{f_i}:

$$\begin{aligned} v_i(\vec{r}) \simeq \hat{v}_i(\vec{r}) &= \sum_{f_i=1}^{n_{\text{fe}}} N_{f_i}(\vec{r}) v_{f_i} \\ &= \mathbf{N}_i(\vec{r}) \mathbf{v}_i. \end{aligned} \tag{3.9}$$

The nodal values v_{f_i} belonging to each of the n_{fe} nodes are gathered in the vector of the (generalized) degrees of freedom $\mathbf{v}_i$. The row vector $\vec{N}_i$ collects the n_{fe} shape functions N_{f_i}. These shape functions only have a nonzero value inside the element to which they belong. Moreover, each shape function has a value of 1 for only one degree of freedom of the element and is zero at all others. The polynomial shape functions do not exactly satisfy the differential equations describing the physical problem to solve, nor the imposed boundary conditions. Typically, a weighted residual formulation is applied, and these errors are orthogonalized with respect to a set of weighting functions

and minimized. In the FEM, typically a Galerkin-weighted procedure is applied, expanding the weighting functions in terms of the same locally defined shape functions as for the field variables.

The FEM approach allows to generate models for the structural and acoustic problem in both the time and the frequency domain with a total of n degrees of freedom. For the time-domain simulation, the resulting (semi-discretized) model is obtained as

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{3.10}$$

with $\mathbf{K} \in \mathbb{R}^{n\times n}$ being the stiffness matrix, $\mathbf{C} \in \mathbb{R}^{n\times n}$ being the damping matrix, $\mathbf{M} \in \mathbb{R}^{n\times n}$ being the mass matrix, and $\ddot{\mathbf{x}}$ and $\dot{\mathbf{x}}$ representing respectively the first and second time derivatives of the nodal time-domain degrees of freedom $\mathbf{x} \in \mathbb{R}^n$, with external (time-domain) loads $\mathbf{f} \in \mathbb{R}^n$.

In the frequency domain, the resulting discretized model becomes

$$(\mathbf{K} + \jmath\omega\mathbf{C} - \omega^2\mathbf{M})\mathbf{x} = \mathbf{f}, \tag{3.11}$$

with $\mathbf{x}$ and $\mathbf{f}$ now represented in the frequency domain. In general vibro-acoustic problems, the model matrices $\mathbf{K}$, $\mathbf{C}$, and $\mathbf{M}$ can moreover be frequency-dependent as well. In the remainder of this chapter we will mainly focus on this frequency-dependent problem, as many of the time-domain aspects have been covered already in Chapter 2 of this volume. Nevertheless, several additional points are raised, specifically focusing on converting these frequency-domain models into equivalent time-domain models.

3.1.3.2 Properties

The discretization strategy of the FEM and the use of simple polynomial interpolation functions has its advantages and disadvantages. In practice this approach leads to the following general characteristics:

Problem discretization and degrees of freedom The finite element approach divides the problem domain into a large number of small elements. The degrees of freedom in an finite element model are the nodal values of the field variables, and inside the elements, the dynamic field is approximated using simple polynomial shape functions. As frequencies of interest increase and wavelengths shorten, the finite element mesh needs to be refined to retain a sufficient accuracy as driven by interpolation and pollution errors [21, 34]. Practically, for linear elements, a rule of thumb is to apply at least 10 element per wavelength. Calculations at higher frequencies than considered by this rule of thumb for a given mesh can be considered erroneous and are physically not meaningful.

Problem geometric complexity Due to the fine discretization typically necessary to capture the wavelengths, the FEM has almost no restrictions regarding the geometrical complexity, as the elements are required to be small anyhow.

System matrix properties In general, for undamped, viscously damped, or proportionally damped structures, the system matrices of the uncoupled acoustic and structural finite element model are real-valued, large, frequency-independent, symmetric, and sparsely populated with a banded structure. These properties allow for an efficient storage, solution, and reuse of the matrices for different frequencies [62].

Computational performance Although the finite element matrices are in general sparse and symmetric, because of the large number of finite element degrees of freedom, the solution of the finite element models is still computationally demanding. The CPU time required to build and solve the system is proportional to $n\Delta^2$, with n being the number of degrees of freedom for the FEM and Δ being the bandwidth of the system matrix.

3.1.3.3 Vibro-acoustic coupling

The uncoupled acoustic and structural subproblem result in systems of equations of the format presented in equations (3.10)–(3.11), in which the primary variables for the acoustic domain are the nodal pressure vector $\mathbf{p}$ and for the structural domain the structural displacement vector $\mathbf{u}$. By accounting for the coupling conditions between both domains, the following finite element system of equations is obtained in the frequency domain:

$$\left(\begin{bmatrix} \mathbf{K}_s & \mathbf{K}_c \\ 0 & \mathbf{K}_a \end{bmatrix} + \jmath\omega \begin{bmatrix} \mathbf{C}_s & 0 \\ 0 & \mathbf{C}_a \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{M}_s & 0 \\ -\rho_0 \mathbf{K}_c^T & \mathbf{M}_a \end{bmatrix} \right) \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_a \end{bmatrix}, \tag{3.12}$$

where the subscripts a, s, and c denote the uncoupled acoustic and structural system matrices and the coupling matrices, respectively. Equation (3.12) can be written more compactly as

$$(\mathbf{K}_{up} + \jmath\omega\mathbf{C}_{up} - \omega^2\mathbf{M}_{up})\mathbf{x}_{up} = \mathbf{f}_{up}, \tag{3.13}$$

where the subscript up refers to the use of the structural displacements and acoustic pressure as primary field variables in the structural and acoustic subproblems, respectively. This system of equations has the same shape as the uncoupled structural dynamic and acoustic finite element models. The coupled system matrices are still sparse and frequency-independent [15]. It is worth noting that the coupled stiffness matrix $\mathbf{K}_{up}$ and mass matrix $\mathbf{M}_{up}$ are no longer symmetric due to the presence of the coupling matrix $\mathbf{K}_c$.

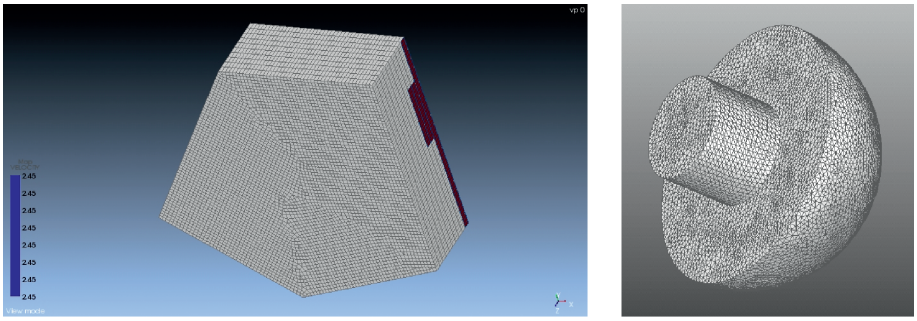
3.2 General comments on model order reduction for vibrations

3.2.1 Choice of linear system solver

Before we dive into model order reduction methods, a comment on the solution of a sparse linear system is in order. The linear system of equation (3.11) is usually hard to solve by an iterative method, even for low frequencies. The condition number of the frequency-dependent system matrix is usually high and the matrix is often far from definite. An additional difficulty is that classical preconditioners, such as multigrid and incomplete factorizations, are often only reliable when specific mathematical properties are satisfied, such as (positive) definiteness or the **M**-matrix property. The literature proposes techniques to overcome this difficulty by applying these preconditioners to another, modified system matrix that is more favorable to preconditioning. For undamped acoustics ($\mathbf{C} = 0$), the incomplete factorization was applied to $\mathbf{K} + \alpha^2\mathbf{M}$ [33] with an optimal choice of α, which is currently known as the shifted Laplace preconditioner. For damped acoustics with nonzero $\mathbf{C}$, one could apply the preconditioner to $\mathbf{K} + \jmath\omega\mathbf{C} + \omega\mathbf{C} + 2\jmath\omega^2\mathbf{M}$ (e. g., see [36]).

We now present numerical examples to illustrate properties of direct methods, i. e., methods based on a sparse lower-upper (LU) factorization, and preconditioned Krylov solvers. Consider the numerical example from [35] with the three-dimensional mesh shown in Figure 3.2a. The matrix $\mathbf{K} - \omega^2\mathbf{M}$ is real symmetric, is of order 140,228, and has 1,822,668 nonzero elements. The LU factorization of $\mathbf{K} - \omega^2\mathbf{M}$ using MUMPS [40] on a Dell Latitude 6400 required 13 seconds in 2009. The construction of a Krylov basis of dimension 50 by the Lanczos method (excluding the LU factorization) required 15 seconds. This shows that the factorization cost is highly dominant. Consider another example from [36], which is a finite element problem with spherical infinite elements. The mesh is given in Figure 3.2b. The system matrix in (3.11) is nonsymmetric and complex-valued. The order is 72,976 and the matrix contains 1,541,904 nonzero elements. Timings on a Dell Latitude 6400 in 2009 showed that the direct solve with MUMPS required 119.13 seconds (LU factorization and forward and backward solve). BiCGStab with the ILU preconditioner from [36] required 52.20 seconds at a frequency of 200 Hz and 103.71 seconds at 500 Hz. It is well known that iterative methods perform typically badly for higher frequencies, which is confirmed by this experiment. We also see that direct methods are competitive with iterative ones for this problem.

In our experience, iterative methods are not competitive with direct methods for most applications in finite element (vibro-)acoustics. Therefore, the use of a direct linear solver is very common in model order reduction. For a direct linear solver, the highest cost is the sparse LU factorization. Once the LU factorization is performed, the solve cost is only a fraction, typically 10 % or less, of the factorization cost. For this reason, the methods based on Krylov and rational Krylov sequences are very popular and



a. [35] b. [36]

Figure 3.2: Meshes for linear solver benchmarks.

by far the most efficient methods in terms of computational cost, since they reduce the number of matrix factorizations. In practice, it implies that methods such as the dominant pole algorithm, the iterative rational Krylov algorithm (IRKA) [2], and variations on these methods are only used for models of relatively small scale. Unfortunately, Krylov methods usually lead to larger orders of the ROM for the same accuracy than balanced truncation or IRKA.

3.2.2 Frequency limitation

For model order reduction, one should be aware of the fact the mesh is only valid for a limited frequency range. This is another reason why (rational) Krylov methods are popular, as they focus on a limited interval on the $\jmath\omega$-axis. Sometimes, the high-frequency eigenvalues of the discrete model are the most dominant ones, so that default implementations of methods such as the dominant pole algorithm, balanced truncation, and IRKA cannot be used. However, there are modifications of these methods that limit the frequency range as well. For the dominant pole algorithm, it is sufficient to modify the definition of dominance, taking into account the frequency limitation [49]. For balanced truncation the frequency limitation can be taken into account: In [7], the right-hand side of the Lyapunov equation is modified in such a way that the high-frequency content is filtered out. However, the obtained ROM may no longer be stable.

3.2.3 Modal approximation

For the order reduction of mechanical models, modal approximation approaches have been very popular over the past decades both in research and engineering practice. These approaches are based on the practical observation that the response of

a mechanical second-order system is typically dominated by its lowest-frequency modes. This is the result of the dynamics stiffness increasing considerably for higher-frequency modes, resulting in small-amplitude contributions to the overall response.

In its most basic version this approach uses a set of free-free eigenmodes $\mathbf{V}$ of the undamped model for the reduced-order basis [24]:

$$(-\mathbf{M}_s\omega_i^2 + \mathbf{K}_s)V_i = 0, \quad \forall\omega_i \leq \omega_{\max}, \tag{3.14}$$

where V_i is the modal shape vector at pulsation ω_i. An important benefit of this approach is the inherent inclusion of the limited frequency range in which the model discretization is valid. However, this basic approach tends to lead to poor accuracy as it does not account for the particular interface conditions of the model. In order to robustify with respect to these conditions, a range of extensions have been proposed which augment the initial modal basis with specific interface modes. The best-known reduced-order basis in this framework is the component mode synthesis approach [16]. However, over the years many authors have proposed a range of approaches which fit this framework [12].

A major reason why these approaches tend to be popular in practice is the clear physical interpretation of these ROMs. This if often important because practicing engineers tend to prefer approaches which they understand, as the ROM setup typically requires tuning for different applications. However, these methods also suffer from two important drawbacks:

- No reliable error estimators exist for modal approximation approaches, leading to tedious tuning by the user to achieve the desired accuracy.
- The computation of the free-free modes can be expensive for-large scale models. This can limit the overall gains in the model evaluation time when also accounting for the MOR setup time.

As a result of these drawbacks, they are not discussed in more detail in the remainder of this chapter. Nevertheless, these modal reduction approaches are still very popular in practice. They moreover serve as inspiration for a range of novel nonlinear model order reduction schemes like the modal derivative approach for nonlinear problems, as discussed in Chapter 2 of this volume.

3.2.4 Rational Krylov methods

For the sake of notation, let us repeat the idea of Krylov methods from [9, Chapter 3] and their mathematical and algorithmic properties important for this chapter. Consider the following linear (descriptor) state-space model in the Laplace variable:

$$\begin{aligned} \mathbb{A}\mathbf{x} - s\mathbb{E}\mathbf{x} &= \mathbf{b}, \\ H &= \mathbf{c}^T\mathbf{x}, \end{aligned} \tag{3.15}$$

where $\mathbb{A}, \mathbb{E} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{\tilde{n}}$ with $\tilde{n}$ large. (Rational) Krylov methods build a basis of dimension k, which we denote by matrix $\mathbf{V} \in \mathbb{C}^{\tilde{n}\times k}$, for the column space of

$$[\mathbf{K}_{m_1}((\mathbb{A}-\sigma_1\mathbb{E})^{-1}\mathbb{E}, (\mathbb{A}-\sigma_1\mathbb{E})^{-1}\mathbf{b}), \ldots, \mathbf{K}_{m_p}((\mathbb{A}-\sigma_p\mathbb{E})^{-1}\mathbb{E}, (\mathbb{A}-\sigma_p\mathbb{E})^{-1}\mathbf{b})],$$

with $\mathbf{K}_m(\mathbb{T}, \mathbf{b})$ the order m Krylov matrix

$$\mathbf{K}_m(\mathbb{T}, \mathbf{b}) = [\mathbf{b}, \mathbb{T}\mathbf{b}, \ldots, \mathbb{T}^{m-1}\mathbf{b}].$$

This sequence uses p different shifts, the i-th shift with multiplicity m_i. Now let $\mathbf{W} \in \mathbb{C}^{n\times k}$ be an arbitrary rank k matrix. The order k ROM

$$\begin{aligned} \widehat{\mathbb{A}}\hat{\mathbf{x}} - s\widehat{\mathbb{E}}\hat{\mathbf{x}} &= \widehat{\mathbf{b}}, \\ \widehat{H} &= \hat{\mathbf{c}}^T\hat{\mathbf{x}}, \end{aligned} \tag{3.16}$$

with $\widehat{\mathbb{A}} = \mathbf{W}^T\mathbb{A}\mathbf{V}$, $\widehat{\mathbb{E}} = \mathbf{W}^T\mathbb{E}\mathbf{V} \in \mathbb{C}^{k\times k}$, and $\widehat{\mathbf{b}} = \mathbf{W}^T\mathbf{b}$, $\hat{\mathbf{c}} = \mathbf{V}^T\mathbf{c} \in \mathbb{C}^k$, has the following moment matching properties:

$$\frac{d^j}{ds^j}\widehat{H}(\sigma_i) = \frac{d^j}{ds^j}H(\sigma_i), \quad j = 0, \ldots, m_i - 1, \quad i = 1, \ldots, p.$$

If, in addition, the columns of $\mathbf{W}$ span the adjoint space

$$[\mathbf{K}_{m_1}((\mathbb{A}-\sigma_1\mathbb{E})^{-T}\mathbb{E}^T, (\mathbb{A}-\sigma_1\mathbb{E})^{-T}\mathbf{c}), \ldots, \mathbf{K}_{m_p}((\mathbb{A}-\sigma_p\mathbb{E})^{-T}\mathbb{E}^T, (\mathbb{A}-\sigma_p\mathbb{E})^{-T}\mathbf{c})],$$

the ROM (3.16) has the following moment matching properties:

$$\frac{d^j}{ds^j}\widehat{H}(\sigma_i) = \frac{d^j}{ds^j}H(\sigma_i), \quad j = 0, \ldots, 2m_i - 1, \quad i = 1, \ldots, p.$$

The advantage of Krylov methods is that the poles can be chosen so that
1. the limitation to a frequency range is respected; and
2. the number of large-scale LU factorizations is small (in this case p).

The downside of the IRKA [2] and the dominant pole algorithm [46] is the large computational cost due to a large number of sparse LU factorizations. In fact, if the number of factorizations is as large as the number of points required by a POD approach for the frequency axis, there is no interest in using such methods. Greedy methods (see [10]) do not guarantee the same approximation error as $\mathcal{H}_\infty$- and $\mathcal{H}_2$-minimization of the error, but they require fewer matrix factorizations.

The main disadvantage of a Krylov method is that, in general, stability is not guaranteed. We will see in Section 3.5 an application for which stability is guaranteed. Another disadvantage is that the size of the reduced model may not be minimal for the same accuracy as balanced truncation. Krylov methods are popular because they produce a reasonably good reduction in a relatively low computational time for the ROM setup. As an extension, the obtained ROMs can be further reduced by balanced truncation, e. g., [26], in order to mitigate the high setup cost of the balanced truncation model on a high-order model. We give other examples in Section 3.4.4.

3.2.5 Concept of linearization

The standard approach for mapping the second-order matrix polynomial in (3.11) to a linear form is called linearization. Linearization allows to use model order reduction methods for linear models as in (3.15). Best known is the *companion linearization*. It doubles the dimension of the state space, but the system is linear and produces the same transfer function as the system with quadratic frequency dependency. System (3.11), with output $H = \mathbf{c}^T\mathbf{x}$, can be rewritten as the linear model

$$\left(\begin{bmatrix}\mathbf{K} & 0\\ 0 & \mathbf{I}\end{bmatrix} + \jmath\omega\begin{bmatrix}\mathbf{C} & \mathbf{M}\\ -\mathbf{I} & 0\end{bmatrix}\right)\begin{bmatrix}\mathbf{x}\\ \jmath\omega\mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{f}\\ 0\end{bmatrix}, \tag{3.17}$$

$$H = \begin{bmatrix}\mathbf{c}\\ 0\end{bmatrix}^T\begin{bmatrix}\mathbf{x}\\ \jmath\omega\mathbf{x}\end{bmatrix}.$$

Any method for linear systems can be used for reducing (3.17). It should be noted that the state vector now has two components: $\mathbf{x}$ and $\jmath\omega\mathbf{x}$. The Krylov vectors now have length $\tilde{n} = 2n$. However, due to the structure of the state space the memory cost of the iteration vectors can be halved [3].

Linearizations are also used for other polynomial, rational, or even fully nonlinear dependencies on the frequency. Efficient implementations of Krylov methods (one- and two-sided) rely on a similar property as the state vector of second-order problems [53, 30]. We will give more examples in Section 3.4.

3.3 Structure-preserving model order reduction

In many cases, it may be important to have a ROM that respects the structure of the large-scale model. Such structures can take various forms: real matrices, symmetric matrices, polynomial frequency dependence, and so on. For example, the typical structure for frequency-domain vibration analysis is the form of (3.11). Finding a ROM of exactly the same structure (symmetric matrices, quadratic frequency dependence) may be beneficial for keeping spectral properties, e. g., but also to physically interpret the ROM.

3.3.1 Quadratic frequency-domain structure

In this section, we discuss the exploitation of quadratic structure as in (3.11). The choice of ROM depends on its purpose. If the model is to be coupled with other models in the time domain, a linear model is usually preferred, since the connection with the time domain is straightforward. For reliable time-domain simulation a stable model (which mechanical systems inherently are) is required, but this is not always the case

for the ROM. By respecting the quadratic structure, the stability can sometimes be guaranteed.

For second-order systems (3.11), Krylov methods rely on linearization (3.17). Assume that the reduced model is obtained by projection of (3.17) on subspace $\mathcal{V}$ spanned by the columns of the full-rank matrix $\mathbf{V} \in \mathbb{C}^{2n\times k}$. Decompose

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}, \quad \mathbf{V}_1, \mathbf{V}_2 \in \mathbb{C}^{n\times k}.$$

Let $\mathbf{V}$ be computed by an interpolatory model order reduction method, i. e., a (rational) Krylov method, on (3.17). Interpolation at $\omega = \omega_i$ implies that $\mathbf{x}(\omega_i)$ is spanned by the columns of $\mathbf{V}_1$ and $\jmath\omega_i\mathbf{x}(\omega_i)$ by the columns of $\mathbf{V}_2$. This suggests that there is a strong connection between $\mathbf{V}_1$ and $\mathbf{V}_2$. Indeed, second-order Arnoldi (SOAR) [3] and two-level orthogonal Arnoldi [31], for example, exploit this property and express $\mathbf{V}_i = \mathbf{Q}\mathbf{U}_i$, $i = 1, 2$, with $\mathbf{Q} \in \mathbb{C}^{n\times\ell}$, $\mathbf{U}_i \in \mathbb{C}^{\ell\times k}$, where $\ell \leq k$. Instead of building a linear reduced model, one might, as in SOAR, compute a second-order reduced model by projection on the column span of $\mathbf{Q}$:

$$\begin{aligned} (\widehat{\mathbf{K}} + \jmath\omega\widehat{\mathbf{C}} - \omega^2\widehat{\mathbf{M}})\widehat{\mathbf{x}} &= \widehat{\mathbf{f}}, \\ \widehat{H} &= \widehat{\mathbf{c}}^T\widehat{\mathbf{x}}, \end{aligned} \tag{3.18}$$

with $\widehat{\mathbf{K}} = \mathbf{Q}^T\mathbf{K}\mathbf{Q}$, $\widehat{\mathbf{C}} = \mathbf{Q}^T\mathbf{C}\mathbf{Q}$, $\widehat{\mathbf{M}} = \mathbf{Q}^T\mathbf{M}\mathbf{Q}$, $\widehat{\mathbf{f}} = \mathbf{Q}^T\mathbf{f}$, and $\widehat{\mathbf{c}} = \mathbf{Q}^T\mathbf{c}$.

The situation is more complicated for two-sided model order reduction. Indeed, the state vector of the adjoint of the linear problem

$$\begin{aligned} \begin{bmatrix} \mathbf{K}^T + \jmath\omega\mathbf{C}^T & -\jmath\omega\mathbf{I} \\ \jmath\omega\mathbf{M} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ -\jmath\omega\mathbf{M}^T\mathbf{z} \end{bmatrix} &= \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}, \\ H &= \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}^T \begin{pmatrix} \mathbf{z} \\ -\jmath\omega\mathbf{M}^T\mathbf{z} \end{pmatrix} \end{aligned} \tag{3.19}$$

has the two components $\mathbf{z} \in \mathbb{C}^n$ and $-\jmath\omega\mathbf{M}^T\mathbf{z} \in \mathbb{C}^n$. Now assume that the reduced model is obtained by projection on subspace $\mathcal{W}$ spanned by the columns of the full-rank matrix $\mathbf{W} \in \mathbb{C}^{2n\times k}$. Decompose

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}, \quad \mathbf{W}_1, \mathbf{W}_2 \in \mathbb{C}^{n\times k}.$$

Let $\mathbf{W}$ be computed by an interpolatory method on (3.19), i. e., there are ω so that $\mathbf{z}(\omega)$ is spanned by the columns of $\mathbf{W}_1$ and $\jmath\omega\mathbf{M}^T\mathbf{z}(\omega)$ by the columns of $\mathbf{W}_2$. This suggests that there is also a strong connection between $\mathbf{W}_1$ and $\mathbf{W}_2$. Indeed, in [30] this property is exploited by expressing $\mathbf{W}_1 = \mathbf{Z}\mathbf{T}_1$ and $\mathbf{W}_2 = \mathbf{M}^T\mathbf{Z}\mathbf{T}_2$ with $\mathbf{Z} \in \mathbb{C}^{n\times\ell}$, $\mathbf{T}_i \in \mathbb{C}^{\ell\times k}$, where $\ell \leq k$.

A linear ROM is obtained by projecting (3.17) on the right by $\mathbf{V}$ and the left by $\mathbf{W}^T$. A structure-preserving alternative is to project (3.17) by

$$\begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{Q} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{Z}^T & 0 \\ 0 & \mathbf{Z}^T\mathbf{M} \end{bmatrix},$$

on the right and the left, respectively. Note that the first spans the columns of $\mathbf{V}$ and the second the rows of $\mathbf{W}^T$, so the interpolation properties are preserved with these bases. This leads to

$$\begin{bmatrix} \widehat{\mathbf{K}} + \jmath\omega\widehat{\mathbf{C}} & \jmath\omega\widehat{\mathbf{M}} \\ -\jmath\omega\widehat{\mathbf{M}} & \widehat{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}} \\ \jmath\omega\hat{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}} \\ 0 \end{bmatrix},$$

$$\widehat{H} = \begin{bmatrix} \hat{\mathbf{c}} \\ 0 \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{x}} \\ \jmath\omega\hat{\mathbf{x}} \end{bmatrix},$$

with $\widehat{\mathbf{K}} = \mathbf{Z}^T\mathbf{K}\mathbf{Q}$, $\widehat{\mathbf{C}} = \mathbf{Z}^T\mathbf{C}\mathbf{Q}$, $\widehat{\mathbf{M}} = \mathbf{Z}^T\mathbf{M}\mathbf{Q}$, $\hat{\mathbf{f}} = \mathbf{Z}^T\mathbf{f}$, and $\hat{\mathbf{c}} = \mathbf{Q}^T\mathbf{c}$. Assuming that $\widehat{\mathbf{M}}$ has full rank, this is equivalent to the frequency-nonlinear model (3.18).

We now illustrate one-sided linear and second-order reduced models for the following example, whose description is taken from [60]. The model describes a footbridge located over the Dijle river in Mechelen (Belgium). It is about 31.354 m in length and four tuned mass dampers (TMDs) are located at nodes at 11.299 m, 19.314 m, 10.549 m, and 20.309 m, respectively, each of which weighs 40.72 kg. The discretized model is

$$\left(\mathbf{K}_0 + \jmath\omega\mathbf{C}_0 + \sum_{i=1}^{4} \mathbf{K}_i - \omega^2\mathbf{M}_0\right)\mathbf{x} = \mathbf{f},$$

where $\mathbf{K}_0$ and $\mathbf{M}_0$ are obtained from a finite element model with 25,962 degrees of freedom, as shown in Figure 3.3a. Here, $\mathbf{C}_0 = 0.1003\mathbf{M}_0 + 0.0001591\mathbf{K}_0$ represents Rayleigh damping, $\mathbf{K}_i$ is a matrix with four nonzero entries that represents the interaction between the i-th TMD and the footbridge, and the input vector f represents an excitation equally spread among the locations of the TMDs. All matrices are symmetric positive semi-definite. We used 20 iterations of Arnoldi's method (Krylov method) with single shift $50.5\jmath$ on (3.17). A linear ROM was obtained by one-sided projection $\widehat{\mathbb{A}} = \mathbf{V}^*\mathbb{A}\mathbf{V}$ and $\widehat{\mathbb{B}} = \mathbf{V}^*\mathbb{B}\mathbf{V}$ and a second-order model of the form (3.18). Figure 3.3b shows the residual norm of (3.11) for $\omega \in \jmath[0, 100]$ when (3.17) is projected on the Krylov space (linear model) and when (3.11) is projected on the column range of $\mathbf{Q}$ (second-order model). It is easily seen that around the interpolation point, the error is of the same order of magnitude but that further away, the second-order model has lower residual norms. This can be explained by the fact that projection of (3.11) is equivalent to projecting (3.17) on a larger subspace, similarly to two-sided models. An additional advantage for the second-order model is that the matrices in (3.18) are Hermitian semi-positive definite, so that stability of the reduced model is guaranteed. This is not the case for the linear ROM.

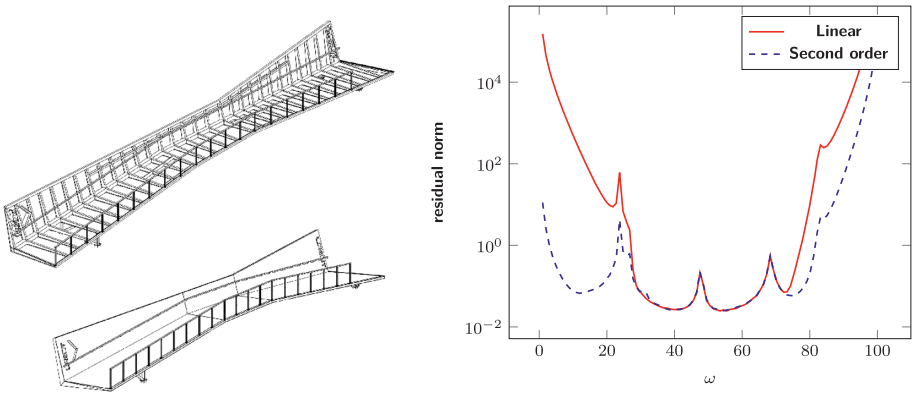


Figure 3.3: (a) Mesh. (b) Residual norm on (3.11) for the linearized model and the original second-order model.

3.3.2 Lagrangian structure in mechanical models

For classical mechanical systems, the equations of motion in the form as presented in equation (3.10) can be obtained from a Lagrangian description of the system, with the Lagrangian $\mathcal{L}$:

$$\begin{aligned}\mathcal{L} &= \mathcal{K} - \mathcal{V} + \mathcal{W} \\ &= \frac{1}{2}\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}} - \frac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{x} - \mathbf{x}^T\mathbf{f},\end{aligned} \tag{3.20}$$

where $\mathcal{K}$, $\mathcal{V}$, and $\mathcal{W}$ are respectively the kinetic energy, the internal elastic energy, and external work. The equations of motion are then obtained by applying Hamilton's principle to the Lagrangian:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{f}. \tag{3.21}$$

This Lagrangian structure inherently embeds energy-preserving behavior in the system, ensuring stability for long-term simulations. It is therefore important that the ROMs for mechanical systems respect this structure. In order to ensure that the resulting ROM equations of motion comply with this Lagrangian structure, a direct substitution in the Lagrangian can be performed. For a constant reduced-order basis, one can substitute for the reduced degrees of freedom q:

$$\mathbf{x} \simeq \mathbf{V}\mathbf{q}, \quad \dot{\mathbf{x}} \simeq \mathbf{V}\dot{\mathbf{q}}, \tag{3.22}$$

which results in the Lagrangian expressed as a function of the reduced-order degrees of freedom $\mathbf{q}$:

$$\mathcal{L} = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{V}^T\mathbf{M}\mathbf{V}\dot{\mathbf{q}} - \frac{1}{2}\mathbf{q}^T\mathbf{V}^T\mathbf{K}\mathbf{V}\mathbf{q} - \mathbf{q}^T\mathbf{V}^T\mathbf{f}, \tag{3.23}$$

such that after application of Hamilton's principle for the reduced degrees of freedom, the reduced equations of motion become

$$\mathbf{V}^T\mathbf{M}\mathbf{V}\ddot{\mathbf{q}} + \mathbf{V}^T\mathbf{K}\mathbf{V}\mathbf{q} = \mathbf{V}^T\mathbf{f}. \tag{3.24}$$

As this form results from the (reduced) Lagrangian of the system, it inherently preserves the stability of the underlying mechanical model.

This implies some care on how the model order reduction on a mechanical system is applied, as the models are often presented in different form. For example, for practical time-domain simulations, equation (3.10) is often converted to a first-order state-space model as

$$\begin{bmatrix}\dot{\mathbf{x}}\\ \ddot{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}0 & \mathbf{I}\\ -\mathbf{M}^{-1}\mathbf{K} & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} + \begin{bmatrix}0\\ \mathbf{f}\end{bmatrix}. \tag{3.25}$$

In a regular projection framework, state-space system (3.25) would be reduced as

$$\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} = \mathbf{V}\mathbf{q} = \begin{bmatrix}\mathbf{V}_1\\ \mathbf{V}_2\end{bmatrix}\mathbf{q}, \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{3.26}$$

and one would obtain a reduced system of equations as

$$\dot{\mathbf{q}} = \begin{bmatrix}0 & \mathbf{V}_1^T\mathbf{I}\mathbf{V}_2\\ -\mathbf{V}_2^T\mathbf{M}^{-1}\mathbf{K}\mathbf{V}_1 & 0\end{bmatrix}\mathbf{q} + \begin{bmatrix}0\\ \mathbf{V}_2^T\mathbf{f}\end{bmatrix}. \tag{3.27}$$

It is now clear that this system of equations cannot be converted back into a Lagrangian form like (3.23), such that time-stable and energy-preserving behavior is not naturally preserved. From this perspective, it is advisable to always apply symmetric projection of the second-order system for mechanical models if the ROM will be employed in time-domain simulations. This again shows the conceptual benefit of approaching mechanical ROMs through a one-sided projection on the second-order system.

3.4 Complex frequency dependencies

3.4.1 Sources of nonlinearities

To mitigate noise and vibration issues in mechanical engineering applications, damping treatments are often applied. The physical behavior of these treatments are mostly represented by (complex) frequency-dependent behavior. Poro-elastic materials, for example, are often used in a sound absorption context. An overview of modeling approaches for poro-elastic materials can be found in amongst others [1, 17]. Viscoelastic

materials are often applied in a constrained component setting as this ensures shear, and thus more dissipation in the sample. Their behavior strongly depends on temperature and frequency. Detailed descriptions of various mathematical models are given in [20, 45].

When unbounded acoustic domains are considered (exterior problems), using finite element models, the domain has to be truncated in practice. In this case the Sommerfeld radiation condition [14], which ensures that no acoustic energy reflects back from infinity, has to be approximated. Commonly applied approaches, like absorbing boundary conditions [4] and perfectly matched layers [11], also result in complex frequency dependencies.

3.4.2 Rational and polynomial frequency dependencies

We assume that the frequency-dependent full-order system, of order n, can be written as

$$\begin{aligned} \mathbb{A}(s)\mathbf{x} &= \mathbf{f}, \\ H &= \mathbf{c}^T\mathbf{x}, \\ \text{with} \quad \mathbb{A}(s) &= \sum_{i=0}^{d-1} \phi_i(\mathbb{A}_i + s\mathbb{B}_i) \quad \text{and} \quad s = \jmath\omega. \end{aligned} \tag{3.28}$$

We introduce the following notation:

$$\begin{aligned} \mathbf{A} + s\mathbf{B} &= [\mathbb{A}_0 + s\mathbb{B}_0, \ldots, \mathbb{A}_{d-1} + s\mathbb{B}_{d-1}], \\ \mathbf{\Phi} &= [\phi_0, \ldots, \phi_{d-1}]^T, \end{aligned}$$

where $\mathbf{\Phi}$ forms a set of polynomials or rational polynomials that satisfy

$$(\mathbf{P} + s\mathbf{R})\mathbf{\Phi} = 0,$$

with $\mathbf{P}, \mathbf{R} \in \mathbb{C}^{(d-1)\times d}$ constant matrices. This leads to

$$\mathbb{A}(s) = (\mathbf{A} + s\mathbf{B})(\mathbf{\Phi} \otimes \mathbf{I}_n).$$

As a result (3.28) can be rewritten as an order nd linear system:

$$\begin{bmatrix} \mathbf{A} + s\mathbf{B} \\ (\mathbf{P} + s\mathbf{R}) \otimes \mathbf{I}_n \end{bmatrix} \begin{pmatrix} \phi_0\mathbf{x} \\ \phi_1\mathbf{x} \\ \vdots \\ \phi_{d-1}\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$H = [\mathbf{c}^T\ 0\ \cdots\ 0] \begin{pmatrix} \phi_0\mathbf{x} \\ \vdots \\ \phi_{d-1}\mathbf{x} \end{pmatrix}$$

or in compact form

$$\begin{bmatrix} \mathbf{A} + s\mathbf{B} \\ (\mathbf{P} + s\mathbf{R}) \otimes \mathbf{I}_n \end{bmatrix} (\mathbf{\Phi} \otimes x) = e_1 \otimes \mathbf{f}, \tag{3.29}$$

$$H = (e_1 \otimes \mathbf{c})^T (\mathbf{\Phi} \otimes \mathbf{x}),$$

which is clearly linear in s. For the companion linearization from Section 3.2.5 of second-order problems, $\mathbf{P} = [-1, 0]$, $\mathbf{R} = [0, 1]$, $\mathbf{\Phi}(s) = [1, s]^T$. For the rational matrix polynomial

$$\mathbb{A}(s) = (\mathbb{A}_0 + s\mathbb{B}_0) + \frac{1}{s - \sigma_0}(\mathbb{A}_1 + s\mathbb{B}_1) + \frac{1}{s - \sigma_1}(\mathbb{A}_2 + s\mathbb{B}_2),$$

a possible linearization is

$$\begin{bmatrix} \mathbb{A}_0 + s\mathbb{B}_0 & \mathbb{A}_1 + s\mathbb{B}_1 & \mathbb{A}_2 + s\mathbb{B}_2 \\ \mathbf{I} & \sigma_0\mathbf{I} - s\mathbf{I} & 0 \\ \mathbf{I} & 0 & \sigma_1\mathbf{I} - s\mathbf{I} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \frac{1}{s-\sigma_0}\mathbf{x} \\ \frac{1}{s-\sigma_1}\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ 0 \end{pmatrix}.$$

In this case, the choice of basis and the choice of linearization are important. Some formulations lead to a descriptor system, which may require additional care for proper setup of the ROM. However, for models in the frequency domain, we have never experienced difficulties in this case.

It can be proven that the (rational) Arnoldi method for (3.29) produces iteration vectors that take the following form: If the k vectors are collected in an $nd \times k$ matrix $\mathbf{V}_k$, this matrix can be factored as

$$\mathbf{V}_k = (\mathbf{I}_d \otimes \mathbf{Q})\mathbf{U}_k, \quad \mathbf{Q} \in \mathbb{C}^{n \times \ell}, \ \mathbf{U} \in \mathbb{C}^{\ell d \times k}, \tag{3.30}$$

with $\ell \leq k + d$ [31, 53]. The advantage of this factorization is that large-scale operations only happen with columns of $\mathbf{Q}$ and that all other operations are small-scale. This leads to a reduction of the full unstructured storage of Krylov vectors of order ndk to at most $n\ell + \ell dk$ with $\ell \leq k + d$ [53]. For two-sided Krylov methods, the exploitation of the structure of (3.29) is possible, but is more involved [42, 22, 30]. In each iteration of a Krylov method, a linear system is solved. In [53], it is shown that this requires matrix vector multiplications with $\mathbb{A}_i$ and $\mathbb{B}_i$ and a linear solve with $\mathbb{A}(\sigma_k)$ at step k.

As for second-order problems, $\mathbf{Q}$ can be used to develop a nonlinear frequency-dependent reduced model:

$$\widehat{\mathbb{A}} = \sum_{j=0}^{d-1} (\widehat{\mathbb{A}}_j + s\widehat{\mathbb{B}}_j)\phi_j, \quad \widehat{\mathbb{A}}_j + s\widehat{\mathbb{B}}_j = \mathbf{Q}^*(\mathbb{A}_j + s\mathbb{B}_j)\mathbf{Q}, \quad j = 0, \ldots, d-1.$$

If all shifts $\sigma_1, \ldots, \sigma_k$ are distinct, then

$$\text{Range}(\mathbf{Q}) = \text{Range}([\mathbf{x}(\sigma_1), \ldots, \mathbf{x}(\sigma_k)]) = \text{Range}([\mathbb{A}(\sigma_1)^{-1}\mathbf{f}, \ldots, \mathbb{A}(\sigma_k)^{-1}\mathbf{f}]). \tag{3.31}$$

When the shifts are all equal,

$$\text{Range}(\mathbf{Q}) = \text{Range}([\mathbf{x}(\sigma_1), d\mathbf{x}(\sigma_1)/ds, \ldots, d^{k-1}\mathbf{x}(\sigma_1)/ds^{k-1}]). \tag{3.32}$$

In other words, the range of $\mathbf{Q}$ spans the moments of the state vector $\mathbf{x}$ of (3.28). If only $\mathbf{Q}$ is required, a linearization is not useful, since $\mathbf{Q}$ can be computed directly using (3.31) or (3.32), which requires linear solves with $\mathbb{A}(s)$. However, in the case of higher-order interpolation, as in (3.32), the computation of the derivatives is equally complicated as performing a Krylov step with the linearization.

If a linear ROM is required, then a linearization is useful. In this case, (3.29) is projected using the full Krylov vectors given by the factorization (3.30). For two-sided model order reduction, with simple interpolation points, a linear reduced model can equally be obtained by putting samples of H from (3.28) in a Loewner matrix; see [9, Chapter 6]. In this case, the reduced model is defined with

$$\widehat{\mathbb{E}} = \begin{bmatrix} \frac{H(\sigma_1)-H(\tau_1)}{\sigma_1-\tau_1} & \cdots & \frac{H(\sigma_k)-H(\tau_1)}{\sigma_k-\tau_1} \\ \vdots & \ddots & \vdots \\ \frac{H(\sigma_1)-H(\tau_k)}{\sigma_1-\tau_k} & \cdots & \frac{H(\sigma_k)-H(\tau_k)}{\sigma_k-\tau_k} \end{bmatrix},$$

$$\widehat{\mathbb{A}} = \begin{bmatrix} \frac{\sigma_1 H(\sigma_1)-\tau_1 H(\tau_1)}{\sigma_1-\tau_1} & \cdots & \frac{\sigma_k H(\sigma_k)-\tau_1 H(\tau_1)}{\sigma_k-\tau_1} \\ \vdots & \ddots & \vdots \\ \frac{\sigma_1 H(\sigma_1)-\tau_k H(\tau_k)}{\sigma_1-\tau_k} & \cdots & \frac{\sigma_k H(\sigma_k)-\tau_k H(\tau_k)}{\sigma_k-\tau_k} \end{bmatrix},$$

and

$$\widehat{\mathbf{b}} = \begin{bmatrix} H(\sigma_1) \\ H(\sigma_2) \\ \vdots \\ H(\sigma_k) \end{bmatrix}, \quad \widehat{\mathbf{c}} = \begin{bmatrix} \bar{H}(\tau_1) \\ \bar{H}(\tau_2) \\ \vdots \\ \bar{H}(\tau_k) \end{bmatrix}.$$

When the interpolation points are not too close to each other, this is a reliable and easy-to-implement method.

We now present two cases of nonlinear frequency dependency. The first case, presented in Section 3.4.3, uses (3.31) to directly construct $\mathbf{Q}$ and computes a linear ROM using Loewner matrices. The second case uses (3.32) to reduce the number of matrix factorizations; see Section 3.4.4.

3.4.3 Matrix-free model order reduction for vibro-acoustic systems with complex noise control treatments

Many model order reduction methods cannot straightforwardly cope with frequency dependencies in vibro-acoustic models, resulting from, amongst others, complex

damping treatments or infinite acoustic domains. In this section, a matrix-free Krylov MOR method [27] is presented that does not require knowledge on the underlying mathematical model and that can straightforwardly handle problems with frequency-dependent parameters in exactly the same manner as problems with constant properties. The method only works in the frequency domain and is based on a two-sided model order reduction approach with distinct interpolation points using the Loewner matrices as discussed in Section 3.4.2. The frequency-dependent transfer function between input and output is indicated by $H(\omega)$. The use of the Loewner matrices leads to a new approximative transfer function $\hat{H}$ which interpolates the original transfer function H, using the forced responses at the $2k$ distinct interpolation points.

3.4.3.1 Frequency selection and convergence evaluation

In the following, let $\hat{H}_k$ denote the transfer function (or frequency response function) obtained using a Loewner model of order k, i. e., using k right interpolation points and k left interpolation points. Jonckheere et al. [47, 48] present two criteria to assess the convergence of the ROM, which leads to a greedy method for determining the interpolation points. The first convergence criterion is based on the relative error between two subsequent ROM approximations, which is computationally cheap to evaluate:

$$\varepsilon_{\text{ROM,ROM}} = \max_{\omega} \left| \frac{\hat{H}_k(\omega) - \hat{H}_{k-1}(\omega)}{\hat{H}_k(\omega)} \right|. \tag{3.33}$$

However, the relative error between two subsequent approximations may be low in case the newly added frequencies are close to the frequencies added in the previous iteration and bring limited additional information to the interpolation. Therefore, a second convergence criterion is embedded by comparing the (approximated) transfer function obtained using the current ROM with the (exact) transfer function, calculated by solving the full system, at the newly selected frequency lines ω_k which would be needed to build a new ROM in the next iteration:

$$\varepsilon_{\text{ROM,Full}} = \max_{\omega_k} \left| \frac{\hat{H}_k(\omega) - H_k(\omega)}{\hat{H}_k(\omega)} \right|. \tag{3.34}$$

The relative error between two subsequent ROMs (3.33) is used to select new frequency lines to enrich the ROM in the next iteration. This selection is done in a cascading way: (i) the frequency at which the maximum error occurs is selected, (ii) the direct vicinity of the new frequency line is masked to avoid selecting neighboring maxima, thus to spread out the new frequency lines a bit, and (iii) from the remaining frequencies the one corresponding to the new maximum error is selected, and the procedure continues along (ii)–(iii) until the requested number of additional frequency lines is reached.

3.4.3.2 Calculation example

One example considering a mechanical system with complex damping treatments taken from [47] is presented. The example geometry is shown in Figure 3.4, which is taken from the Comsol manual [39]. This interior car cavity has dimensions of approximately 4.5 m by 1.5 m by 2 m and has 26 acoustic modes below 300 Hz. All acoustic boundaries are rigid, except for the fire wall (indicated in light blue), where an acoustic acceleration of $1\,\mathrm{m/s^2}$ is imposed. The density of air is specified as $1.2\,\mathrm{kg/m^3}$ and the speed of sound as 343.8 m/s. On the seats a frequency-dependent impedance is applied, where the impedance is calculated from tabulated values of absorption coefficients. The acoustic pressure is tracked at the driver's ear as output quantity. The finite element model consists of 1,372,332 degrees of freedom. The calculation of a single frequency line took approximately 69.5 s for the damped case on a Linux Cluster, using two 10-core Ivy Bridge Xeon E5-2680v2 CPUs (2.8 GHz, 128 GB RAM). The frequency range from 1 Hz to 300 Hz is simulated with a 1 Hz step to set up the reference data.

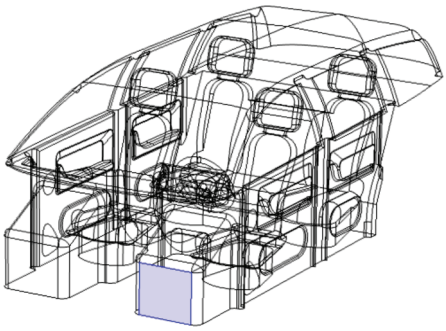


Figure 3.4: Car cavity geometry with porous seats with complex frequency-dependent behavior.

For this example, the starting point is a steady-state dynamic equation of the form

$$[\mathbf{K}(\omega) - \omega^2\mathbf{M}(\omega)]\mathbf{x} = \mathbf{f}, \tag{3.35}$$

where $\mathbf{M}, \mathbf{K} \in \mathbb{R}^{n\times n}$ are in this case frequency-dependent complex mass and stiffness matrices accounting for damping effects. Note that in this case $s = (\jmath\omega)^2$, in contrast to Section 3.4.2.

Figure 3.5 shows the original and fitted frequency response functions. The matrix-free approach starts with an initial calculation of two frequency lines, setting the boundaries of the frequency range of interest (5 Hz and 300 Hz). Thereafter, two additional frequency lines are computed and used to build a new ROM. When the requested accuracy of 1 % is met, both on $\varepsilon_{\mathrm{ROM,ROM}}$ and on $\varepsilon_{\mathrm{ROM,Full}}$, the algorithm terminates

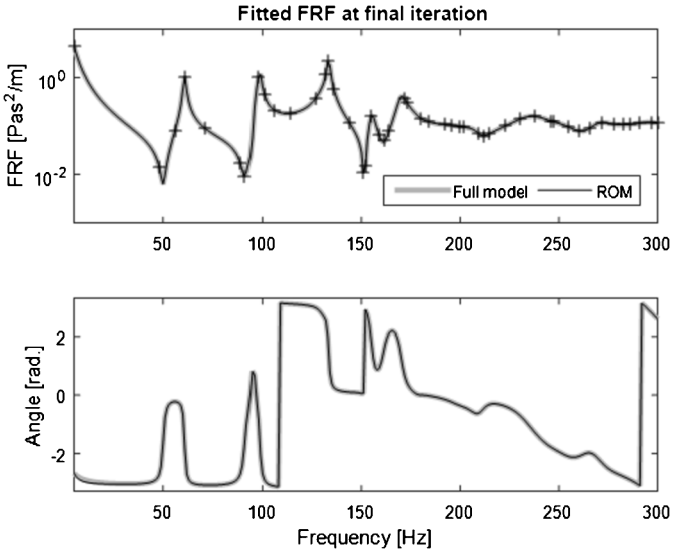


Figure 3.5: Acoustic frequency response function for car cavity with porous seats and final approximation for the acceleration transfer function; crosses indicate the interpolation frequencies.

(after 24 iterations, requiring only 48 frequency evaluations). A speedup factor of 6 is obtained for the acoustic frequency response assessment in this particular case.

3.4.4 Rational approximation

When $\mathbb{A}(s)$ is generally nonlinear, i. e., not polynomial or rational, linearizations can still be used on polynomial or rational approximations. Rational approximation, in particular, is very powerful, as we will now illustrate. The idea presented here is to approximate $\mathbb{A}(s)$ by a rational polynomial with d terms as in (3.28). Over the years, a number of approaches have been developed to perform these approximations. In [59], a Taylor expansion is used. In [52] and [50], $\mathbb{A}(s)$ is approximated by a truncated Padé series. In [50], windowing is used to cover the entire frequency range, and in [38], a spectral discretization is used. In [29], a rational approximation based on the adaptive Antoulas–Anderson method, also known as AAA and pronounced as "Triple A," is proposed.

Here, we will present the idea from [29], since it is an elegant and user-friendly way to find a rational approximation for the model. For the implementation details, we refer to [29]. Assume that $\mathbb{A}$ has the following form:

$$\mathbb{A}(s) = \mathbf{K} + s\mathbf{C} + s^2\mathbf{M} + \sum_{p=1}^{m} (\mathbb{A}_p + s\mathbb{B}_p) f_p(s),$$

with $f_p : \mathbb{C} \to \mathbb{C}$ a scalar function in s. First, the general nonlinearity is approximated by a rational function, expressed in barycentric rational form [41]. After the approximation, the matrix can be written as the rational matrix polynomial

$$\mathbb{A}(s) \simeq \mathbf{K} + s\mathbf{C} + s^2\mathbf{M} + \sum_{i=0}^{d-1}\left(\sum_{p=1}^{m}(\mathbb{A}_p + s\mathbb{B}_p)f_p(\zeta_i)\right)\frac{\omega_i/(\jmath\omega - \zeta_i)}{\sum_{k=0}^{d-1}\omega_k/(\jmath\omega - \zeta_k)},$$

where $\omega_i,\ i = 0,\dots,d-1$, are called the weights and ζ_i the support points. The AAA method chooses d, ω_i, ζ_i for $i = 0,\dots,d-1$ so that

$$\sup_{\omega\in[\omega_{\min},\omega_{\max}]}\left|f_p(\jmath\omega) - \sum_{i=0}^{d-1} f_p(\zeta_i)\frac{\omega_i/(\jmath\omega - \zeta_i)}{\sum_{k=0}^{d-1}\omega_k/(\jmath\omega - \zeta_k)}\right| \tag{3.36}$$

is below a given tolerance for $p = 1,\dots,m$. The AAA method determines the parameters automatically in very little time. Only the function values of f_p for the test set are required. There is no need to compute derivatives. The criterion (3.36) is discretized in a number of points on the frequency axis. The support points ζ_i are chosen using a greedy method. The minimization of the residual leads to the weights ω_i.

The following example is based on a system from [25, 37]. The results are reported in [28]. For this example, we consider the following time delay differential equation:

$$\frac{\partial v}{\partial t} = \nabla^2 v + \sum_{j=1}^{3} a_j v + bu,$$
$$q = \langle c, v\rangle.$$

We set the domain to $[0,\pi]\times[0,\pi]\times[0,\pi]$, and let

$$a_1(x,y,z) = 2\sin(\pi - x)\sin(y)\sin(z),$$
$$a_2(x,y,z) = 2\sin(x)\sin(\pi - y)\sin(z),$$
$$a_3(x,y,z) = 2\sin(x)\sin(y)\sin(\pi - z),$$

and $\tau_1 = 1, \tau_2 = 2, \tau_3 = 4$. For the domain discretization we use central differences with N discretization points in each spatial variable. After transforming to the frequency domain, we get

$$\begin{cases} (\mathbb{A}_0 - sI + \sum_{j=1}^{m}\mathbb{A}_j e^{-\tau_j s})\mathbf{x}(s) = -\mathbf{b}u(s), \\ H(s) = \mathbf{c}^* x(s), \end{cases} \tag{3.37}$$

where $s = \jmath\omega =$, $\mathbb{A}_j \in \mathbb{R}^{n\times n}$, $\mathbf{x}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, and $n = N^3$. We are interested in solutions in the range $\omega \in [0.01, 10]$. The resulting system matrices are symmetric. Furthermore, the input application vector $\mathbf{b}$ is chosen as a vector containing random values in $[0,1]$ and $\mathbf{c} = \mathbf{b}$. In the following, we show the relative error of the transfer function:

$$\epsilon(s) = \frac{|\widehat{H}(s) - H(s)|}{|H(s)|},$$

with

$$H(s) = \mathbf{c}^* \mathbb{A}(s)^{-1}\mathbf{b}, \quad \text{and} \quad \widehat{H}(s) = \widehat{\mathbf{c}}^* (\widehat{\mathbb{A}} - s\widehat{\mathbb{E}})^{-1}\widehat{\mathbf{b}}.$$

The error is computed over the test set ω_j, $j = 1, \ldots, 500$, with logarithmically spaced points in the interval $[10^{-2}, 10]$, as values closer to zero are of particular interest.

For the AAA approximation, a test set of size 10,000 is chosen over the interval $[0.01, 10]$. The AAA algorithm results in an approximation of the three nonlinear functions with $d = 22$.

We now compare the results of three numerical methods:

RKS: A rational Krylov sequence of order $K = 150$ obtained by using the three shifts $0\jmath, 5\jmath, 10\jmath$, each one 50 times. This leads to a linear reduced model.

LÖW: A linear reduced model using Loewner matrices is used of order $k = 40$ with equidistant interpolation points in $[0.01\jmath, 10\jmath]$.

HYB: A hybrid approach where the reduced model of order $K = 150$ from RKS is further reduced to order $k = 40$ by applying LÖW.

For $N = 40$, i. e., $n = 64{,}000$, the relative error for RKS of size $K = 150$, for LÖW for $k = 40$, and for HYB is plotted in Figure 3.6. From the figure, it is clear that the Loewner pencil leads to a lower reduced dimension for the same error level in the higher-frequency range. The hybrid approach produces a model of size 40 with an error that is the maximum of the errors of RKS and LÖW. The difference between the Loewner and hybrid approaches lies in the execution time. For the Loewner approach, the simple interpolation using Loewner matrices required 80 large sparse matrix factorizations, where for the rational Krylov and hybrid approaches only three matrix factorizations were performed. The cost of the Loewner approximation on the model of order 150 in the hybrid approach is therefore negligible. The computations were timed and averaged over three runs, on a machine with 64-bit Intel processor, 28 cores, 2.6 GHz processors, and 128 GB RAM. The hybrid approach required 384.7 seconds, where the Loewner approach required 1,814.9 seconds, which is a significant factor of 4.7 of the computation time for the hybrid approach.

A second example is described in [29]. This model was generated using a mesh from Siemens Industry Software and applied poro-elastic material properties. The following matrix-valued function describes the nonlinear behavior of the sound pressure inside a car cavity with porous seats, represented by a Johnson–Champoux–Allard equivalent fluid model [1]:

$$\mathbb{A}(\omega) = \mathbf{K}_0 + h_K(\omega)\mathbf{K}_1 - \omega^2(\mathbf{M}_0 + h_M(\omega)\mathbf{M}_1),$$

where $\mathbf{K}_0, \mathbf{K}_1, \mathbf{M}_0, \mathbf{M}_1 \in \mathbb{R}^{n\times n}$ with $n = 15{,}036$ are symmetric, positive semi-definite matrices and ω is the angular frequency. The nonlinear functions are given by

$$h_K(\omega) = \frac{\phi}{\alpha(\omega)}, \quad \alpha(\omega) = \alpha_\infty + \frac{\sigma\phi}{i\omega\rho_0}\sqrt{1 + i\omega\rho_0 \frac{4\alpha_\infty^2\eta}{\sigma^2\Lambda^2\phi^2}},$$

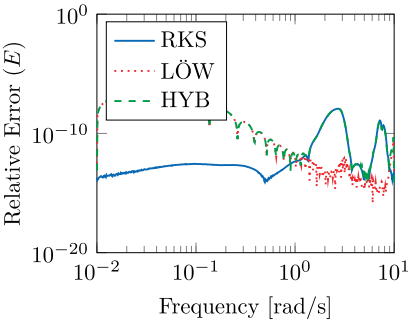


Figure 3.6: Relative error as a function of the frequency $f = \mathrm{Re}(s)$ for $n = 30$. Linear pencil of reduced dimension $K = 150$, Loewner with $k = 40$, and IRKA applied to the reduced pencil with $k = 40$.

and

$$h_M(\omega) = \phi\left(\gamma - \frac{\gamma - 1}{\alpha'(\omega)}\right), \quad \alpha'(\omega) = 1 + \frac{8\eta}{i\omega\rho_0\Lambda'^2 P_r}\sqrt{1 + i\omega\rho_0\frac{\Lambda'^2 P_r}{16\eta}},$$

with the parameters defined in Table 3.1.

Table 3.1: Constants of the car cavity model.

α_∞	1.7	σ	13500 kg m^{-3} s^{-1}	ϕ	0.98
η	$1.839 \cdot 10^{-5}$	Λ	$80 \cdot 10^{-6}$ m	Λ'	$160 \cdot 10^{-6}$ m
γ	1.4	ρ_0	1.213	P_r	0.7217

The AAA approximation was built by approximating h_K and h_M in the ω range $[50, 1000\pi]$. This led to a rational approximation with 12 support points with a relative error of 10^{-12} for both functions on this interval. Six shifts, namely, $1595.8\jmath$, $2173.92\jmath$, $2742.78\jmath$, $1125.87\jmath$, $3039.57\jmath$, and $50\jmath$, were selected using a greedy approach on the linear system's residual norm, where 20 Krylov iterations were performed for each shift. The dimension 120 was further reduced by applying POD in the frequency domain on this model to obtain an order of 77 for a relative tolerance of 10^{-8}.

3.5 Vibro-acoustic model order reduction

3.5.1 Stability-preserving model order reduction for vibro-acoustics

For coupled vibro-acoustic problems, the advantages of MOR have been demonstrated in the literature in the frequency domain (see, e. g., [23, 44]). However, popular

reduced-order techniques for frequency-domain analysis cannot directly be applied to time-domain analysis as these techniques do not necessarily preserve the stability of the original system [8]. While this loss of stability has no negative consequences on the frequency-domain analysis of the vibro-acoustic system, it may lead to a diverging response in the time domain.

3.5.1.1 Conditions for stability of a linear descriptor system in a one-sided projection

In [54] general conditions are derived that ensure to preserve the stability of a linear descriptor system in a one-sided projection. The semi-discretized system model in the time domain

$$\begin{aligned} \mathbb{E}\dot{\mathbf{x}}(t) &= \mathbb{A}\mathbf{x}(t) + \mathbf{b}\mathbf{u}(t), \\ H(t) &= \mathbf{c}^T\mathbf{x}(t) \end{aligned} \tag{3.38}$$

is critically stable if all of the generalized eigenvalues of the matrix pair $\{\mathbb{A}, \mathbb{E}\}$ have a negative real part [18]. It is proven that stability is preserved in the one-sided projection of the descriptor system in (3.38) if $\mathbb{E} = \mathbb{E}^T$ and if $\mathbb{E}$ is positive definite and if $\mathbb{A}$ is negative semi-definite. A matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is positive semi-definite if

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} \geq 0$$

for any nonzero vector $\mathbf{x} \in \mathbb{R}^n$.

3.5.1.2 *u-p* formulation and modified *u-ϕ* formulation

The vibro-acoustic model is generally described by (3.12), but depending on the exact formulation used for the model, different coupling matrices are present. Through a Galerkin approach for the finite element model for the acoustic problem, it follows that the acoustic system matrices $\mathbf{K}_a$, $\mathbf{M}_a$, and $\mathbf{C}_a$ are all symmetric positive (semi-)definite. Typically, $\mathbf{M}_a$ is of full rank, which makes it strictly positive definite. Whether $\mathbf{K}_a$ is strictly or semi-positive definite depends on the boundary conditions. Similar conclusions can be drawn for the structural finite element model and the corresponding system matrices $\mathbf{K}_s$, $\mathbf{M}_s$, and $\mathbf{C}_s$.

In the case $\mathbf{K}_a$ and $\mathbf{K}_s$ are positive semi-definite, the zero eigenvalues of these matrices correspond to rigid body modes of the system, manifesting themselves as system poles at the origin of the complex plane. These poles do not adversely affect the stability of the system and are disregarded in the subsequent analysis without loss of generality. This implies that we can consider $\mathbf{K}_s$ and $\mathbf{K}_a$ to be strictly positive definite.

In [54] the definiteness of the global system matrices of the coupled vibro-acoustic finite element model equation (3.13) are investigated. While the coupled damping matrix $\mathbf{C}_{up}$ is positive (semi-)definite, $\mathbf{K}_{up}$ and $\mathbf{M}_{up}$ are generally not positive definite due to the presence of $\mathbf{K}_c$ and $\mathbf{M}_c$. Consequently, also $\mathbb{E}_{up}$ and $\mathbb{A}_{up}$ of the equivalent descriptor system (3.38) are indefinite. Even though the vibro-acoustic system itself is stable, this stability is possibly lost in a regular one-sided projection.

Everstine [19] proposes an alternative, symmetric formulation for the vibro-acoustic problem which uses the scalar fluid velocity potential ϕ instead of the pressure p to describe the state of the fluid part of the vibro-acoustic system. This fluid velocity potential is defined by

$$p = -\rho\dot{\phi}, \tag{3.39}$$

with fluid density ρ, such that the vector containing the nodal pressure values in the fluid can be expressed as $\mathbf{p} = -\rho\dot{\boldsymbol{\phi}}$. The use of this u-ϕ formulation results in the symmetric system of equations which can be constructed from the u-p model matrices:

$$\mathbf{M}_{u\phi}\ddot{\mathbf{x}}_{u\phi} + \mathbf{C}_{u\phi}\dot{\mathbf{x}}_{u\phi} + \mathbf{K}_{u\phi}\mathbf{x}_{u\phi} = \mathbf{f}_{u\phi}, \tag{3.40}$$

with

$$\mathbf{M}_{u\phi} = \begin{bmatrix} \mathbf{M}_s & 0 \\ 0 & -\rho\mathbf{M}_a \end{bmatrix}, \quad \mathbf{C}_{u\phi} = \begin{bmatrix} \mathbf{C}_s & -\rho\mathbf{K}_c \\ -\rho\mathbf{K}_c^T & -\rho\mathbf{C}_a \end{bmatrix}, \quad \mathbf{K}_{u\phi} = \begin{bmatrix} \mathbf{K}_s & 0 \\ 0 & -\rho\mathbf{K}_a \end{bmatrix},$$
$$\mathbf{x}_{u\phi} = \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{bmatrix}, \quad \mathbf{f}_{u\phi} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_\phi \end{bmatrix}, \tag{3.41}$$

where $\dot{\mathbf{f}}_\phi = \mathbf{f}_a$. Note that also for $\mathbb{E}_{u\phi}$ and $\mathbb{A}_{u\phi}$, the matrices of the equivalent descriptor system are indefinite such that stability is not preserved in a one-sided projection.

It is shown in [54] that by changing the sign of the set of equations governing the acoustic degrees of freedom, resulting in the so-called modified u-ϕ, leads to global symmetric positive definite $\mathbf{M}_{mu\phi}$ and $\mathbf{K}_{mu\phi}$ and positive semi-definite $\mathbf{C}_{mu\phi}$, ensuring that the conditions for stability for the equivalent descriptor formulation are fulfilled. Even though the modified u-ϕ formulation lacks symmetry as compared to the standard u-ϕ formulation, a one-sided projection-based MOR preserves stability and the resulting ROMs are well suited for time-domain simulation.

3.5.1.3 Extended projection basis-preserving stability using the *u-p* formulation

As the poles of the u-p formulation and the modified u-ϕ formulation are equal, as shown in [54], stability should also be preserved when converting a system from the u-p formulation to the u-ϕ formulation and vice versa. In order to do so, the block-partitioned structure is required, which gets lost in the reduction process. By carefully

selecting the structure of the projection basis, the block structure within the second-order system matrices can be retained, allowing for this conversion between both formulations.

Let the regular projection basis $\mathbf{V} \in \mathbb{C}^{n \times k}$, obtained through any one-sided projection-based MOR method (e. g., Krylov subspace projection, modal truncation, ...), be given by

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_s \\ \mathbf{V}_a \end{bmatrix}, \tag{3.42}$$

with $\mathbf{V}_s \in \mathbb{C}^{n_s \times k}$ the part of the projection matrix that corresponds to the structural degrees of freedom and $\mathbf{V}_a \in \mathbf{C}^{n_a \times k}$ the part corresponding to the acoustic degrees of freedom. By augmenting this projection basis with zero blocks, an extended projection $\widetilde{\mathbf{V}}$

$$\widetilde{\mathbf{V}} = \begin{bmatrix} \mathbf{V}_s & 0 \\ 0 & \mathbf{V}_a \end{bmatrix} \tag{3.43}$$

is obtained that achieves block structure-preserving MOR for coupled systems, similar to the approach presented in [6]. The space spanned by the columns of $\widetilde{\mathbf{V}}$ contains the space spanned by the columns of $\mathbf{V}$, such that when using $\widetilde{\mathbf{V}}$ the reduced system will be at least as accurate as when using $\mathbf{V}$. For practical implementation it is recommended to orthogonalize $\widetilde{\mathbf{V}}$, which equates to orthogonalizing both $\mathbf{V}_s$ and $\mathbf{V}_a$. Also note that $\widetilde{\mathbf{V}}$ has twice as many columns as $\mathbf{V}$ such that the projected system will contain twice as many degrees of freedom. Since $\mathbf{V}_s$ and $\mathbf{V}_a$ may be rank-deficient this can be slightly ameliorated by using a rank-revealing algorithm for the orthogonalization of $\mathbf{V}_s$ and $\mathbf{V}_a$, but in our experience the total number of columns in $\widetilde{\mathbf{V}}$ still remains close to twice the number of columns in $\mathbf{V}$.

3.5.1.4 Work flow

Both approaches, starting from the modified u-ϕ formulation and the u-p formulation, are summarized in Figure 3.7.

It is advisable to use the modified u-ϕ formulation for stability-preserving MOR of coupled vibro-acoustic finite element models. Keeping the system in u-p formulation necessitates the use of an extended projection basis $\widetilde{\mathbf{V}}$, resulting in a ROM with possibly up to twice as many degrees of freedom. This ROM approach was recently extended towards time-stable coupled exterior vibro-acoustic finite element simulations [55].

3.5.1.5 Calculation example – car interior with vibrating roof

As an example, the vibro-acoustic behavior of a car interior is studied. The roof of the car is modeled as a flexible steel panel ($E = 200\,\text{GPa}$, $\nu = 0.3$, $\rho = 8000\,\text{kg/m}^3$,

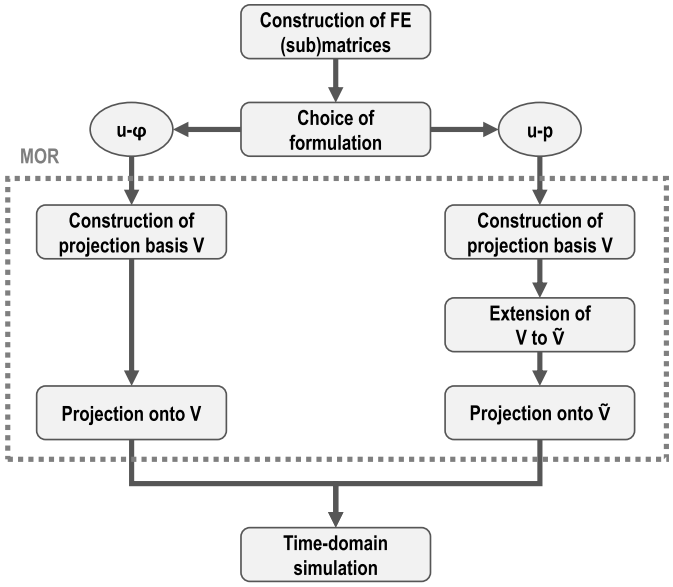


Figure 3.7: Work flow for stability preserving vibro-acoustic model order reduction [54].

and Rayleigh damping [$C_s = \alpha M_s + \beta K_s$] with $\alpha = 10$ and $\beta = 1 \cdot 10^{-7}$) with a thickness of 2 mm which is clamped at its boundaries and the interior is filled with air ($\rho = 1.225\,\mathrm{kg/m^3}$, $c = 340\,\mathrm{m/s}$). Linear elements are used with a resolution of at least six elements per wavelength up to 200 Hz. The resulting finite element mesh is shown in Figure 3.8. Near the plate the mesh is more refined since the wavelength of the bending waves in the structure is smaller than the acoustic wavelength at 200 Hz. A normal impedance boundary condition of twice the characteristic impedance of air ($Z_n = 2 \cdot 1.225 \cdot 340\,\mathrm{Pa\,s/m}$) is imposed on the surfaces of the seats in the interior cavity. All other boundaries of the acoustic domain are considered rigid ($v_n = 0\,\mathrm{m/s}$). A point force at (1.50, 0.08, 1.09) m excites the structure and the sound pressure is calculated at (0.73, 0.23, 0.68) m.

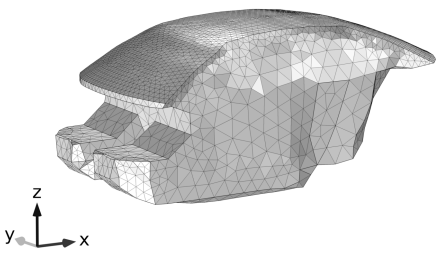


Figure 3.8: Finite element mesh of the car interior geometry [54].

The finite element model of the roof and acoustic cavity consist of 7,455 and 6,583 degrees of freedom, respectively. A one-sided Krylov subspace projection was constructed using SOAR [3, 58]. All ROMs consist of 60 degrees of freedom, except for the *u-p* projection with extended basis (equation (3.43)), which results in 120 degrees of freedom.

Figures 3.9 and 3.10 show the frequency response functions and the relative error with the full model, obtained applying model order reduction on each of the models. All ROMs are able to accurately describe the full system behavior in the frequency domain up to about 200 Hz. The *u-p* projected system with the extended basis is more accurate than the ROM that is obtained with the nonextended basis. Figure 3.11 shows the location of the poles of the ROMs in the complex plane. Both the *u-p* projection without extended basis and the projection of the system in standard *u-ϕ* formulation lead to unstable ROMs.

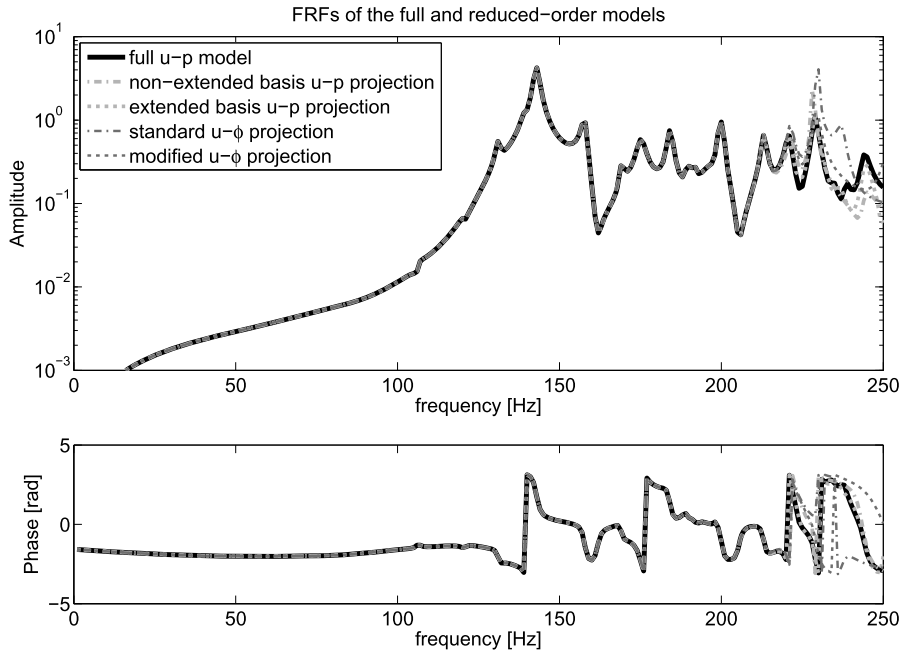


Figure 3.9: Frequency response function of the car cavity model calculated with the full-order model and the reduced-order models.

Finally, Figure 3.12 shows the convergence of the $\mathcal{H}_2$-norm of the relative frequency response function amplitude error ε of the different reduction methods, evaluated over the range of 1 Hz to 200 Hz. The method using an extended projection basis performs substantially worse than the other methods when comparing them in terms of accuracy per degree of freedom.

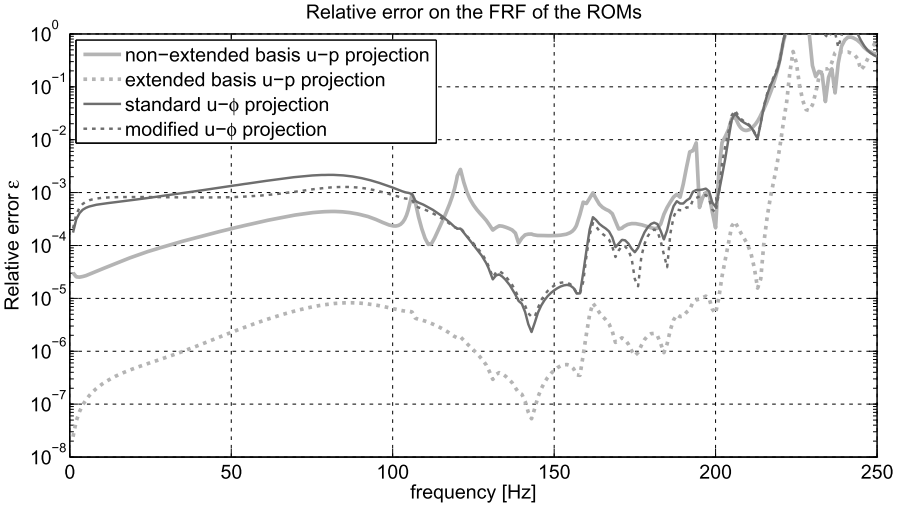


Figure 3.10: Relative error on the amplitude of the frequency response function of the reduced-order models of the car cavity.

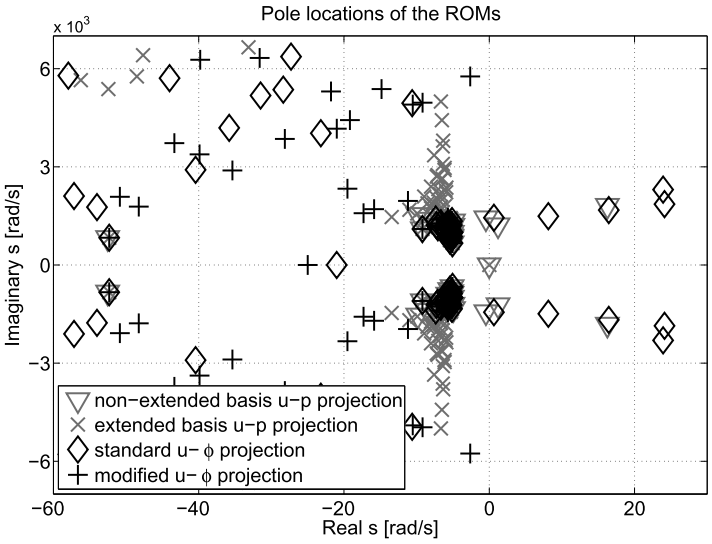


Figure 3.11: Poles of the different ROMs of the car cavity models. The nonextended basis *u-p* projection and standard *u-ϕ* projection result in unstable time-domain behavior.

3.6 Conclusions

We presented an overview of methods for model order reduction of dynamical systems arising in (coupled) acoustics and vibrations problems encountered in mechanical

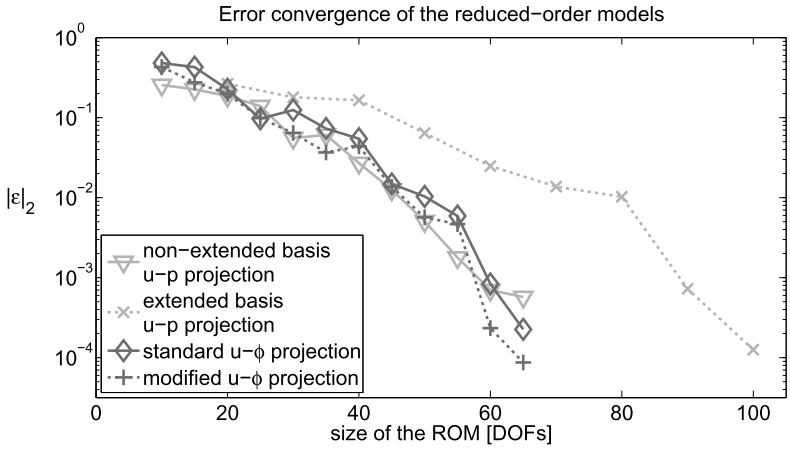


Figure 3.12: Efficiency of the MOR methods in terms of accuracy per degree of freedom for the car cavity vibro-acoustic model.

engineering. The choice of methods took into account the range of relevant requirements in practical engineering problems: computational efficiency, frequency limitation, stability, nonlinearity in the frequency variable, and preservation of underlying model structure.

For computational efficiency, the number of large sparse LU factorizations should be small. This can be achieved by using Krylov methods with a greedy selection of the interpolation points and using higher-order Hermite interpolation. We advocate two-level ROM methods, where a (rational) Krylov method with greedy selection of (a small number of) shifts leads to an ROM of moderate size. This ROM can be further reduced by applying balanced truncation, or methods in the Loewner framework.

In mechanical engineering models, the relevant structure can often be preserved by not building a frequency-linear ROM, but a model that respects the frequency dependency by projecting the system matrices on a well-chosen block of the Krylov vectors.

Bibliography

[1] J.-F. Allard and N. Atalla, *Propagation of sound in porous media: modelling sound absorbing materials*, John Wiley & Sons, 2009.

[2] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*, SIAM, Philadelphia, PA, USA, 2020.

[3] Z. Bai and Y. Su, Dimension reduction of second-order dynamical systems via a second-order Arnoldi method, *SIAM J. Matrix Anal. Appl.*, **26** (5) (2005), 1692–1709.

[4] A. Bayliss, M. Gunzburger, and E. Turkel, Boundary Conditions for the Numerical Solution of Elliptic Equations in Exterior Regions, *SIAM J. Appl. Math.*, **42** (1982), 430–451.

[5] A. Ben-Menahem and S. J. Singh, *Seismic Waves and Sources*, Springer Science & Business Media, 2012.
[6] P. Benner and L. Feng, Model order reduction for coupled problems, *Int. J. Appl. Comput. Math.*, **14** (2015), 3–22.
[7] P. Benner, P. Kürschner, and J. Saak, Frequency-limited balanced truncation with low-rank approximations, *SIAM J. Sci. Comput.*, **38** (1) (2016), A471–A499.
[8] P. Benner, V. Mehrmann, and D. C. Sorensen, *Dimension Reduction of Large-Scale Systems*, Springer, 2003.
[9] Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wilhelmus H. A. Schilders, and Luis Miguel Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.
[10] Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wilhelmus H. A. Schilders, and Luis Miguel Silveira (eds.), *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, 2020.
[11] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.*, **114** (2) (1994), 185–200.
[12] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. E. Hochstenbach, and W. H. A. Schilders, A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control, *J. Sound Vib.*, **332** (19) (2013), 4403–4422.
[13] D. T. Blackstock, *Fundamentals of physical acoustics*, John Wiley & Sons, New York, 2000.
[14] D. Colton and R. Kress, *Inverse acoustic and electromagnetic scattering theory*, 2nd, Springer-Verlag, Berlin, Heidelberg, New York, 1998.
[15] A. Craggs, The transient response of a coupled plate-acoustic system using plate and acoustic finite elements, *J. Sound Vib.*, **15** (4) (1971), 509–528.
[16] R. R. Craig and M. Bampton, Coupling of substructures in dynamic analysis, *AIAA J.*, **6** (7) (1968), 1313–1321.
[17] E. Deckers, S. Jonckheere, D. Vandepitte, and W. Desmet, Modelling techniques for vibro-acoustic dynamics of poroelastic materials, *Arch. Comput. Methods Eng.*, **22** (2015), 183–236.
[18] G. Duan, *Analysis and Design of Descriptor Linear Systems*, Springer, 2010.
[19] G. C. Everstine, A symmetric potential formulation for fluid-structure interaction. Letter to the editor, *J. Sound Vib.*, **79** (1981), 157–160.
[20] J. D. Ferry, *Viscoelastic Properties of Polymers*, John Wiley & Sons, New York, 1980.
[21] R. Freymann, *Advanced numerical and experimental methods in the field of vehicle structural-acoustics*, Hieronymus Buchreproduktions GmbH, 2000.
[22] S. W. Gaaf and E. Jarlebring, The infinite bi-Lanczos method for nonlinear eigenvalue problems, *SIAM J. Sci. Comput.*, **39** (5) (2017), S898–S919.
[23] U. Hetmaniuk, R. Tezaur, and C. Farhat, Review and assessment of interpolatory model order reduction methods for frequency response structural dynamics and acoustics problems, *Int. J. Numer. Methods Eng.*, **90** (2012), 1636–1662.
[24] W. C. Hurty, Dynamic analysis of structural systems using component modes, *AIAA J.*, **3** (4) (1965), 678–685.
[25] E. Jarlebring, K. Meerbergen, and W. Michiels, An Arnoldi like method for the delay eigenvalue problem, *SIAM J. Sci. Comput.*, **32** (6) (2010), 3278–3300.
[26] M. Lehner and P. Eberhard, A two-step approach for model reduction in flexible multibody dynamics, *Multibody Syst. Dyn.*, **17** (2-3) (2007), 157–176.
[27] X. Li, *Power flow prediction in vibrating systems via model reduction*. PhD thesis, Boston University, College of Engineering, 2004.

[28] P. Lietaert and K. Meerbergen, Comparing Loewner and Krylov based model order reduction for time delay systems, in *Proceedings of the European Control Conference*, p. 2018, 2018.
[29] P. Lietaert, K. Meerbergen, J. Perez, and B. Vandereycken, Automatic rational approximation and linearization of nonlinear eigenvalue problems. Technical Report arXiv:1801.08622, 2018. Submitted for publication.
[30] P. Lietaert, K. Meerbergen, and F. Tisseur, Compact two-sided Krylov methods for nonlinear eigenvalue problems, *SIAM J. Sci. Comput.*, **40** (5) (2018), A2801–A2829.
[31] D. Lu, Y. Su, and Z. Bai, Stability analysis of two-level orthogonal Arnoldi procedure, *SIAM J. Matrix Anal. Appl.*, **37** (1) (2016), 192–214.
[32] R. Lyon and R. DeJong, *Theory and application of statistical energy analysis*, 2nd, Butterworth-Heinemann, 1995.
[33] Mardochée Magolu monga Made, Robert Beauwens, and Guy Warzée, Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix, *Commun. Numer. Methods Eng.*, **16** (2000), 801–817.
[34] S. Marburg, Six boundary elements per wavelength: is that enough?, *J. Comput. Acoust.*, **10** (2002), 25–51.
[35] K. Meerbergen and Z. Bai, The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides, *SIAM J. Matrix Anal. Appl.*, **31** (4) (2010), 1642–1662.
[36] K. Meerbergen and J. P. Coyette, Connection and comparison between frequency shift time integration and a spectral transformation preconditioner, *Numer. Linear Algebra Appl.*, **16** (2009), 1–17.
[37] W. Michiels, G. Hilhorst, G. Pipeleers, T. Vyhlídal, and J. Swevers, Reduced modelling and fixed-order control of delay systems applied to a heat exchanger, *IET Control Theory Appl.* (2017).
[38] W. Michiels, E. Jarlebring, and K. Meerbergen, Krylov based model order reduction of time-delay systems, *SIAM J. Matrix Anal. Appl.*, **32** (4) (2011), 1399–1421.
[39] COMSOL Multiphysics, 2018. https://www.comsol.com/model/sedan-interior-acoustics-15013.
[40] MUMPS. MUlfrontal Massively Parallel Solver, 2009. http://graal.ens-lyon.fr/MUMPS/.
[41] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, The AAA algorithm for rational approximation, *SIAM J. Sci. Comput.*, **40** (3) (2018), A1494–A1522.
[42] J. Peeters and W. Michiels, Computing the H2 norm of large-scale time-delay systems, in *Proceedings of the IFAC Joint conference, Grenoble, 2013*, vol. 11, pp. 114–119, 2013.
[43] A. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications*, McGraw-Hill series in mechanical engineering, McGraw-Hill, 1981.
[44] R. S. Puri, D. Morrey, A. J. Bell, J. F. Durodola, E. B. Rudnyi, and J. G. Korvink, Reduced order fully coupled structural–acoustic analysis via implicit moment matching, *Appl. Math. Model.*, **33** (2009), 4097–4119.
[45] M. D. Rao, Recent applications of viscoelastic damping for noise control in automobiles and commercial airplanes, *J. Sound Vib.*, **262** (2003), 457–474.
[46] J. Rommes and N. Martins, Computing transfer function dominant poles of large-scale second-order dynamical systems, *SIAM J. Sci. Comput.*, **30** (4) (2008), 2137–2157.
[47] W. Desmet, S. Jonckheere, and E. Deckers, A matrix-free model order reduction scheme for vibro-acoustic systems including complex noise control treatments, in *Proceedings of Internoise 2018, 26–29 August 2018, Chicago, US*, 2018.
[48] W. Desmet, S. Jonckheere, and X. Li, A matrix-free model order reduction scheme for vibro-acoustic problems with complex damping treatments, in *Proceedings of ISMA2016, Leuven, 19–21 September 2016, Leuven, Belgium*, 2016.

[49] M. Saadvandi, K. Meerbergen, and W. Desmet, Parametric dominant pole algorithm for parametric model order reduction, *J. Comput. Appl. Math.*, **295** (2014), 259–280.
[50] R. D. Slone, R. Lee, and J.-F. Lee, Multipoint Galerkin asymptotic waveform evaluation for model order reduction of frequency domain FEM electromagnetic radiation problems, *IEEE Trans. Antennas Propag.*, **49** (10) (2001), 1504–1513.
[51] W. Soedel, *Vibrations of Shells and Plates*, CRC Press, 2004.
[52] Y. Su and Z. Bai, Solving rational eigenvalue problem via linearization, *SIAM J. Matrix Anal. Appl.*, **32** (1) (2011), 201–216.
[53] R. Van Beeumen, K. Meerbergen, and W. Michiels, Compact rational Krylov methods for nonlinear eigenvalue problems, *SIAM J. Matrix Anal. Appl.*, **36** (2) (2015), 820–838.
[54] A. van de Walle, F. Naets, E. Deckers, and W. Desmet, Stability-preserving model order reduction for time-domain simulation of vibro-acoustic FE models, *Int. J. Numer. Methods Eng.*, **109** (6) (2017), 889–912.
[55] S. van Ophem, O. Atak, E. Deckers, and W. Desmet, Stable model order reduction for time-domain exterior vibro-acoustic finite element simulations, *Comput. Methods Appl. Mech. Eng.*, **325** (2017), 240–264.
[56] K. Vergote, *Dynamic analysis of structural components in the mid frequency range using the wave based method, Non-determinism and inhomogeneities*. KULeuven, division PMA, PhD thesis 2012D03, 2012.
[57] O. Von Estorff, *Boundary Elements in Acoustics: Advances and Applications*, WITpress, 2000.
[58] S. Wyatt, *Issues in Interpolatory Model Reduction: Inexact Solves, Second-order Systems and DAEs*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2012.
[59] X. Xie, Z. Hui, S. Jonckheere, A. van de Walle, B. Pluymers, and W. Desmet, Adaptive model reduction technique for large-scale dynamical systems with frequency-dependent damping, *Comput. Methods Appl. Mech. Eng.*, **332** (2018), 363–381.
[60] Y. Yue and K. Meerbergen, Accelerating optimization of parametric linear systems by model order reduction, *SIAM J. Optim.*, **23** (2) (2013), 687–1370.
[61] Matthew J. Zahr, Philip Avery, and Charbel Farhat, A multilevel projection-based model order reduction framework for nonlinear dynamic multiscale problems in structural and solid mechanics, *Int. J. Numer. Methods Eng.* (2017).
[62] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, and P. Nithiarasu, *The Finite Element Method - The three volume set*, 6th, Butterworth-Heinemann, 2005.

Behzad Nouri, Emad Gad, Michel Nakhla, and Ram Achar

4 Model order reduction in microelectronics

Abstract: This chapter deals with the application of model order reduction (MOR) in the area of microelectronics. It mainly focuses on the diligent efforts of the MOR community in addressing one of the main challenges pertaining to circuit simulation, namely, the simulation of high-speed interconnects. A general framework for formulating the circuit equations that is commonly used in commercial circuit simulators is presented. Incorporation of high-speed interconnect structures within the general formulation of the circuit equations is described. Current challenges in the MOR of interconnect circuits with a large number of ports are presented along with some of the recent MOR techniques to handle this kind of circuits. In addition, techniques for the reduction of active stable circuits are reviewed with emphasis on guaranteeing the stability of the reduced circuits by construction. Several application examples are presented to highlight the performance and computational advantages attained by using MOR techniques within the circuit simulation environments.

Keywords: microelectronics, model order reduction, high-speed interconnect, multiport network, stability preservation

MSC 2010: 78A55, 62P30, 93A15, 78M34, 37M05, 34K20, 34H15

4.1 Introduction

The interest in large-scale model order reduction (MOR) in the area of microelectronics was mainly initiated in the community of electronic design automation in both the academic and industrial circles around the early 1990s. This interest was mainly spurred by the fast-paced technological development which allowed the very large-scale integration (VLSI) of millions of devices on a tiny chip of silicon. With such potential in the VLSI industry, computer-aided design tools needed to cope, to allow the designers to reach closure on their designs, with what is typically described as tight market windows.

The initial thrust for developing MOR in microelectronics was instigated by the push in the industry for operating at a higher speed. In fact, semi-conductor fabrication technology helped this push through enabling the reduction in device sizes (e. g., MOSFET transistors), ultimately leading to reduced processing time. Nonetheless, it was the interconnect wires between those devices that represented the dominant fac-

Behzad Nouri, Michel Nakhla, Ram Achar, Dept. of Electronics, Carleton Univ., Ottawa, ON, Canada, K1S 5B6, e-mails: sbnouri@doe.carleton.ca, msn@doe.carleton.ca, achar@doe.carleton.ca
Emad Gad, School of Electrical Engineering and Computer Science, Univ. of Ottawa, Ottawa, ON, Canada, K1N 6N5, e-mail: egad@uottawa.ca

https://doi.org/10.1515/9783110499001-004

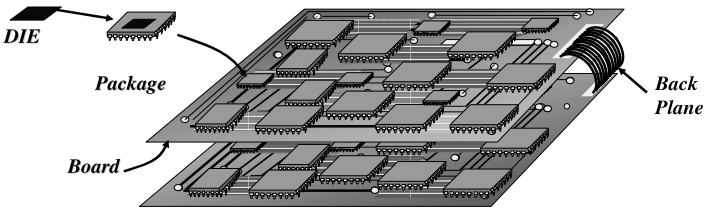


Figure 4.1: Electrical interconnects are encountered at all levels of the design hierarchy.

tor in dictating how high the operating frequencies can go. The reason for this is that at high frequency, one can no longer treat interconnects (referred to, henceforth, as high-speed interconnects) as simple wires. Rather, their accurate modeling would necessitate using the theory of electromagnetic wave propagation, thereby leading to large-size networks, whose simulations became oftentimes cumbersome.

This chapter serves the following two main purposes.

1. Describing the context in which the need for MOR arose in microelectronics. To achieve this purpose, Section 4.2 presents the general circuit formulation in a compact mathematical form. This section also highlights the difficulty and the high cost associated with fitting the high-speed interconnects in such formulation.
2. Presenting an overview of the applications of MOR to address the issues of high-speed interconnects in microelectronics at various levels of the design hierarchy. As illustrated in Figure 4.1, high-speed interconnects are encountered at all levels of the design hierarchy, be it on the silicon (on-chip or die), package, board, or backplanes level. The application of MOR to high-speed interconnects was adapted based on where (which level in the hierarchy) the interconnects are used.

In writing this chapter, the authors focused mainly on the application of MOR to address the challenges of high-speed interconnects in microelectronics. Indeed, from a historical perspective, it was those challenges that ushered in the introduction of MOR to the area of microelectronics. Nevertheless, the application of MOR in microelectronics is by no means limited to high-speed interconnects. In fact, it was the reported success in that area that brought it to the attention of the electronic design automation community at large and prompted its application to the general area of circuit simulation. For example, efforts to extend MOR to nonlinear circuits and linear time-varying circuits date back to the end of the 1990s (e. g., [55, 101, 94, 72, 99, 27, 117, 46, 45, 74]). Another application of MOR in microelectronics was proposed to enable the reduced system to capture the original large system along the dimensions spanned by several circuit parameters, leading to the parameterized MOR (PMOR), of which the works [54, 24] are notable examples.

4.2 Formulation of circuits with high-speed interconnects in the time domain

This section presents the modified nodal analysis (MNA) approach that is used by virtually all commercial circuit simulators to represent general circuits in the mathematical domain. The presentation of this topic is carried out in two phases. First, the MNA formulation for circuits with only lumped elements is considered in Section 4.2.1. Next, Section 4.2.2 considers incorporating the high-speed interconnects in the general circuit formulation, dwelling upon the challenges therein, in order to pave the way to show the indispensable role that MOR plays in this regard.

4.2.1 Formulation of circuits with lumped elements

The presence of nonlinear elements in virtually all circuit designs mandates that the natural domain for mathematically describing general circuits is the time domain. A widely adopted time-domain formulation is the MNA approach [57, 116, 82]. The MNA formulation is derived through:

1. Writing the Kirchhoff current law at each node.
2. Expressing the currents in the circuit elements in terms of the node voltages using some form of Ohm's law (i. e., the constitutive relation of the element). The node voltages are then considered as the unknowns in the circuit formulation to be solved for.
3. Contriving a special representation for the elements which do not have a simple Ohm's law representation, such as voltage sources, or elements whose constitutive relation requires integration in the time domain, e. g., inductors. The currents in those elements are appended to the set of unknowns to be solved for, along with node voltages. Such representation is typically termed "impedance representation."
4. Representing some nonlinear elements, such as nonlinear capacitors or nonlinear inductors, by the charge/flux-oriented formulation in which the charge or flux are included in the unknowns [82].

By using the above steps, a large system of equations is generated and assembled, taking the form of a system of a mixed set of differential and algebraic equations (DAEs) that is known as the MNA formulation. Thus, a general circuit with lumped elements, such as resistors, inductors, capacitors, etc., is described by the following DAE:

$$\mathbf{C}\frac{d\mathbf{x}(t)}{dt} + \mathbf{G}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t), \tag{4.1}$$

where

- $\mathbf{C}, \mathbf{G} \in \mathbb{R}^{N \times N}$ are real matrices describing the memory and memoryless elements in the network, respectively;
- $\mathbf{x}(t) \in \mathbb{R}^N$ is a vector whose components are time-dependent waveforms of (1) node voltages (2) currents in elements with impedance representation and (3) charges/fluxes in elements with charge- or flux-based representation (e. g., nonlinear capacitors or inductors);
- $\mathbf{b}(t) \in \mathbb{R}^N$ is a vector of voltage and current waveforms of independent voltage and current sources representing the external stimulus of the circuits;
- $\mathbf{f}(\mathbf{x}(t)) \in \mathbb{R}^N$ is a vector whose entries are scalar nonlinear functions $\mathbb{R}^N \to \mathbb{R}$ that represent the nonlinear elements in the circuit; and
- N is the total number of circuit variables in the MNA formulation.

To solve (4.1) for the circuit variables $\mathbf{x}(t)$, various time marching techniques can be used. Examples of these techniques are the trapezoidal rule or its high-order variants such as the backward-differentiation formulas [116], or the Obreshkov-based methods proposed in [47, 121, 31, 32, 73].

With typically thousands or millions of circuit elements in modern circuit designs, the construction of the MNA formulation can only be performed automatically. The approach used in this task is to attach to each circuit element the so-called stamp of the element, which describes, so to speak, the "footprint" that the element leaves on the mathematical structures (matrices and vectors) of the MNA formulation. The circuit is described by a text file typically known as a "netlist." A netlist can be either manually edited by the user or, as in most cases, extracted automatically from circuit schematics created by a plethora of commercial software packages such as OrCAD PSpice Designer, CADENCE Virtuoso, or MultiSim. The netlist file represents the circuit as a sequence of elements, with each line on the netlist file (or a group of lines separated by a specially dedicated character) semantically describing one circuit element at a time. Netlists of large circuits can easily grow to millions of lines. Typically, the software responsible for automatically constructing the MNA formulation reads the netlist one line at a time or, more recently, multiple lines at a time using parallel processing. With the syntax used for each line completely identifying the type of circuit element (e. g., a capacitor), its value (in Farads), and the circuit nodes to which it is connected, the MNA formulation software proceeds, guided by the preprogrammed various elements stamps, to add the contribution of each element to the matrices or vectors in the MNA formulation in (4.1).

To provide more insight into this process, we include in the following subsections a limited sample of circuit elements and their stamps.

4.2.1.1 Resistor's stamp

Consider a resistor with resistance R connected between two circuit nodes labeled on the netlist `n_j` and `n_jprime` (as shown in Figure 4.2(a)), and let j and $j' \in \mathbb{N}$ be the two (integer) indices associated with the two voltage waveforms of those nodes.[1] The line representing the resistor, such as $R = 30\,\Omega$ on a netlist may appear as shown by the following line:

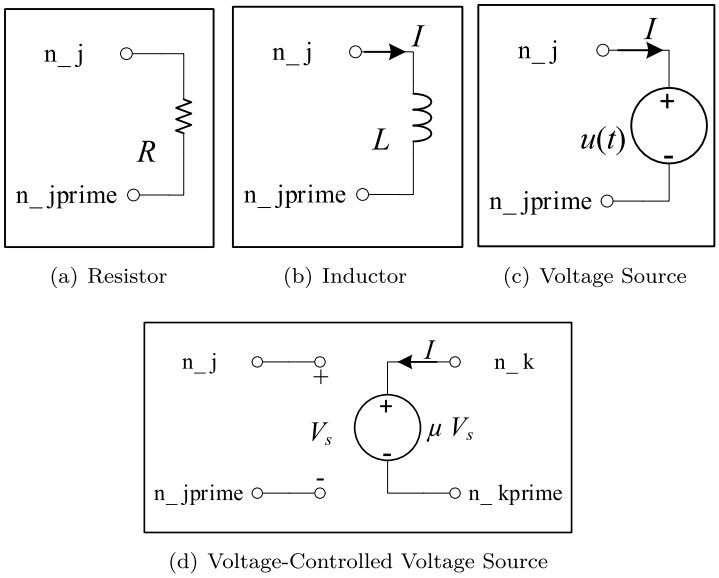
```
R_resistor_label   n_j   n_jprime   R=30
```

The stamp of such a resistor is captured using the formulation

$$\mathbf{G} \leftarrow \mathbf{G} + \left(\begin{array}{c} \\ \text{row } j \\ \text{row } j' \end{array} \begin{array}{c} \begin{array}{cc} \text{column } j & \text{column } j' \end{array} \\ \begin{bmatrix} 1/R & -1/R \\ -1/R & 1/R \end{bmatrix} \end{array} \right).$$

4.2.1.2 Inductor's stamp

In the same style used for the resistor stamp above, the stamp of an inductor (Figure 4.2(b)) with inductance L connected between two nodes labeled on the netlist `n_j`



(a) Resistor (b) Inductor (c) Voltage Source

(d) Voltage-Controlled Voltage Source

Figure 4.2: A sample of common circuit elements.

1 In other words $x_j(t)$ = voltage at node `n_j`, and $x_{j'}(t)$ = voltage at node `n_jprime`.

and `n_jprime`, with corresponding indices j and $j' \in \mathbb{N}$, may be shown by

$$\mathbf{G} \leftarrow \mathbf{G} + \begin{array}{c} \\ \text{row } j \\ \text{row } j' \\ \text{row } m{+}1 \end{array} \begin{array}{c} \begin{array}{cc} \text{column } j & \text{column } j' \end{array} \quad \text{column } m{+}1 \\ \left[\begin{array}{cc|c} & & +1 \\ & & -1 \\ \hline +1 & -1 & \end{array}\right] \end{array},$$

$$\mathbf{C} \leftarrow \mathbf{C} + \begin{array}{c} \\ \text{row } j \\ \text{row } j' \\ \text{row } m{+}1 \end{array} \begin{array}{c} \begin{array}{cc} \text{column } j & \text{column } j' \end{array} \quad \text{column } m{+}1 \\ \left[\begin{array}{cc|c} & & \\ & & \\ \hline & & -L \end{array}\right] \end{array}.$$

The above formulation should be read as follows: The stamp of an inductor requires appending the inductor's current waveform to the set of the MNA variables $\mathbf{x}(t)$. Thus, assuming that there are currently m variables in the formulation, the matrices $\mathbf{G}$ and $\mathbf{C}$ will be appended by an extra row and extra column $(m+1)$ to account for the current in the inductor and its associated constitutive equation. The syntax of the line describing an inductor with $L = 3\,\mu\text{H}$ typically appears as

```
L_inductor_label   n_j   n_jprime   L=3uH
```

4.2.1.3 Stamp of an independent voltage source

Let an independent voltage source (seen in Figure 4.2(c)) have a voltage $u(t)$, and be connected between two nodes labeled `n_j` (positive polarity) and `n_jprime` (negative polarity) with indices j and j', respectively. The stamp of this voltage source appears in both the source vector $\mathbf{b}(t)$ and the matrix $\mathbf{G}$, as illustrated by the following formulation:

$$\mathbf{b}(t) \leftarrow \mathbf{b}(t) + \begin{array}{c} \\ \text{row } j \\ \text{row } j \\ \text{row } m{+}1 \end{array} \begin{array}{c} \\ \left[\begin{array}{c} \\ \hline u(t) \end{array}\right] \end{array},$$

$$\mathbf{G} \leftarrow \mathbf{G} + \begin{array}{c} \\ \text{row } j \\ \text{row } j' \\ \text{row } m{+}1 \end{array} \begin{array}{c} \begin{array}{cc} \text{column } j & \text{column } j' \end{array} \quad \text{column } m{+}1 \\ \left[\begin{array}{cc|c} & & +1 \\ & & -1 \\ \hline +1 & -1 & \end{array}\right] \end{array}.$$

As can seen from the above formulation, the current waveform in the source enters in the set of MNA variables as indicated by the extra row and column in the MNA matrix $\mathbf{G}$ and source vector $\mathbf{b}(t)$.

4.2.1.4 Stamp of a voltage-controlled voltage source

Figure 4.2(d) shows a voltage-controlled voltage source (VCVS). This circuit element has four nodes (with indices denoted j, j', k, k') and its stamp requires that the current in the controlled source (μV_s in Figure 4.2(d)) be included in the vector of MNA variables $\mathbf{x}(t)$. The following formulation can visualize the stamp of VCVS:

$$\mathbf{G} \leftarrow \mathbf{G} + \begin{array}{c} \\ \text{row } j \\ \text{row } j' \\ \text{row } k \\ \text{row } k' \\ \text{row } m{+}1 \end{array} \begin{array}{c} \begin{array}{ccccc} \text{column } j & \text{column } j' & \text{column } k & \text{column } k' & \text{column } m{+}1 \end{array} \\ \left[\begin{array}{cccc|c} & & & & \\ & & & & \\ & & & & +1 \\ & & & & -1 \\ \hline -\mu & \mu & +1 & -1 & \end{array}\right] \end{array}.$$

A typical netlist line representing a VCVS with $\mu = 0.4$ is shown below.

```
E_VCVS_label n_j  n_jprime  n_k  n_kprime  0.4
```

4.2.2 Incorporating high-speed interconnects in circuit formulation

When the circuit includes (in addition to the above-mentioned lumped elements) a subcircuit or subnetwork that is characterized as a high-speed interconnect circuit, the mathematical formulation given by (4.1) will have to be amended with a new term that significantly alters the nature of the formulation. The reason for that is that those networks have their physics grounded in the theory of electromagnetic propagation, which is often described by Maxwell's equations. Under certain assumptions, more specifically, a quasi-transverse electromagnetic wave propagation mode [91], the voltages and currents on those network are properly described as functions of time t and a spatial variable z that denotes the spatial distance along the line.

To further elucidate the above paragraph, Figure 4.3 shows the physical structure of a multiconductor interconnect with a ground plane, and Figure 4.4 shows its schematic representation as a network of m conductors with $2m$ ports.

The schematic representation highlights that voltages and currents are no longer localized to a certain node or a certain branch as in the case of lumped elements, but are rather distributed along the lines as a function of both time t and distance z. Thus, the constitutive relation between the voltage and currents, in this case, is no longer an algebraic or simple differential relation, but rather takes the form of a set of partial differential equations (PDEs), in which derivatives of the voltages and currents with respect to t and z appear.

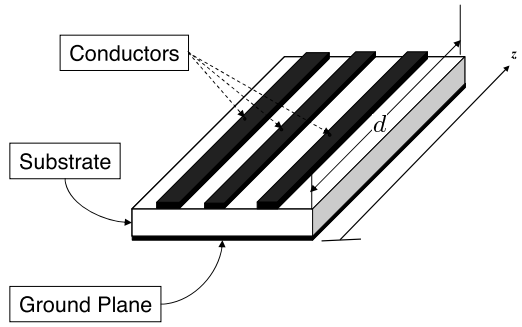


Figure 4.3: A physical representation for a high-speed interconnect structure.

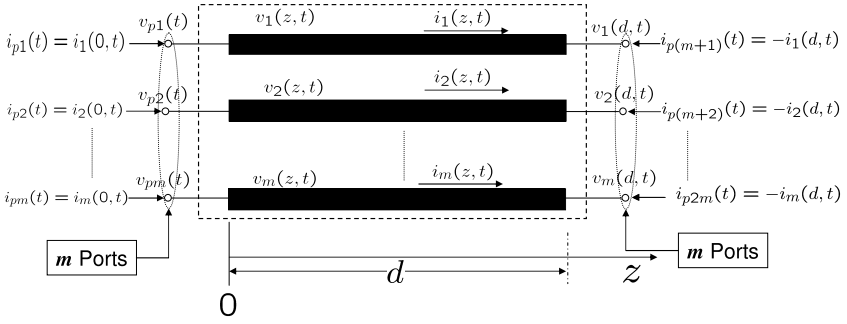


Figure 4.4: A schematic representation for the high-speed interconnect structure shown in Figure 4.3.

As a consequence of this fact, an attempt to develop a *compact* stamp, in the same spirit as was shown for conventional lumped elements listed above, would be successful only if it is carried out in the Laplace or frequency domain. This is because, in the Laplace domain, derivative with respect to t is replaced by the Laplace-domain variable s leaving only the derivative with respect to z, therefore, transforming the PDE to an ordinary differential equation (ODE) in z, the solution of which can be written analytically using the matrix exponential function.

Unfortunately, such a stamp comes with a complex dependence on the Laplace-domain variable (s), which makes deriving the required time-domain stamp a complex (if not impossible) task. Indeed, the only possible way to incorporate this stamp in the time-domain MNA formulation of (4.1) is to add another term, which involves a convolution with $\mathbf{x}(t)$. With a convolution term present in the MNA formulation, the utilization of time marching to simulate the circuit becomes very inefficient.

The approaches that are often used to circumvent this difficulty, i. e., to derive a time-domain stamp for the interconnect element, can be developed, but usually at the cost of compactness. Those approaches, which are collectively known under the term "macromodeling," seek to convert the PDE into a time-domain ODE that can be easily stamped into the MNA formulation of (4.1).

Macromodeling techniques typically approach the problem by discretizing the PDE along the spatial dimension z, approximating the z derivative using an appropriate approximation operator. Examples of these techniques are matrix rational approximation (MRA) [28, 29] of the exponential function and delay-based approaches such as method of characteristics (MoC) [53] and the DEPACT algorithm [83]. Macromodeling, used in that sense, solves the issue of stamping the high-speed interconnect in the MNA formulation, but at the cost of compactness as it produces a large network of extra circuit elements. The goal of using MOR in high-speed interconnects is to address the complexity of simulating the circuit with large macromodels.

The next subsection presents the macromodeling based on lumped segmentation, while the rest of the chapter will review the application of various MOR approaches used to reduce the complexity of the models.

4.2.3 Time-domain macromodeling based on discretization

Discretization techniques represent a very straightforward approach to overcome the above difficulties and incorporate high-speed interconnects in circuit simulators. Following the central idea of these techniques, one first needs to introduce the parameters of the line, which are the resistance (R), inductance (L), conductance (G), and capacitance (C) per unit length. In the case of an interconnect with m conductors, those parameters are $m \times m$ matrices.

Discretization proceeds by dividing the line into segments of length Δz, chosen to be a small fraction of the shortest wavelength in the driving signal.[2] If the length of each of these segments is electrically small (i. e., compared to the shortest wavelength), then each segment can be replaced by the model shown in Figure 4.5.

It is of practical interest to know how many of these segments are required to reasonably approximate the interconnect. For illustration consider a lossless line, i. e., $R = 0$ and $G = 0$, with only LC elements which can be viewed as a low-pass filter. For a reasonable approximation, this filter must pass at least some multiples of the highest

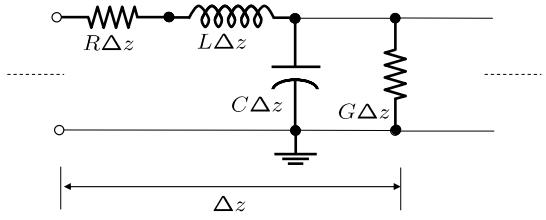


Figure 4.5: Modeling a segment of a single-conductor transmission line using lumped circuit elements.

2 The shortest wavelength is obtained by dividing the speed of light in the medium of the interconnect by the highest frequency in the propagating signal.

frequency $f_{\max}$ of the propagating signal (say, 10 times, $f_0 \geq 10 f_{\max}$). To relate these parameters, we make use of the 3 dB passband frequency of the LC filter given by [63]

$$f_0 = \frac{1}{\pi\sqrt{LdCd}} = \frac{1}{\pi\tau d}, \tag{4.2}$$

where d is the length of the line and $\tau = \sqrt{LC}$ represents the delay per unit length (p. u. l.). Typically, we set $f_{\max} = 0.35/t_r$, where t_r is the rise time of the signal. Using (4.2), we can express the relation $f_0 \geq 10 f_{\max}$ in terms of the delay of the line and the rise time as $1/(\pi\tau d) \geq 10 \times 0.35/t_r$ or $t_r \geq 3.5(\pi\tau d) \approx 10\tau d$. In other words, the delay allowed per segment is approximately $t_r/10$. Hence the total number of segments (P) needed to represent the total delay of τd is given by

$$P = \tau d/(t_r/10) = 10\tau d/t_r. \tag{4.3}$$

As an example, consider a digital signal with rise time $t_r = 0.2$ ns propagating on a lossless wire of length 10 cm with a p. u. l. delay of 70.7 ps, which can be represented by a distributed model with p. u. l. parameters of $L = 5$ nH/cm and $C = 1$ pF/cm. If the same transmission line were to be represented by lumped segments, one needs $P \approx 35$ segments.

One of the major drawbacks of the above approach is that it requires a large number of sections, especially for circuits with many multiple conductors, high operating speeds, and sharper rise times. This leads to large circuit sizes and the simulation becomes CPU-inefficient.

4.3 Model order reduction: application perspective

A methodological approach to describe the application of MOR in microelectronics can begin by considering the MNA formulation for general time-invariant large-scale continuous circuits as presented in (4.1). In the absence of nonlinear elements, a general linear circuit with several input and output terminals can be described by neglecting the nonlinear term in (4.1). Additional mathematical terms are also included to delineate their input-to-output behavior. In particular, to describe a network with only linear elements, having n_{in} inputs and n_{out} outputs, the MNA formulation takes on the following form:

$$\mathbf{C}\frac{d}{dt}\mathbf{x}(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t), \tag{4.4a}$$

$$\mathbf{y}(t) = \mathbf{L}^{\mathrm{t}}\mathbf{x}(t), \tag{4.4b}$$

where $\mathbf{x}(t) \in \mathbb{R}^N$, $\mathbf{C}, \mathbf{G} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times n_{\text{in}}}$, and $\mathbf{L} \in \mathbb{R}^{N\times n_{\text{out}}}$.

For engineering applications such as microelectronics, microelectromechanical systems, and electromagnetism, the size of the system in (4.4) can be very large, in

the range of millions. To reduce the computational cost associated with such large networks, MOR [6, 14, 113, 105, 13] has proven to be an effective tool to develop simpler models that capture the essential features of the given large system and accurately approximate their input-to-output behavior. The resulting model replaces the complex original system to ensure fast and reliable simulations.

4.3.1 Explicit moment-matching

Given a linear time-invariant (LTI) circuit in MNA form (4.4), its input-to-output relationship under the assumption of zero initial state can be described in the Laplace domain as $\mathbf{Y}(s) = \mathbf{H}(s)\,\mathbf{U}(s)$, where

$$\mathbf{H}(s) = \mathbf{L}^{\mathrm{t}}(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B} \tag{4.5}$$

is a complex-valued transfer function. The Laplace variable "s" is in the form of a complex frequency. The importance of transfer functions in characterizing LTI systems stems from the fact that the unit-impulse response of the system in the time domain can be recovered by the inverse Laplace transform of $\mathbf{H}(s)$.

A natural idea for MOR is to construct a reduced-order model such that the Taylor series expansion of its transfer function $\hat{\mathbf{H}}(s)$, with respect to s, matches a number of the leading terms in the Taylor series expansion of the original $\mathbf{H}(s)$ (4.5). Assume $\mathbf{G}$ is invertible; then DC ($s = 0$ rad/s) can be a prompt choice for the expansion point. Taylor series expansion of $\mathbf{H}(s)$ around $s = 0$ can be obtained by expanding $\mathbf{X}(s)$ as

$$\mathbf{H}(s) = \mathbf{L}^{\mathrm{t}}\mathbf{X}(s) = \mathbf{L}^{\mathrm{t}} \sum_{i=0}^{\infty} \mathbf{M}_i\, s^i, \tag{4.6}$$

where the coefficient of s^i in the expansion is called the i-th moment at $s = 0$.

4.3.2 Moment computation in MNA formulation

Applying Laplace transform to (4.4a), assuming the unit-impulse excitations at the inputs, and expanding its $\mathbf{X}(s)$ using Taylor series at $s = 0$, we obtain

$$(\mathbf{G} + s\mathbf{C})(\mathbf{M}_0 + \mathbf{M}_1 s + \mathbf{M}_2 s^2 + \cdots + \mathbf{M}_i s^i + \cdots) = \mathbf{B}. \tag{4.7}$$

Moments of the MNA variables $\mathbf{X}(s)$ are derived in a recursive form as

$$\mathbf{M}_{i+1} = \mathbf{A}\mathbf{M}_i \tag{4.8}$$

where

$$\mathbf{M}_0 = \mathbf{R}, \qquad \mathbf{R} = \mathbf{G}^{-1}\mathbf{B}, \qquad \mathbf{A} = -\mathbf{G}^{-1}\mathbf{C}. \tag{4.9}$$

Computation of moments only need one LU decomposition of $\mathbf{G}$. The moment computation formula can be equivalently written as

$$\mathbf{M}_i = \mathbf{A}^i\mathbf{R}, \qquad \text{for} \quad i = 0, 1, 2, \ldots. \tag{4.10}$$

4.3.3 Asymptotic waveform evaluation

Asymptotic waveform evaluation (AWE) [98, 96, 21] was the first MOR method that was based on the moment-matching idea. Following the steps of AWE, first, $2m$ leading moments of the circuit are obtained through a fast recursive moment computation. These $2m$ explicit moments (4.10) are used to find m poles and residues of the m-th-order macromodel via the Padé approximation.

The explicit moment-matching approaches, namely, AWE, suffer from numerical limitations and become ineffective due to the inherent ill-conditioning nature of explicit moment generation. Thus, these methods can be used for small-order approximations which require only a few moments to be matched.

4.3.4 Projection for order reduction

An elegant solution to overcome the innate ill-conditioning of the methods based on explicit moment generation is to use projection-based MOR methods, which are based on implicit moment-matching [34, 38, 40, 111, 88, 20, 50, 25]. The enabling idea is to first reduce the number of MNA variables by projecting $\mathbf{x}(t)$ to smaller subspace, as shown in (4.11). For this, an orthogonal projection matrix $\mathbf{V} \in \mathbb{R}^{N\times m}$ with $m \ll N$ is used as

$$\mathbf{x}(t) = \mathbf{V}\hat{\mathbf{x}}(t). \tag{4.11}$$

Then, a left-projection matrix $\mathbf{U} \in \mathbb{R}^{N\times m}$ is used, in general, to reduce the size of the resulting circuit equation as

$$\mathbf{U}^{\mathrm{t}}\mathbf{C}\mathbf{V}\frac{d}{dt}\hat{\mathbf{x}}(t) + \mathbf{U}^{\mathrm{t}}\mathbf{G}\mathbf{V}\hat{\mathbf{x}}(t) = \mathbf{U}^{\mathrm{t}}\mathbf{B}\mathbf{u}(t). \tag{4.12}$$

Given the projection matrices $\mathbf{U}$ and $\mathbf{V}$ ($\in \mathbb{R}^{N\times m}$), the reduced-order model for the MNA formulation in (4.4), obtained through a Petrov–Galerkin projection scheme [6], can be formalized as

$$\hat{\mathbf{C}}\frac{d}{dt}\hat{\mathbf{x}}(t) + \hat{\mathbf{G}}\hat{\mathbf{x}}(t) = \hat{\mathbf{B}}\mathbf{u}(t), \tag{4.13a}$$

$$\mathbf{y}(t) = \hat{\mathbf{L}}^{\mathrm{t}}\hat{\mathbf{x}}(t), \tag{4.13b}$$

where

$$\hat{\mathbf{C}} = \mathbf{U}^{\mathrm{t}}\mathbf{C}\mathbf{V} \in \mathbb{R}^{m\times m}, \qquad \hat{\mathbf{G}} = \mathbf{U}^{\mathrm{t}}\mathbf{G}\mathbf{V} \in \mathbb{R}^{m\times m}, \tag{4.13c}$$

$$\hat{\mathbf{B}} = \mathbf{U}^{\mathrm{t}}\mathbf{B} \in \mathbb{R}^{m\times n_{\mathrm{in}}}, \qquad \hat{\mathbf{L}}^{\mathrm{t}} = \mathbf{L}^{\mathrm{t}}\mathbf{V} \in \mathbb{R}^{n_{\mathrm{out}}\times m}. \tag{4.13d}$$

The model reduction methods differ in the choice of their projection matrices $\mathbf{U}$ and $\mathbf{V}$. For the techniques based on the idea of split congruence transformations [65], such as passive reduced-order interconnect macromodeling algorithm (PRIMA) [88], both projection matrices span the same subspace and thus a single projection matrix is used, $\mathbf{U} = \mathbf{V}$.

Computation of the orthogonal basis $\mathbf{V}$ is an essential step in the split congruence transformation-based reduction process and can be the main differentiation factor between the reduction methods.

4.3.5 Krylov subspace techniques

The Krylov subspace is defined as a subspace of the space spanned with the columns of block moments of circuit response, as given in (4.14). Let $\mathbf{V} \in \mathbb{R}^{m\times n}$ be the sought orthogonal basis matrix spanning the Krylov subspace as

$$\mathrm{colspan}(\mathbf{V}) = \mathcal{K}\boldsymbol{r}(\mathbf{A}, \mathbf{R}, q) = \mathrm{colspan}\{\mathbf{R}, \mathbf{AR}, \ldots, \mathbf{A}^{(q-1)}\mathbf{R}\}, \tag{4.14}$$

such that $\mathbf{V}^{\mathrm{t}}\,\mathbf{V} = \mathbf{I}_{m\times m}$ (orthogonal), where $m = q \times n_{\mathrm{in}}$. Using the (block) Krylov basis $\mathbf{V}$ as a projection matrix is a common approach in MOR [38, 20, 8, 50]. The Lanczos algorithm [69] and Arnoldi process [7] are two numerically robust methods to generate the Krylov basis matrix $\mathbf{V}$.

The Padé via Lanczos (PVL) method was the first projection-based method [34] to implicitly match the reduced model and the original system to a certain order of moments [51]. The matrix PVL (MPVL) algorithm is an extension of PVL to general multiple-input multiple-output systems [35, 3]. To deal with circuits with symmetric matrices, the SyPVL algorithm [42] or its multiport counterpart (SyMPVL) [43] were developed.

4.3.6 Arnoldi algorithm

The Arnoldi process using the modified Gram–Schmidt orthogonalization recursively produces a set of orthonormal vectors as the basis for a given Krylov subspace $\mathcal{K}\boldsymbol{r}(\mathbf{A}, \mathbf{R}, q)$ [7]. The algorithm generates an orthogonal projection matrix $\mathbf{V} \in \mathbb{R}^{N\times m}$ and a block upper Hessenberg matrix $\mathcal{H} \in \mathbb{R}^{m\times m}$ which satisfy

$$\mathbf{V}^{\mathrm{t}}\mathbf{A}\mathbf{V} = \mathcal{H}. \tag{4.15}$$

For the practical implementations of the Arnoldi algorithm for single-input single-output systems and the block-Arnoldi algorithm for the multiple-input multiple-output cases, one can refer to [102, 20, 1].

4.4 Formulation of RC circuits representing on-chip interconnects

As stated in Section 4.1 and shown in Figure 4.1, the presence of high-speed interconnects is ubiquitous at all levels of the electronics design hierarchy, be it on the silicon (on-chip or die), package, board, or backplanes level. It is often the case that the on-chip interconnects are characterized by increased resistance and negligible inductance. In this situation, a mesh of RC elements can accurately model the on-chip interconnect. Based on a general MNA circuit formulation (4.4), the impedance parameter realization for RC network presenting an on-chip interconnect is obtained with $\mathbf{L} = \mathbf{B}$ as

$$\mathbf{G}\mathbf{x}(t) = -\mathbf{C}\frac{d\mathbf{x}(t)}{dt} + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{z}(t) = \mathbf{B}^{\mathrm{t}}\mathbf{x}(t), \tag{4.16}$$

where $\mathbf{G}$ and $\mathbf{C}$ are symmetric and respectively contain the stamps of parasitic resistors and capacitors and $\mathbf{u}(t)$ contains the current source excitations at input terminals. The capacitors being real and positive, matrix $\mathbf{C}$ is positive semi-definite. For this realization, $\mathbf{G}$ has positive diagonal entries that are greater than or equal to the sum of the absolute value of the off-diagonal elements in its row, and are so-called irreducibly diagonally dominant [82]. This means that none of the eigenvalues of either matrix is negative.

4.4.1 Reduced RC macromodel for on-chip interconnects

The central idea to obtain an efficient and accurate macromodel for the RC interconnect in (4.16) is to exploit the symmetry and positive definiteness properties of its MNA matrices. Assuming $\mathbf{G}$ is invertible, and hence the circuit has a DC solution, it is

$$\mathbf{G} = \mathbf{G}^{\mathrm{t}} > 0 \quad \text{and} \quad \mathbf{C} = \mathbf{C}^{\mathrm{t}} \geq 0. \tag{4.17}$$

A circuit with matrices possessing the properties in (4.17) is referred to as a symmetric system. For symmetric positive definite $\mathbf{G}$ shown in (4.17), the Cholesky factorization exists as [49]

$$\mathbf{G} = \mathbf{G}_{\mathbf{L}}\mathbf{G}_{\mathbf{L}}{}^{\mathrm{t}}, \tag{4.18}$$

where $\mathbf{G}_{\mathbf{L}}$ is lower triangular with positive diagonal elements. By substituting (4.18) in (4.16) and defining $\mathbf{J} \stackrel{\Delta}{=} \mathbf{G}_{\mathbf{L}}{}^{-1}$, we get

$$\mathbf{x}(t) = -\mathbf{J}\mathbf{C}\mathbf{J}^{\mathrm{t}}\frac{d\mathbf{x}(t)}{dt} + \mathbf{J}\mathbf{B}\mathbf{u}(t), \qquad \mathbf{z}(t) = (\mathbf{J}\mathbf{B})^{\mathrm{t}}\mathbf{x}(t). \tag{4.19}$$

Using the projection matrix $\mathbf{V}$ obtained from running the block Arnoldi algorithm with matrix $\mathbf{R} = \mathbf{JB}$ and the symmetric negative semi-definite matrix $\mathbf{A} = -\mathbf{JCJ}^{\mathrm{t}}$, considering (4.15), the reduced macromodel for (4.19) is constructed as

$$\hat{\mathbf{x}}(t) = \mathbf{\Lambda}\frac{d\hat{\mathbf{x}}(t)}{dt} + \hat{\mathbf{B}}\mathbf{u}(t), \qquad \mathbf{z}(t) = \hat{\mathbf{B}}^{\mathrm{t}}\mathbf{x}(t), \tag{4.20}$$

where $\mathbf{\Lambda} = -\mathbf{V}^{\mathrm{t}}\mathbf{JCJ}^{\mathrm{t}}\mathbf{V} \in \mathbb{R}^{m\times m}$ is symmetric and block tridiagonal and $\hat{\mathbf{B}} = \mathbf{V}^{\mathrm{t}}\mathbf{JB}$. The macromodel in (4.20) is an m-th-order system matching the first $2q$ moments of the original system.

Proposition 4.1. *For any symmetric system, using a projection matrix formed by the Krylov bases as* $\mathbf{V} = \mathcal{K}r(-\mathbf{JCJ}^{\mathrm{t}}, \mathbf{JB}, m)$, *the first 2q (block) moments of the original and reduced-order system of order* $m = q \times n_{\mathrm{in}}$ *match.*

The proof is possible by induction and is straightforward by following the similar steps presented in [103].

An attempt to generalizing this approach to the cases of RL and LC circuits has been presented in [44].

4.5 Model order reduction of RLC on-chip interconnects

As a long interconnect is imposed with faster on-chip rise times, the impact of its inductive property becomes noticeable. The wide wires which are frequently encountered in clock distribution networks and upper metal layers can be considered as typical examples. These wires, having a low resistance, exhibit inductive effects with a dominant impact on signal propagation. In these cases, inductance cannot be neglected anymore and needs to be included in the models for realistic simulations of VLSI designs. To handle the resulting extremely large RLC circuit models for on-chip interconnects, several algorithms are available in the literature [110, 67, 37]. While these methods can produce an accurate reduction for RLC networks, they cannot guarantee the passivity. However, preserving passivity in the reduction of general RLC networks is a practical necessity. Passivity implies that a network cannot generate more energy than it absorbs from its sources. It is important because the cascade connections of (strictly) passive circuits will be (asymptotically) stable [19]. However, interconnections of stable but nonpassive macromodels may not necessarily be stable. For a detailed account of passivity and the importance of passivity preservation, [12, Chapter 5] can be referred to.

4.5.1 Passive reduced-order interconnect macromodeling algorithm

To establish the idea, let us consider the admittance parameter realization [20] for a general multiport RLC network in the time domain described using MNA circuit equations in the form presented in (4.4) with $\mathbf{L} = \mathbf{B}$, where $\mathbf{u}(t)$ contains the voltage source excitations at input terminals and outputs are the currents entering the same terminal. Logical partitioning of the unknown vector $\mathbf{x}$ can be given as $\mathbf{x} = [\mathbf{v}_{\text{nodes}}, \mathbf{i}_{\text{branches}}]^{\text{t}}$, where $\mathbf{v}_{\text{nodes}}$ is voltages at the nodes and $\mathbf{i}_{\text{branches}}$ contains the currents in the branches of inductors and voltage sources. Correspondingly, $\mathbf{G}$, $\mathbf{C}$, and $\mathbf{B}$ in (4.4) can be partitioned as

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{12}^{\text{t}} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}, \tag{4.21}$$

where $\mathbf{G}_{11}$ is symmetric positive definite provided each internal node has a DC path (to the ground), $\mathbf{G}_{12}$ is a block containing zeros and ones, $\mathbf{C}_{11}$ is symmetric and positive semi-definite, and $\mathbf{C}_{22}$ is symmetric negative semi-definite. The passive reduced-order interconnect macromodeling algorithm (PRIMA) [20, 88] is a Krylov subspace-based projection method using the Arnoldi process. By taking advantage of the particular block structure of linear RLC circuits as given in (4.21), PRIMA creates passive reduced-order models. The passivity preservation requires a simple modification in the circuit formulation as negating the rows in conductance matrix $\mathbf{G}$, susceptance matrix $\mathbf{C}$, and input matrix $\mathbf{B}$ corresponding to the current variables as [20, 110]

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ -\mathbf{G}_{12}^{\text{t}} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_{22} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ -\mathbf{B}_2 \end{bmatrix}. \tag{4.22}$$

Using the modified matrices in (4.22) and the projection matrix from the Arnoldi process, the reduced model in the form presented in (4.13) can be computed. The following theoretical results can be stated for the resulting reduced macromodel.

Corollary 4.1 (Preservation of moments). *Given an $m \times m$ Krylov basis projection matrix as $\mathbf{V} = \mathcal{K}r(-\mathbf{G}^{-1}\mathbf{C},\ \mathbf{G}^{-1}\mathbf{R})$, the reduced PRIMA macromodel in* (4.13) *preserves the first $q = \lfloor m/n_{\text{in}} \rfloor$[3] block moments of the original system.*

For the proof, [20, 88] can be referred to.

4.5.1.1 Passivity of the reduced model

A linear network is passive if its admittance or impedance transfer function matrix $\hat{\mathbf{H}}(s)$ is positive real by satisfying the following necessary and sufficient conditions

3 $\lfloor x \rfloor$ Floor operator rounds down x to the largest integer number less than or equal to x.

[5, 84, 18]:

$$\begin{cases} \hat{\mathbf{H}}(s) \text{ is defined and analytic in } \Re\mathfrak{e}(s) > 0\,, & \text{(a)} \\ \hat{\mathbf{H}}^*(s) = \hat{\mathbf{H}}(s^*)\,, & \text{(b)} \\ \boldsymbol{\Phi}(s) = (\hat{\mathbf{H}}(s) + \hat{\mathbf{H}}^{*t}(s)) \geq 0 \quad \forall s \in \mathbb{C}\colon \Re\mathfrak{e}(s) > 0, & \text{(c)} \end{cases} \tag{4.23}$$

where superscript $*$ is the complex conjugate operator. For the reduced models the real reduced matrices $\hat{\mathbf{C}}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{L}}$ are real, the passivity can be equivalently investigated by checking [68, 88]

$$\boldsymbol{\Phi}(s) = (\hat{\mathbf{H}}(s) + \hat{\mathbf{H}}^{t}(s^*)) \geq 0 \quad \forall s \in \mathbb{C}\colon \Re\mathfrak{e}(s) > 0. \tag{4.24}$$

For the macromodel such as impedance and admittance realizations where $\mathbf{L} = \mathbf{B}$, we have the following.

Corollary 4.2 (Preservation of passivity). *Given a continuous-time linear system with the real-valued matrices* $(\mathbf{G} + \mathbf{G}^{t}) \geq 0$ *and* $\mathbf{C}^{t} = \mathbf{C} \geq 0$ *and any full rank reduction projection matrix* $\mathbf{V} \in \mathbb{R}^{N\times m}$*, the reduced PRIMA macromodel is passive.*

The proof is possible by showing the satisfaction of the passivity condition in (4.24) following the steps presented in [20, 88].

From Corollary 4.1, it is seen, after selecting the required number of block moments q to achieve a desired predefined accuracy, that the order of the reduced system m proportionally increases with the increase in the number of ports, i. e., $m = q \times n_{\text{in}}$.

4.5.2 Example: reduction using PRIMA

In this example, we consider an RLC mesh shown in Figure 4.6. It is connected to the rest of the circuit through its 24 ports. The order of the subcircuit (excluding terminations) is $N = 5{,}800$. The original subcircuit is reduced using the PRIMA algorithm to form a passive reduced macromodel of order $m = 290$.

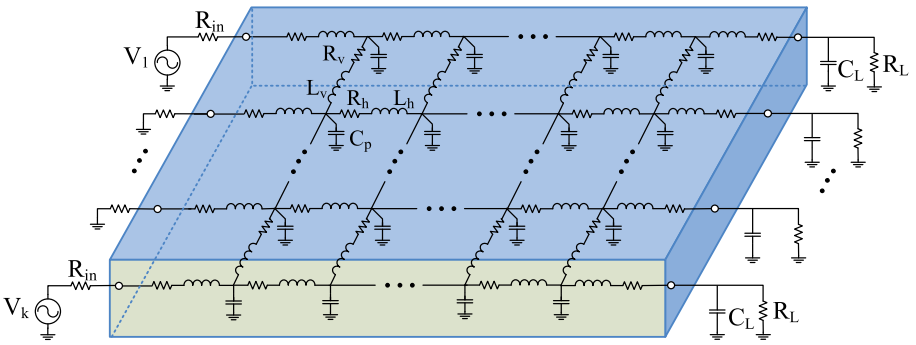


Figure 4.6: A network including a 24-port RLC mesh as its subcircuit (Section 4.5.2).

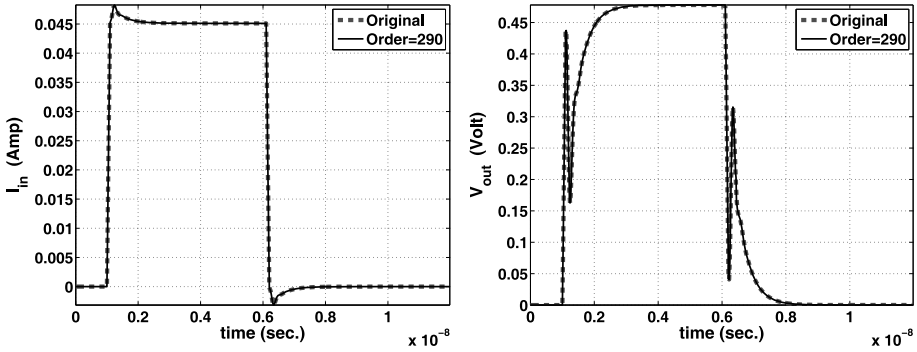


Figure 4.7: Transient responses at the terminal at the left of the horizontal trace#1 (top) and trace#10 (bottom) (Section 4.5.2).

The reduced model is plugged in the MNA equations of the rest of the circuit, which has three input voltage sources connected to the near-ends terminals 1, 6, and 12. The test excitations are set to trapezoidal pulses with rise/fall times of 0.1 ns, a delay of 1 ns, and a pulse width of 5 ns. The simulation results obtained from the original circuit of Figure 4.6 and from the network using the reduced macromodel are compared in Figure 4.7, which shows excellent agreement.

4.5.3 Structure-preserving model order reduction of RLC interconnects

The prominence of the PRIMA method presented in Section 4.5.1 is mainly due to its passivity preservation. As previously described in Section 4.5.1, to create passive reduced-order models, PRIMA relies on the special block structure of linear RLC circuits (4.22). Later, in [40, 39, 41], the structure-preserving reduced-order interconnect macromodeling (SPRIM) method was developed. Besides passivity, SPRIM can preserve other characteristics inherent to RLC circuits, such as the block structure of the circuit matrices and the reciprocity.

In the SPRIM approach, first, a projection matrix $\mathbf{V} \in \mathbb{R}^{N\times m}$ is obtained by running the Arnoldi process. According to the block structure of the system matrices, the projection matrix is partitioned to

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}. \tag{4.25}$$

Next, the blocks $\mathbf{V}_1$ and $\mathbf{V}_2$ are rearranged to form a new projection matrix as

$$\hat{\mathbf{V}} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{bmatrix} \in \mathbb{R}^{N\times 2m}. \tag{4.26}$$

Using $\hat{\mathbf{V}} \in \mathbb{R}^{N\times 2m}$ (4.26) and based on the concurrence transformation, the reduced model is obtained.

It can be shown that the SPRIM constructs the passive reduced-order models, which preserve twice as many moments as the corresponding PRIMA models obtained with (almost) the same computational cost. Hence, SPRIM offers comparable accuracy with the Padé-type approximation in the sense that both match twice as many moments. For the real frequency expansion point s_0, SPRIM is more accurate than PRIMA. The SPRIM model, written in the form of first-order DAEs, would be twice as large as the corresponding PRIMA model. However, the SPRIM model can always be represented in the second-order form with the same size as the PRIMA model.

4.6 Model order reduction of RLC interconnect structures with many ports

For designers to accurately assess on-chip layout-dependent parasitics before fabrication, extremely large RLC representations for on-chip interconnects are automatically extracted from layout and are included as subnetworks in the netlist for circuit simulation. Often these subnetworks have many interfaces with other parts of the on-chip design. Some challenges arise in the MOR of such RLC networks with a large number (e. g., thousands) of input/output terminals. The direct application of the conventional implicit moment-matching MOR techniques such as PRIMA and SPRIM on a multiport network often leads to inefficient transient simulations due to the large and dense reduced models. As shown in Section 4.5.1, to achieve the desired accuracy, for every increase in the number of ports, the order of the reduced system increases proportionally to the number of block moments.

Several attempts have been made to confront this problem via port-compression. Early studies in [33, 36] reveal that there may exist a large degree of correlation between various input and output terminals. Incorporating this correlation information in the matrix transfer function at the I/O ports of the reduced model during the reduction process became the common theme in the existing terminal-reduction methods. However, the major difficulty in port-compression algorithms such as SVDMOR [33], ESVDMOR [79], RecMOR [36], and several others [15, 78, 76, 80, 77] is that the correlation relationship is frequency-dependent and in many cases also input-dependent. In general, practical networks with many ports rarely exhibit a high degree of correlation [118]. As a consequence, such a reduction can lead to accuracy loss.

Also, due to the importance of on-chip RC interconnect, various efforts have been reported in the literature tackling this issue for the case of RC networks, such as [66, 119, 60, 89].

Recently, for the general case of on-chip interconnect RLC circuits with a large number of ports, an efficient MOR has been presented in [86]. This method exploiting the superposition paradigm [106, 11] proposes a reduction strategy based on a flexible

clustering of inputs. Thereby, the problem of reducing networks with many ports is simplified by clustering inputs into small groups, and reducing each subsystem individually. Next, the reduced subsystems are concatenated to constitute a global macromodel resulting in an accurate and sparse block-diagonal reduced model (see the right graph in Figure 4.10), which is stable by construction.

Since subsystems are treated independently, passivity is not always guaranteed. However, the flexible clustering scheme of the method, along with the information from the geometry of the design, is used to improve the passivity of the resulting model. The flexibility of the technique allows passivity preservation to be considered as the primary criterion when grouping the lines into the clusters. Utilizing well-established passivity enforcement techniques as presented in [12, Chapter 5] can also be considered as an alternative strategy. To this end, based on the a posteriori passivity check, a postprocessing procedure can be applied to enforce the model passivity.

4.6.1 Example: reduction of multiport network

The multiconductor interconnect circuit shown in Figure 4.8 has 64 terminals through which it is connected to the rest of the design.

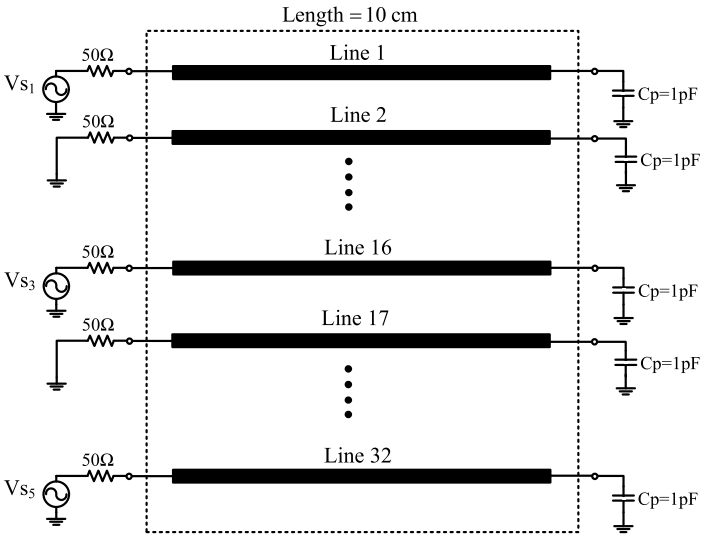


Figure 4.8: 32-conductor coupled interconnect network with terminations (Section 4.6.1).

Applying the multiport-MOR method in [86] to the network in Figure 4.8, a multiport reduced-order model is obtained. Figure 4.9 demonstrates the accuracy of the resulting reduced model by sample comparisons of time-domain responses, depicting an excellent agreement of the responses.

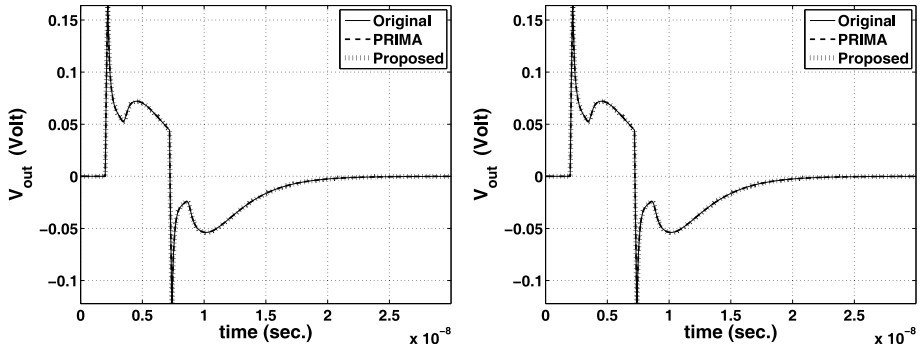


Figure 4.9: Transient responses at victim line near-end of line#2 (top) and far-end of line#31 (bottom) (Section 4.6.1).

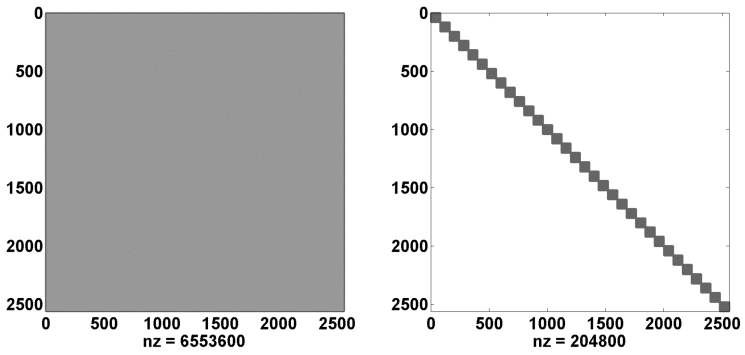


Figure 4.10: (left) Sparsity pattern of reduced MNA equations using conventional PRIMA (dense). (right) Sparsity pattern of reduced MNA equations using the method in [86] (Section 4.6.1).

Figure 4.10 illustrates the block diagonal structure of the resulting reduced model compared to using the conventional PRIMA algorithm, which admits a significant sparsity advantage.

Table 4.1 compares the CPU time expense for the transient simulation of the multiport circuit in Figure 4.9 using different approaches. As seen, while applying PRIMA leads to a macromodel that is prohibitively expensive even compared to the original

Table 4.1: CPU cost comparison between the original system, PRIMA, and the method in [86].

	Original	PRIMA	Method in [86]
Size	29.195	2.560	2.560
Total CPU time (s)	645.9	1730	111.7

circuit, the multiport reduction algorithm in [86] achieves a speedup factor of 15.5 compared to PRIMA.

4.7 Model order reduction of active circuits

Active circuits are often used (inserted) along the on-chip interconnects for example to minimize the propagation delay of the signals transmitted through those interconnect lines [61, 9, 115, 10, 2, 112, 26, 59, 4]. An illustration of the idea is presented in Figure 4.11.

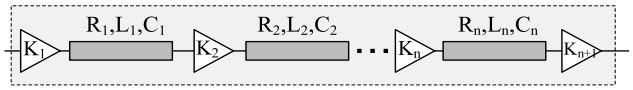


Figure 4.11: Amplifiers inserted in an RLC line to minimize the propagation delay by dividing the interconnect line into shorter sections [61].

The active circuits generally include dependent voltage and/or dependent current sources, e. g., in small-signal device models, amplifier circuits, etc. Considering the stamps of dependent sources as presented in Section 4.2.1.4, it is straightforward to see that in the presence of these sources, $(\mathbf{G} + \mathbf{G}^{t})$ is not always positive semi-definite. For such circuits, the congruence transformation does not guarantee the stability of the reduced models. Therefore, using conventional MOR techniques such as PRIMA (Section 4.5.1) and SPRIM (Section 4.5.3) for the reduction of these circuits will not guarantee the stability of the resulting model. However, it is desirable and often crucial that resulting reduced-order models inherit the stability of the original circuit. Unstable models can lead to inaccurate or totally unfeasible time-domain simulation.

There are a few approaches for stability preservation of the resulting reduced models, such as [107, 17, 75, 64], which are usually based on postprocessing. For example, in [64, 62, 90] a method is proposed to eliminate the unstable poles of the reduced system by using implicitly and explicitly restarted Arnoldi and Lanczos algorithms. However, the common concept in these methods is to sacrifice accuracy of the reduced model in order to guarantee stability, which may destroy the integrity of the moment-matching algorithm leading to an inaccurate reduced model [107]. In addition, numerical algorithms for restoring stability [17] are not guaranteed to converge, and in general, they have a relatively high computational cost associated with them. Classical truncated balanced realization (TBR) [81, 6] is another method to preserve stability. However, it is computationally expensive, which makes it not suitable for very large circuits. In addition, the existence of a solution for general circuit formulation is not guaranteed.

In [87] a projection framework is presented for constructing stable reduced macromodels for stable active linear circuits. To this end, the right-projection matrix $\mathbf{V} \in \mathbb{R}^{N\times m}$ is formed through implicitly matching the first m moments of the original circuit equations as described in Sections 4.3.5 and 4.3.6. Next, a full rank left-projection matrix $\mathbf{U} \in \mathbb{R}^{N\times m}$ is constructed through implicitly satisfying a stability condition in the form of a generalized Lyapunov inequality [6, 17].

Given the generalized eigenvalue and eigenvector matrices of the matrix pencil $(\mathbf{C}^t, -\mathbf{G}^t)$ as $\mathbf{D} \in \mathbb{C}^{m\times m}$ and $\mathbf{\Gamma} \in \mathbb{C}^{N\times m}$, respectively, which are obtained by generalized eigenvalue decomposition as

$$\mathbf{C}^t \mathbf{\Gamma} = -\mathbf{G}^t \mathbf{\Gamma} \mathbf{D}, \tag{4.27}$$

a full column rank left-projection matrix can be formed as

$$\mathbf{U} = \left[\mathbf{\Gamma}_r, \frac{1}{2}(\mathbf{\Gamma}_c + \mathbf{\Gamma}_c^*), \frac{-j}{2}(\mathbf{\Gamma}_c - \mathbf{\Gamma}_c^*) \right] \in \mathbb{R}^{N\times m} \tag{4.28}$$

where $\mathbf{\Gamma}_r$ contains the real eigenvectors corresponding to the m_r real eigenvalues in $\mathbf{D}$ and $\mathbf{\Gamma}_c$ and $\mathbf{\Gamma}_c^*$ contain the complex eigenvectors corresponding to the m_c complex and conjugate eigenvalues of $\mathbf{D}$. The computational steps to generate stability-preserving left-projection matrix are illustrated in Algorithm 4.1.

The resulting left-projection matrix $\mathbf{U}$ plays the role of guaranteeing stability of the reduced-order model by ensuring that the Lyapunov stability constraint is satisfied by the resulting reduced model. Thereby, the proposed algorithm guarantees the stability of the reduced model by construction without numerical optimization or postprocessing.

4.7.1 Example: reduction of active circuit

We consider an amplifier circuit shown in Figure 4.12 as an example of large active circuit. The interconnect structure consists of four coupled lines of length $L = 10$ cm. The high-frequency equivalent-circuit model for amplifier blocks is shown in Figure 4.13.

The transmission structure whose geometry is shown in Figure 4.12, is modeled using lumped RLGC segmentation with p. u. l. parameters obtained from HSPICE. Applying the MOR method in Section 4.7, a stable reduced-order model for the original (stable) network in Figure 4.12 is obtained. The accuracy of the resulting reduced model is demonstrated in Figure 4.14, by comparing the transient voltage responses of the reduced model with the simulation results of the original (unreduced) model. The input signal is trapezoidal-pulse with the delay time $t_d = 1$ ns, pulse width $t_{\text{pw}} = 5$ ns, and the rise and fall times $t_r = 0.25$ ns and $t_f = 0.25$ ns, respectively.

The resulting reduced model provided an accurate time-domain response, whereas the time-domain simulation of the Arnoldi-based model failed to converge because of the unstable poles.

Algorithm 4.1: The proposed method for stable model order reduction of active circuits.

Input: Original $\mathbf{G}, \mathbf{C}, \mathbf{B}, \mathbf{L}$, Reduction-order m.
Output: Stable reduced-order model: $\hat{\mathbf{G}}, \hat{\mathbf{C}}, \hat{\mathbf{B}}, \hat{\mathbf{L}}$.

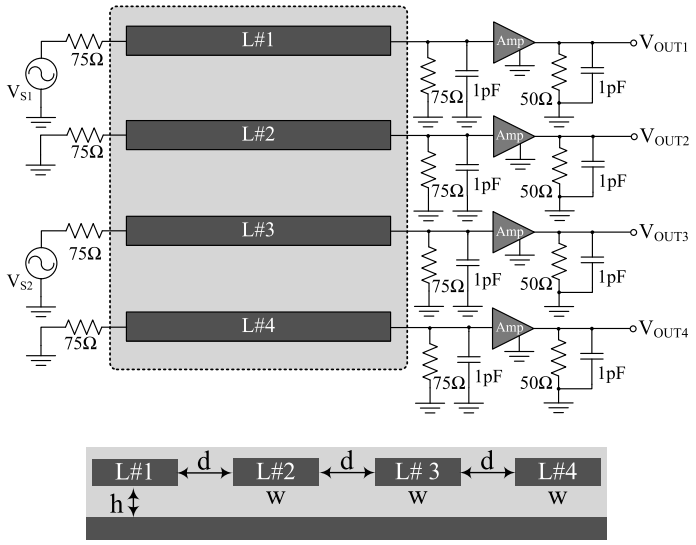
$\mathbf{V} \leftarrow \text{Arnoldi}(-\mathbf{G}^{-1}\mathbf{C}, \mathbf{G}^{-1}\mathbf{B}, m)$; // right-projection matrix
$\mathbf{\Gamma}, \mathbf{D} \leftarrow \text{eigs}(\mathbf{C}^{\text{t}}, -\mathbf{G}^{\text{t}}, m)$; // Sparse-generalized eigenproblem solver
$\mathbf{D} \leftarrow \text{diag}(\mathbf{D})$;
if $\mathbf{D}(m) \notin \mathbb{R}$ **then**
  if $\mathbf{D}(m) \neq \mathbf{D}(m-1)^*$ **then**
    $\mathbf{\Gamma}(:, m) = [\,]$;
  end
end
$i \leftarrow 1$;
while $i \leq m$ **do**
  $\boldsymbol{y}_i \leftarrow \mathbf{\Gamma}(:, i)$; // $\boldsymbol{y}_i$ is the i-th column vector in $\mathbf{\Gamma}$
  if $\mathbf{D}(i) \in \mathbb{R}$ **then**
    $\mathbf{U} \overset{\text{add to}}{\longleftarrow} \boldsymbol{y}_i$;
    $i \leftarrow i + 1$;
  else
    $\mathbf{U} \overset{\text{add to}}{\longleftarrow} \mathcal{R}e(\boldsymbol{y}_i),\ \mathcal{I}m(\boldsymbol{y}_i)$;
    $i \leftarrow i + 2$;
  end
end
$\hat{\mathbf{C}} \leftarrow \mathbf{U}^{\text{t}}\mathbf{C}\mathbf{V}, \quad \hat{\mathbf{G}} \leftarrow \mathbf{U}^{\text{t}}\mathbf{G}\mathbf{V}, \quad \hat{\mathbf{B}} \leftarrow \mathbf{U}^{\text{t}}\mathbf{B}, \quad \hat{\mathbf{L}} \leftarrow \mathbf{L}\mathbf{V}$;

Table 4.2: Comparison of the original and reduced models (Section 4.7.1).

	Dimension	Stability
Original circuit	12,024	Yes
Arnoldi-based reduced model	46	No
Proposed reduced model	46	Yes

The sizes and stability properties of the original and reduced models are compared in Table 4.2.

The CPU-time for frequency-domain simulation of the original is compared with the proposed reduced model in Table 4.3.



$$W{=}300\,\mu m,\quad d{=}100\,\mu m,\quad h{=}100\,\mu m,$$
$$\text{Conductivity: } \sigma{=}5.8\times10^{7}\,S\,m^{-1},\quad \text{Relative Permittivity: } \varepsilon_r{=}4.5$$

Figure 4.12: A stable active design consisting of four coupled interconnects and amplifier blocks (top), Cross-section of transmission line structure (bottom) – (Section 4.7.1).

Table 4.3: Comparison of the CPU time for frequency simulation using original and reduced models (Section 4.7.1).

	Original	**Reduced**
Order	12,024	46
Model generation time (s)	–	0.393
Simulation time (s)	33.306	0.195
Speedup factor		≈57

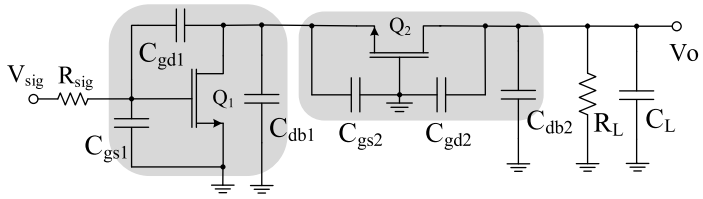


Figure 4.13: High-frequency equivalent-circuit model for MOS cascade amplifier (the biasing network is not presented) (Section 4.7.1).

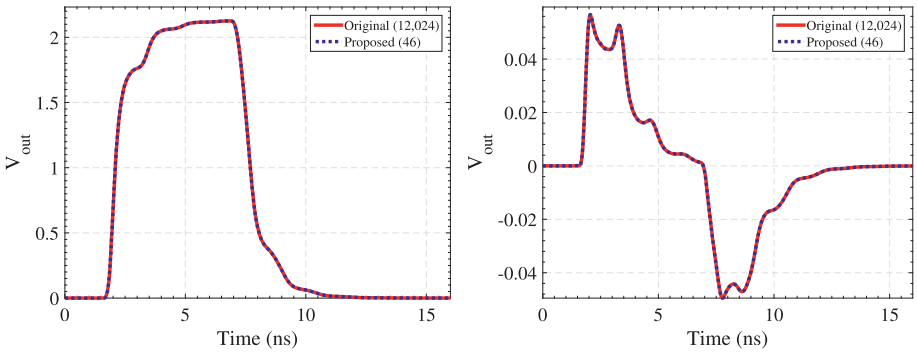


Figure 4.14: Comparison of the transient responses at V_{out3} at the far-end of line 3 (top), and at V_{out4} at the far-end of line 4 (bottom) (Section 4.7.1).

4.8 Conclusions

This chapter described the application of MOR for efficient analysis of microelectronic structures in circuit simulation environments. A general framework for formulating the circuit equations based on the MNA approach that is commonly used in commercial circuit simulators was presented. The alternative means to incorporate high-speed interconnect structures within the general formulation of the circuit equations were introduced. This leads, in general, to circuit models with a large number of lumped components. MOR techniques for RC and RLC interconnect circuits with emphasis on stability and passivity preservation were reviewed. Current challenges in the MOR of interconnect circuits with a large number of ports were presented along with some of the recent MOR techniques to handle this kind of circuits. In addition, existing techniques for the reduction of active stable circuits were reviewed with emphasis on guaranteeing the stability of the reduced circuits by construction.

It should be noted here that the presentations of MOR application in high-speed interconnects have been restricted to those techniques that are based on projecting the system into its Krylov subspace. Other projection-based methods have also been proposed to handle the high-speed interconnect. Worthy of note among those methods are the truncated balanced realization algorithms [48, 71, 93, 92, 70, 95]. Another class of projection-based methods known as proper orthogonal decomposition (POD) or principal component analysis were proposed for both linear and nonlinear systems [16, 58]. However, they have not been widely applied to microelectronics.

In addition to the above projection methods, there are also nonprojection methods, of which the explicit moment-matching of Section 4.3.1 is a known example. There are other well-known approaches based on the Hankel norm of the system [48, 104]. Another group of nonprojection approaches relies on fitting the transfer function of the system [23] in the frequency domain, of which the method based on vector fit-

ting [56, 52, 85] is widely adopted in high-speed interconnects. An alternative class of nonprojection methods is constituted using the basic idea behind a method termed "selective node elimination" [30]. Several methods related to this approach have been developed in the literature (e. g., [97, 109, 114, 120, 100]), and the time constant equilibration reduction [108] for the reduction of RC networks, which was extended to RLC circuits in [22].

Bibliography

[1] R. Achar and M. S. Nakhla, Simulation of high-speed interconnects, *IEEE Proc.*, **89** (5) (May 2001), 693–728.

[2] V. Adler and E. G. Friedman, Repeater design to reduce delay and power in resistive interconnect, *IEEE Trans. Circuits Syst. II*, **45** (5) (May 1998), 607–616.

[3] J. Aliaga, D. Boley, R. Freund, and V. Hernández, A Lanczos-type method for multiple starting vectors, *Math. Comput.*, **69** (232) (May 2000), 1577–1601.

[4] C. Alpert and A. Devgan, Wire segmenting for improved buffer insertion, in *Proc. 34th Design Automat. Conf.*, pp. 588–593, Jun 1997.

[5] B. D. O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Prentice Hall, Englewood Cliffs, NJ, 1973.

[6] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, PA, USA, 2005.

[7] W. E. Arnoldi, The principle of minimized iteration in the solution of the matrix eigenvalue problem, *Quat. Appl. Math.*, **9** (1) (Apr 1951), 17–29.

[8] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, *Appl. Numer. Math.*, **43** (1–2) (2002), 9–44.

[9] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison Wesley, Boston, MA, USA, 1990.

[10] H. B. Bakoglu and J. Meindl, Optimal interconnect circuits for VLSI, *IEEE Trans. Electron Devices*, **32** (5) (May 1985), 903–909.

[11] P. Benner, L. Feng, and E. B. Rudnyi, Using the superposition property for model reduction of linear systems with a large number of inputs, in *Proc. 18th Int. Symp. Math. Theory Netw. Syst.*, Blacksburg, Virginia, USA, pp. 1–12, Jul 2008.

[12] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.

[13] P. Benner, M. Hinze, and E. J. W. ter Maten (eds.), *Model Reduction for Circuit Simulation*, Springer-Verlag, Berlin, Heidelberg, Germany, 2011.

[14] P. Benner, V. Mehrmann, and D. C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational Science and Engineering, vol. 45, Springer-Verlag, Berlin, Heidelberg, Germany, 2005.

[15] P. Benner and A. Schneider, Model order and terminal reduction approaches via matrix decomposition and low rank approximation, in J. Roos and L. R. J. Costa (eds.) *Sci. Comput. Elect. Eng. SCEE 2008*, pp. 523–530, Springer-Verlag, Berlin, Heidelberg, Germany, 2010.

[16] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.*, **25** (Jan 1993), 539–575.

[17] B. N. Bond and L. Daniel, Guaranteed stable projection-based model reduction for indefinite and unstable linear systems, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, San Jose, CA, USA*, pp. 728–735, Nov 2008.

[18] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA, USA, 1994.

[19] S. V. Brian and D. O. Anderson, *Network Analysis and Synthesis: a Modern Systems Theory Approach*, 2nd ed., Dover, New York, NY, USA, 2006.

[20] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*, Kluwer Academic, Boston, MA, USA, 2002.

[21] E. Chiprout and M. S. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*, Kluwer Academic, Boston, MA, USA, 1994.

[22] M. H. Chowdhury, C. S. Amin, Y. I. Ismail, C. V. Kashyap, and B. L. Krauter, Realizable reduction of RLC circuits using node elimination, in *Proc. IEEE Int. Circuits Sys. Symp.*, vol. 3, Bangkok, Thailand, pp. III.494–III.497, May 2003.

[23] C. Coelho, J. Phillips, and L. Silveira, Robust rational function approximation algorithm for model generation, in *Proc. 36th ACM/IEEE Design Automat. Conf.*, New Orleans, LA, USA, pp. 207–212, Jun 1999.

[24] L. Daniel, O. C. Siong, L. Chay, K. H. Lee, and J. White, A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **23** (5) (May 2004), 678–693.

[25] C. de Villemagne and R. E. Skelton, Model reductions using a projection formulation, in *Proc. 26th IEEE Conference on Decision and Control*, Los Angeles, CA, USA, pp. 461–466, 1987.

[26] S. Dhar and M. A. Franklin, Optimum buffer circuits for driving long uniform lines, *IEEE J. Solid-State Circuits*, **26** (1) (Jan 1991), 32–40.

[27] N. Dong and J. Roychowdhury, General-purpose nonlinear model-order reduction using piecewise-polynomial representations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **27** (2) (2008), 249–264.

[28] A. Dounavis, R. Achar, and M. Nakhla, A general class of passive macromodels for lossy multiconductor transmission lines, *IEEE Trans. Microw. Theory Tech.*, **49** (10) (Oct 2001), 1686–1696.

[29] A. Dounavis, X. Li, M. S. Nakhla, and R. Achar, Passive closed-form transmission line model for general purpose circuit simulators, *IEEE Trans. Microw. Theory Tech.*, **47** (12) (Dec 1999), 2450–2459.

[30] P. J. H. Elias and N. P. van der Meijs, Extracting circuit models for large RC interconnections that are accurate up to a predefined signal frequency, in *Proc. ACM/IEEE Design Automat. Conf.*, Las Vegas, NV, USA, pp. 764–769, Jun 1996.

[31] M. Farhan, E. Gad, M. Nakhla, and R. Achar, New method for fast transient simulation of large linear circuits using high-order stable methods, *IEEE Trans. Compon. Packag. Technol.*, **3** (4) (2013), 661–669.

[32] M. Farhan, E. Gad, M. Nakhla, and R. Achar, Fast simulation of microwave circuits with nonlinear terminations using high-order stable methods, *IEEE Trans. Microw. Theory Tech.*, **61** (1) (2013), 360–371.

[33] P. Feldmann, Model order reduction techniques for linear systems with large numbers of terminals, in *Proc. Design Auto. Test Eur. Conf. Exhib.*, vol. 2, Paris, France, pp. 944–947, Feb 2004.

[34] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **14** (5) (May 1995), 639–649.

[35] P. Feldmann and R. W. Freund, Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm, in *Proc. ACM/IEEE Design Automat. Conf.*, pp. 474–479, San Francisco, CA, USA, Jun 1995.

[36] P. Feldmann and F. Liu, Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, San Jose, CA, USA*, pp. 88–92, Nov 2004.

[37] R. W. Freund, Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation, in B. N. Datta (ed.) *Applied and Computational Control, Signals, and Circuits*, vol. 1, Chapter 9, pp. 435–498, Birkhäuser Boston, Boston, MA, USA, 1999.

[38] R. W. Freund, Krylov-subspace methods for reduced-order modeling in circuit simulation, *Comput. Appl. Math.*, **123** (1–2) (Nov 2000), 395–421.

[39] R. W. Freund, Structure-preserving model order reduction of RLC circuit equations, in W. Schilders, H. van der Vorstand, and J. Rommes (eds.) *Model Order Reduction: Theory, Research Aspects and Applications*, Chapter 3, pp. 49–73, Springer, Berlin, Heidelberg, Germany, 2008.

[40] R. W. Freund, The SPRIM algorithm for structure-preserving order reduction of general RCL circuits, in P. Benner, M. Hinze, and E. J. W. ter Maten (eds.) *Model Reduction for Circuit Simulation*, Chapter 2, pp. 25–52, Springer-Verlag, Berlin, Heidelberg, Germany, 2011.

[41] R. W. Freund, The SPRIM algorithm for structure-preserving order reduction of general RCL circuits, in P. Benner, M. Hinze, and E. J. W. ter Maten (eds.) *Model Reduction for Circuit Simulation*, Chapter 2, pp. 25–52, Springer-Verlag, Berlin, Heidelberg, Germany, 2011.

[42] R. W. Freund and P. Feldmann, Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, San Jose, CA, USA*, pp. 280–287, Nov 1996.

[43] R. W. Freund and P. Feldmann, The SyMPVL algorithm and its applications to interconnect simulation, in *Proc. Int. Conf. Simul. Semicond. Process. Devices*, Cambridge, MA, USA, pp. 113–116, Sep. 1997.

[44] R. W. Freund and P. Feldmann, Reduced-order modeling of large linear passive multi-terminal circuits using matrix-padé approximation, in *Proc. Design Automat. Test Eur. Conf.*, pp. 530–537, Feb 1998.

[45] E. Gad, R. Khazaka, M. S. Nakhla, and R. Griffith, A circuit reduction technique for finding the steady-state solution of nonlinear circuits, *IEEE Trans. Microw. Theory Tech.*, **48** (12) (2000), 2389–2396.

[46] E. Gad and M. Nakhla, Efficient model reduction of linear periodically time-varying systems via compressed transient system function, *IEEE Trans. Circuits Syst. I*, **52** (6) (Jun 2005), 1188–1204.

[47] E. Gad, M. Nakhla, R. Achar, and Y. Zhou, A-stable and L-stable high-order integration methods for solving stiff differential equations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **28** (9) (Sep 2009), 1359–1372.

[48] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their $\mathcal{L}_\infty$-error bounds, *Int. J. Control*, **36** (6) (Jun 1984), 1115–1193.

[49] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Univ. Press, Baltimore, MD, USA, 1996.

[50] E. Grimme, *Krylov Projection Methods for Model Reduction*, PhD dissertation, Dept. Elect. Comput. Eng., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 1997.

[51] E. J. Grimme, D. C. Sorensen, and P. V. Dooren, Model reduction of state space systems via an implicitly restarted Lanczos method, *Numer. Algorithms*, **12** (1996), 1–31.

[52] S. Grivet-Talocia, The time-domain vector fitting algorithm for linear macromodeling, *AEÜ, Int. J. Electron. Commun.*, **58** (4) (2004), 293–295.

[53] S. Grivet-Talocia, H.-M. Huang, A. E. Ruehli, F. Canavero, and I. M. Elfadel, Transient analysis of lossy transmission lines: An efficient approach based on the method of characteristics, *IEEE Trans. Adv. Packaging*, **27** (1) (Feb 2004), 45–56.

[54] P. Gunupudi, R. Khazaka, and M. Nakhla, Analysis of transmission line circuits using multidimensional model reduction techniques, *IEEE Trans. Adv. Packaging*, **25** (2) (May 2002), 174–180.

[55] P. K. Gunupudi, M. Nakhla, and A. Ramachandra, Simulation of high-speed distributed interconnects using Krylov-space techniques, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **19** (July 2000), 799–807.

[56] B. Gustavsen and A. Semlyen, Rational approximation of frequency domain responses by vector fitting, *IEEE Trans. Power Deliv.*, **14** (3) (Jul 1999), 1052–1061.

[57] C.-W. Ho, A. Ruehli, and P. Brennan, The modified nodal approach to network analysis, *IEEE Trans. Circuits Syst.*, **22** (6) (Jun. 1975), 504–509.

[58] H. Hotelling, Simplified calculation of principal components, *Psychometrika*, **1** (1) (1936), 27–35.

[59] H. Huang, J. B. Bernstein, and M. Peckerar, Combined channel segmentation and buffer insertion for routability and performance improvement of field programmable analog arrays, in *Proc. IEEE Int. Conf. Comput. Design*, pp. 490–495, Oct 2004.

[60] R. Ionutiu, J. Rommes, and W. H. A. Schilders, Sparserc: Sparsity preserving model reduction for RC circuits with many terminals, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **30** (12) (Dec 2011), 1828–1841.

[61] Y. I. Ismail and E. G. Friedman, Effects of inductance on the propagation delay and repeater insertion in VLSI circuits, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **8** (2) (Apr 2000), 195–206.

[62] I. M. Jaimoukha and E. Kasenally, Implicitly restarted Krylov subspace methods for stable partial realizations, *SIAM J. Matrix Anal. Appl.*, **18** (3) (1997), 633–652.

[63] H. W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*, Prentice Hall, New York, NY, USA, 1993.

[64] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, and M. Barone, Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment, *Comput. Methods Appl. Mech. Eng.*, **272** (Apr 2014), 251–270.

[65] K. J. Kerns, *Accurate and Stable Reduction of RLC Networks Using Split Congruence Transformations*, PhD thesis, Dep. Electrical Eng, Univ. Washington, Location Seattle, Washington, USA, Sep 1996.

[66] K. J. Kerns and A. T. Yang, Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **16** (7) (Jul 1997), 734–744.

[67] K. J. Kerns and A. T. Yang, Preservation of passivity during RLC network reduction via split congruence transformations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **17** (7) (Jul 1998), 582–591.

[68] E. S. Kuh and R. A. Rohrer, *Theory of Linear Active Networks*, Holden-Day, San Francisco, CA, USA, 1967.

[69] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Natl. Inst. Bur. Stand.*, **45** (4) (Oct 1950), 255–282.

[70] J.-R. Li, F. Wang, and J. White, An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect, in *Proc. 36th ACM/IEEE Design Automat. Conf.*, New Orleans, LA, USA, pp. 1–6, Jun 1999.

[71] J.-R. Li and J. White, Efficient model reduction of interconnect via approximate system gramians, in *Proc. EEE/ACM Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, pp. 380–383, Nov 1999.

[72] P. Li and L. T. Pileggi, Compact reduced-order modeling of weakly nonlinear analog and RF circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **24** (2) (Feb 2005), 184–203.

[73] Y. Lin and E. Gad, Formulation of the Obreshkov-based transient circuit simulator in the presence of nonlinear memory elements, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **34** (1) (Jan 2015), 86–94.

[74] H. Liu, L. Daniel, and N. Wong, Model reduction and simulation of nonlinear circuits via tensor decomposition, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **34** (7) (2015), 1059–1069.

[75] P. Liu, Z. Qi, A. Aviles, and S. X. Tan, A general method for multi-port active network reduction and realization, in *Proc. IEEE Int. Behav. Model. Simul. Workshop*, San Jose, CA, USA, pp. 7–12, Sep 2005.

[76] P. Liu and W. Shi, Model order reduction of linear networks with massive ports via frequency-dependent port packing, in *Proc. ACM/IEEE Design Automat. Conf.*, pp. 267–272, Jul 2006.

[77] P. Liu, S. Tan, H. Li, Z. Qi, J. Kong, B. McGaughy, and L. He, An efficient method for terminal reduction of interconnect circuits considering delay variations, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, San Jose, CA, USA*, pp. 821–826, Nov 2005.

[78] P. Liu, S. Tan, B. McGaughy, L. Wu, and L. He, TermMerg: an efficient terminal-reduction method for interconnect circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **26** (8) (Aug 2007), 1382–1392.

[79] P. Liu, S. D. Tan, B. Yan, and B. McGaughy, An extended SVD-based terminal and model order reduction algorithm, in *Proc. IEEE Int. Behav. Modeling Simul. Workshop*, San Jose, CA, USA, pp. 44–49, Sep 2006.

[80] P. Liu, S. X. D. Tan, B. McGaughy, and L. Wu, Compact reduced order modeling for multiple-port interconnects, in *Proc. Int. Symp. Qual. Electron. Design*, San Jose, CA, USA, pp. 413–418, Mar 2006.

[81] C. B. Moore, Principal component analysis in linear systems: Controllability, Observability, and Model Reduction, *IEEE Trans. Autom. Control*, **AC-26** (Feb 1981), 17–32.

[82] F. N. Najm, *Circuit Simulation*, John Wiley, Hoboken, NJ, USA, 2010.

[83] N. M. Nakhla, A. Dounavis, R. Achar, and M. S. Nakhla, DEPACT: Delay-extraction-based compact transmission-line macromodeling algorithm, *IEEE Trans. Adv. Packaging*, **28** (1) (Feb 2005), 13–23.

[84] R. W. Newcomb, *Linear Multiport Synthesis*, McGraw-Hill, New York, 1966.

[85] B. Nouri, R. Achar, and M. S. Nakhla, *z*-Domain orthonormal basis functions for physical system identifications, *IEEE Trans. Adv. Packaging*, **33** (1) (Feb 2010), 293–307.

[86] B. Nouri, M. S. Nakhla, and R. Achar, Efficient reduced-order macromodels of massively coupled interconnect structures via clustering, *IEEE Trans. Compon. Packag. Manuf. Technol.*, **3** (5) (May 2013), 826–840.

[87] B. Nouri, M. S. Nakhla, and X. Deng, Stable model-order reduction of active circuits, *IEEE Trans. Compon. Packag. Manuf. Technol.*, **7** (5) (May 2017), 710–719.

[88] A. Odabasioglu, M. Celik, and L. Pileggi, PRIMA: passive reduced-order interconnect macromodeling algorithm, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **17** (8) (Aug 1998), 645–654.

[89] D. Oyaro and P. Triverio, TurboMOR-RC: An efficient model order reduction technique for RC networks with many ports, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **35** (10) (Oct 2016), 1695–1706.

[90] V. Papakos and I. Jaimoukha, A deflated implicitly restarted Lanczos algorithm for model reduction, in *Proc. 42nd IEEE Conf. Decision Control*, vol. 3, Maui, HI, USA, pp. 2902–2907, Dec 2003.

[91] C. R. Paul, *Analysis of Multiconductor Transmission Lines*, 2nd ed., Wiley, Hoboken, NJ, USA, 2008.

[92] J. Phillips, L. Daniel, and L. Silveira, Guaranteed passive balancing transformations for model order reduction, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **22** (8) (Aug 2003), 1027–1041.

[93] J. Phillips, L. Daniel, and L. M. Silveira, Guaranteed passive balancing transformations for model order reduction, in *Proc. 39th Annu. Design Automat. Conf.*, New Orleans, LA, pp. 52–57, 2002.

[94] J. R. Phillips, Projection-based approaches for model reduction of weakly nonlinear, time-varying systems, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **22** (2) (Feb 2003), 171–187.

[95] J. R. Phillips and L. Silveira, Poor man's TBR: a simple model reduction scheme, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **24** (1) (Jan 2005), 43–55.

[96] L. T. Pillage and R. A. Rohrer, Asymptotic waveform evaluation for timing analysis, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **9** (Apr. 1990), 352–366.

[97] Z. Qin and C.-K. Cheng, Realizable parasitic reduction using generalized Y-Δ transformation, in *Proc. ACM/IEEE Design Automat. Conf.*, Anaheim, CA, USA, pp. 220–225, Jun 2003.

[98] V. Raghavan, R. Rohrer, L. Pillage, J. Lee, J. Bracken, and M. Alaybeyi, AWE-inspired, in *Proc. IEEE Custom Integr. Circuits Conf.*, pp. 18.1.1–18.1.8. May 1993.

[99] M. Rewienski and J. White, A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **22** (2) (2003), 155–170.

[100] J. Rommes and W. H. A. Schilders, Efficient methods for large resistor networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **29** (1) (Jan 2010), 28–39.

[101] J. Roychowdhury, Reduced-order modeling of time-varying systems, *IEEE Trans. Circuits Syst. I*, **46** (10) (Oct. 1999), 1273–1288.

[102] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, USA, 2003.

[103] S. B. Salimbahrami, *Structure Preserving Order Reduction of Large Scale Second Order Models*, PhD thesis, Faculty of Mech. Eng., Tech. Univ. Munich, Munich, Germany, Jun 2005.

[104] A. J. Sasane, *Hankel Norm Approximation for Infinite-Dimensional Systems*, Lecture Notes in Control and Information Sciences, vol. 277, Springer, Berlin, Heidelberg, Germany, 2002.

[105] W. H. A. Schilders, H. A. van der Vorstand, and J. Rommes (eds.), *Model Order Reduction: Theory, Research Aspects and Applications*, Springer-Verlag, Berlin, Heidelberg, Germany, 2008.

[106] R. E. Scott, *Linear Circuits*, M. W. Essigmann (ed.), Addison-Wesley, New York, NY, USA, 1960.

[107] R. C. Selga, B. Lohmann, and R. Eid, Stability preservation in projection-based model order reduction of large scale systems, *Eur. J. Control*, **18** (2) (2012), 122–132.

[108] B. N. Sheehan, TICER: Realizable reduction of extracted RC circuits, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, pp. 200–203, Nov 1999.

[109] B. N. Sheehan, Realizable reduction of RC networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **26** (8) (Aug 2007), 1393–1407.

[110] L. M. Silveira, M. Kamon, I. Elfadel, and J. White, A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, pp. 288–294, Nov 1996.

[111] L. M. Silveira, M. Kamon, and J. White, Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures, *IEEE Trans. Compon. Packaging Manuf. Technol., Part B*, **19** (2) (May 1996), 283–288.

[112] P. Singh, J. Seo, D. Blaauw, and D. Sylvester, Self-timed regenerators for high-speed and low-power on-chip global interconnect, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **16** (6) (Jun 2008), 673–677.

[113] S. X. D. Tan and L. He, *Advanced Model Order Reduction Techniques in VLSI Design*, Cambridge Univ. Press, Cambridge, MA, USA, 2007.

[114] N. P. van der Meijs, Model order reduction of large RC circuits, in W. H. Schilders, H. A. van der Vorstand, and J. Rommes (eds.) *Model Order Reduction: Theory, Research Aspects and Applications*, Chapter 7, pp. 421–446, Springer, Berlin, Heidelberg, Germany, 2008.

[115] L. van Ginneken, Buffer placement in distributed RC-tree networks for minimal Elmore delay, in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, pp. 865–868, May 1990.

[116] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd ed., Kluwer Academic, Boston, MA, USA, 2003.

[117] Y. Wan and J. Roychowdhury, Operator-based model-order reduction of linear periodically time-varying systems, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, San Diego, CA, USA, pp. 391–396, Jun. 2005.

[118] B. Yan, S.-D. Tan, L. Zhou, J. Chen, and R. Shen, Decentralized and passive model order reduction of linear networks with massive ports, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **20** (5) (2012), 865–877.

[119] Z. Ye, D. Vasilyev, Z. Zhu, and J. R. Phillips, Sparse implicit projection (SIP) for reduction of general many-terminal networks, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, pp. 736–743, Nov 2008.

[120] Z. Ye, D. Vasilyev, Z. Zhu, and J. R. Phillips, Sparse implicit projection (SIP) for reduction of general many-terminal networks, in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, pp. 736–743, Nov 2008.

[121] Y. Zhou, E. Gad, M. S. Nakhla, and R. Achar, Structural characterization and efficient implementation techniques for *A*-stable high-order integration methods, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **31** (1) (Jan 2012), 101–108.

Daniel Ioan, Gabriela Ciuprina, and Wilhelmus H. A. Schilders

5 Complexity reduction of electromagnetic systems

Abstract: This chapter has two main objectives: first, to propose a computer-aided consistent and accurate description of the behavior of electromagnetic devices at various speeds or frequencies and, second, to describe procedures to generate compact electrical circuits for them, with an approximatively equivalent behavior. The extracted models should have a finite complexity as low as possible, while yielding an acceptable accuracy, as well as preserve essential characteristics, such as passivity. A successful complexity reduction can be obtained if *a priori* and *on-the-fly* reduction strategies are applied before and during the model discretization, followed by *a posteriori* complexity reduction.

Keywords: electromagnetic devices, electromagnetic systems, conceptual models, mathematical models, numerical models, a priori reduction, on-the-fly reduction, a posteriori reduction

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

5.1 Introduction

An electromagnetic (EM) device is a system in which the EM field plays an essential role. In order to extract a model of finite complexity, the first objective is to reduce the complexity of the physical and mathematical models, which are initially of infinite dimension. The reduction from infinity to a finite order can be achieved by discretization with a numerical method. However, due to the structural and geometrical complexity of devices encountered now in real-life, the discrete models thus obtained have an extremely high order of complexity, reaching even orders of magnitude above millions, requiring additional complexity reduction. That is why it is important to reduce the model complexity not only after numerical discretization, but also prior to it, during the modeling stage.

The modeling procedure consists of seven successive steps, each dedicated to building a particular form of the model: *(1) Conceptual modeling* (conceptual model = geometrical model + physical model) establishes consistent geometrical models and physical models of the device including simplifying hypotheses and justified approximations of geometrical and physical nature. *(2) Mathematical modeling* formulates the mathematical equations that describe the operation of the device, presenting the con-

Daniel Ioan, Gabriela Ciuprina, Universitatea Politehnică din Bucureşti, Bucharest, Romania
Wilhelmus H. A. Schilders, TU Eindhoven, Eindhoven, Netherlands

https://doi.org/10.1515/9783110499001-005

ceptual model in a mathematical language as a properly formulated problem. *(3) Approximate analytical modeling* (AAM) determines the approximate analytical relationship between the input and output physical quantities by solving an approximate, simplified version of the model equations. *(4) Numerical modeling* aims to build an algorithm to obtain the solutions of the mathematical model equations. *(5) Computational modeling* (simulation software) aims to create and test a computer program to implement the numerical algorithm conceived in the previous step. (6) Model order reduction (MOR) finds the simplest input/output system which approximates a posteriori, i. e., after the discretization, the behavior of the modeled device, while maintaining the original behavior with an acceptable accuracy. (7) Verification and validation is the final step of the procedure, consisting of the verification of the solution obtained by simulation and the validation of the extracted model, by comparison with experiments. Without this final check the entire procedure is completely useless.

The discretization methods used in the fourth step and their main characteristics are given in Table 5.1. Since the most used one is the finite element method (FEM), the final section of this chapter is devoted solely to it.

Table 5.1: Characteristics of the fundamental numerical methods for EM field analysis.

Method	Discretization mesh	Form of discretized equations
FEM Finite element	Unstructured, formed with triangles, quadrilaterals, tetrahedrons, hexahedrons, etc.	Weak form, variational equations
FDM, FIT Finite differences, Finite integrals	Grid – mesh with regular topology, Pair of staggered dual grids in the case of hyperbolic equations	Differential (FDM) or global equations (FIT)
BEM Boundary elements	Unstructured two-dimensional mesh, on the domain boundary	Integral equations

The complexity of the EM model in steps 1 and 2, which has distributed parameters and is described by partial differential equations (PDEs), can be reduced not only a posteriori, after discretization, but also on-the-fly, during the discretization, or even a priori. All these complexity reduction steps contribute equally to obtain, eventually, a model appropriate for the designer's needs. From this perspective, it becomes apparent that the efficient reduction of the model complexity cannot be limited to a reduction applied to the discrete model with classical MOR methods, or by matrix condensation, and that the best final results require reductions before and during discretization.

This chapter is structured according to the modeling procedure above.

5.2 Fundamental quantities and equations

In the macroscopic electromagnetism, the spatial distribution of the EM field is described at each moment in time by a first set of four *local primitive* physical quantities [43], which are the alphabet of this theory: (1) *electric field strength* $\vec{E} = \vec{f}_1(\vec{r}, t)$ (V/m); (2) *electric flux density* $\vec{D} = \vec{f}_2(\vec{r}, t)$ (C/m^2); (3) *magnetic field strength* $\vec{H} = \vec{f}_3(\vec{r}, t)$ (A/m); and (4) *magnetic flux density* $\vec{B} = \vec{f}_4(\vec{r}, t)$ (T). Mathematically, they are three-dimensional time-dependent vector fields, defined over unbounded spatial and temporal domains. However, in order to be represented on computers, their restrictions are used instead, defined on a so-called bounded "computational" domain $\vec{r} \in \mathcal{D} \subset \mathbb{R}^3$, and on a specified bounded time interval $0 \leq t < T$. A second set of two primitive physical quantities describes locally the EM state of the material objects: (5) *volume charge density* $\rho = f_5(\vec{r}, t)$ (C/m^3); and (6) *volume current density* $\vec{J} = \vec{f}_6(\vec{r}, t)$ (A/m^2).

The *derived global* quantities, defined as spatial integrals on specified lines (open $\mathcal{C}$, closed Γ), surfaces (open $\mathcal{S}_\Gamma$, closed Σ), or domains ($\mathcal{D}_\Sigma$) of the local quantities provide the global description of the EM field and objects: (1) *electric voltage* $u = \int_C \vec{E} \cdot \mathrm{d}\vec{r}$ (V); (2) *electric flux* $\psi = \int_{S_\Gamma} \vec{D} \cdot \mathrm{d}\vec{A}$ (C); (3) *magnetic voltage* $u_m = \int_C \vec{H} \cdot \mathrm{d}\vec{r}$ (A); (4) *magnetic flux* $\varphi = \int_{S_\Gamma} \vec{B} \cdot \mathrm{d}\vec{A}$ (Wb); (5) *electric charge* $q = \int_{D_\Sigma} \rho \, \mathrm{dv}$ (C); and (6) *electric current intensity* $i = \int_{S_\Gamma} \vec{J} \cdot \mathrm{d}\vec{A}$ (A).

The global quantities are time-dependent scalar functions, associated with corresponding one-, two-, or three-dimensional manifolds. The curves and surfaces considered for modeling have to be sufficiently smooth (i. e., Lipschitz manifolds) and oriented. A closed curve Γ is oriented according to the right-hand rule as related to the supported surface, and a closed surface Σ is oriented outwards. Open lines or surfaces are oriented arbitrarily. These definitions highlight that the local quantities are actually associated with differential forms in external calculus. Using a simplified language, we might say that the field strengths ($\vec{E}$, $\vec{H}$) are 1-forms, the fluxes ($\vec{D}$, $\vec{B}$, $\vec{J}$) are 2-forms, and the volume charge density ρ is a 3-form. A p-form is a quantity to be integrated on a corresponding p-manifold (curve, surface, domain for $p = 1, 2, 3$, respectively), by using a corresponding differential dx (elementary length $\mathrm{d}\vec{r}$, area $\vec{dA}$, and volume dv, respectively). If we denote the space of p-forms with W^p, then the primitive EM quantities are the p-forms: $\vec{E}, \vec{H} \in W^1$, $\vec{D}, \vec{B}, \vec{J} \in W^2$, $\rho \in W^3$.

The *general laws* of the EM field describe quantitatively the EM fundamental phenomena [33]. Their global forms, valid for any three-dimensional domain $\mathcal{D}_\Sigma$ and any surface $\mathcal{S}_\Gamma$, with $\Sigma = \partial\mathcal{D}_\Sigma$ and $\Gamma = \partial S_\Gamma$, are relationships between global primitive quantities, stated as follows:

1. *electric flux law* (Gauss)

$$\psi_\Sigma = q_{\mathcal{D}_\Sigma}, \tag{5.1}$$

2. *magnetic flux law* (Gauss)

$$\varphi_\Sigma = 0, \tag{5.2}$$

3. *EM induction law* (Faraday)

$$u_\Gamma = -\frac{\mathrm{d}\varphi_{S_\Gamma}}{\mathrm{d}t}, \tag{5.3}$$

4. *magnetic circuit law* (Ampere–Maxwell)

$$u_{m\Gamma} = i_{S_\Gamma} + \frac{\mathrm{d}\psi_{S_\Gamma}}{\mathrm{d}t}. \tag{5.4}$$

The local forms of (5.1)–(5.4) are called *Maxwell's equations*:

$$\operatorname{div}\vec{D} = \rho, \qquad \nabla\cdot\vec{D} = \rho, \tag{5.5}$$

$$\operatorname{div}\vec{B} = 0, \qquad \nabla\cdot\vec{B} = 0, \tag{5.6}$$

$$\operatorname{curl}\vec{E} = -\frac{\partial\vec{B}}{\partial t}, \qquad \nabla\times\vec{E} = -\frac{\partial\vec{B}}{\partial t}, \tag{5.7}$$

$$\operatorname{curl}\vec{H} = \vec{J} + \frac{\partial\vec{D}}{\partial t}, \qquad \nabla\times\vec{H} = \vec{J} + \frac{\partial\vec{D}}{\partial t}. \tag{5.8}$$

These are general, fundamental, first-order PDEs of the macroscopic electromagnetism, valid only for media at rest, whereas (5.1)–(5.4) are valid also for moving media. Their proofs use the Gauss–Ostrogradsky and Stokes theorems and they can be written using the *del* operator ∇.

On motionless surfaces of discontinuity, the local forms are

$$\operatorname{div}_s\vec{D} = \rho_s, \qquad \vec{n}_{12}\cdot(\vec{D}_2 - \vec{D}_1) = \rho_s, \tag{5.9}$$

$$\operatorname{div}_s\vec{B} = 0, \qquad \vec{n}_{12}\cdot(\vec{B}_2 - \vec{B}_1) = 0, \tag{5.10}$$

$$\operatorname{curl}_s\vec{E} = 0, \qquad \vec{n}_{12}\times(\vec{E}_2 - \vec{E}_1) = \vec{0}, \tag{5.11}$$

$$\operatorname{curl}_s\vec{H} = \vec{J}_s, \qquad \vec{n}_{12}\times(\vec{H}_2 - \vec{H}_1) = \vec{J}_s, \tag{5.12}$$

where $\vec{n}_{12}$ is the unit vector normal to the discontinuity surface, the subscripted vectors indicate their values at opposite sides of the interface, at the same point on it, $\rho_s = f_7(\vec{r},t) = \mathrm{d}q/\mathrm{dA}$ ($\mathrm{C/m^2}$) is the *surface charge density*, and $\vec{J}_s = \vec{f}_8(\vec{r},t) = \vec{t}\,\mathrm{d}i/\mathrm{d}\vec{l}$ is the *surface current density*.

To ensure the completeness of the system of equations (5.1)–(5.4), which describe the EM field, three material-dependent *constitutive relationships* of the macroscopic electromagnetism are added [33]:

1. *Electric conduction law* (Ohm):

$$\vec{J} = f_a(\vec{E}). \tag{5.13}$$

In conductors with affine characteristics $\vec{J} = \sigma(\vec{E} + \vec{E}_i)$ or $\vec{J} = \sigma\vec{E} + \vec{J}_i$.

2. Polarization law:

$$\vec{D} = f_b(\vec{E}). \tag{5.14}$$

In dielectrics with affine characteristics $\vec{D} = \varepsilon\vec{E} + \vec{P}_p$.

3. Magnetization law:

$$\vec{B} = f_c(\vec{H}). \tag{5.15}$$

In magnetic materials with affine characteristics $\vec{B} = \mu\vec{H} + \mu_0\vec{M}_p$.

For the particular case of linear media ($\vec{E}_i = \vec{0}$, $\vec{P}_p = \vec{0}$, $\vec{M}_p = \vec{0}$), these relationships are linear functions described by the material coefficients: conductivity σ (S/m), permittivity ε (F/m), and permeability μ (H/m), which are scalar quantities in isotropic media (function of position in nonhomogeneous ones and constant otherwise). In anisotropic media, they are second-order tensors, described by symmetrical and positive definite 3×3 matrices. In affine media, the characteristics include, along with the linear term, a vector constant: the intrinsic electric field $\vec{E}_i$ or intrinsic current density $\vec{J}_i = \sigma\vec{E}_i$, the permanent polarization $\vec{P}_p$, and the permanent magnetization $\vec{M}_p$, respectively. The affine model is obtained through linearization of the nonlinear model, by using the truncated Taylor series expansion of the characteristic function around a given operating point (usually the origin), by retaining the first two terms only. Polarization $\vec{P}$ (C/m^2) and magnetization $\vec{M}$ (A/m) are physical quantities describing the deviation of the electric displacement and of the magnetic flux density in substance from their values in vacuum condition: $\vec{P} = \vec{D} - \varepsilon_0\vec{E}$, $\vec{M} = \vec{B}/\mu_0 - \vec{H}$. These quantities have a temporal component, dependent (linear or nonlinear) on the field strength, and a permanent component (constant, independent of the field strength): $\vec{P}(\vec{E}) = \vec{P}_t(\vec{E}) + \vec{P}_p$, $\vec{P}_t(\vec{0}) = \vec{0}$, $\vec{P}(\vec{0}) = \vec{P}_p$ $\vec{M}(\vec{H}) = \vec{M}_t(\vec{H}) + \vec{M}_p$, $\vec{M}_t(\vec{0}) = \vec{0}$, $\vec{M}(\vec{0}) = \vec{M}_p$.

In addition to the four general laws and the three constitutive laws, the macroscopic electromagnetism includes two *transfer laws* which describe how the energy or the substance is transferred between an EM and other physical systems:

1. The *power transfer law* (Joule) describes the energy transfer occurring between the EM field and conducting substances during the conduction process:

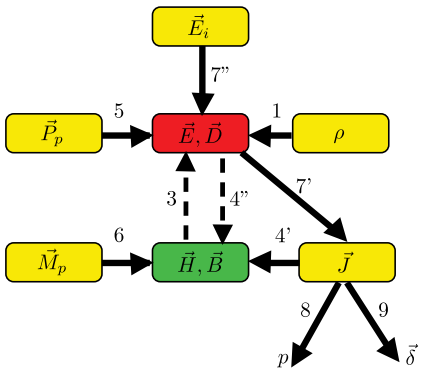
$$p = \vec{J} \cdot \vec{E} \quad (\mathrm{W/m^3}), \tag{5.16}$$

where p is the power volume density transferred from field to substance.

2. The *mass transfer law* (Faraday) describes the mass transfer in electrolytes

$$\vec{\delta} = k\vec{J} \quad (\mathrm{kg/(m^2 s)}), \tag{5.17}$$

where $\vec{\delta}$ is the mass density of the flux rate.



1. $\nabla \cdot \vec{D} = \rho$
2. $\nabla \cdot \vec{B} = 0$
3. $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$
4. $\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$
5. $\vec{D} = \varepsilon \vec{E} + \vec{P}_p$
6. $\vec{B} = \mu \vec{H} + \mu_0 \vec{M}_p$
7. $\vec{J} = \sigma(\vec{E} + \vec{E}_i)$
8. $p = \vec{J} \cdot \vec{E}$
9. $\vec{\delta} = k\vec{J}$

Figure 5.1: Diagram of electromagnetism relationships. Notations x' and x'' refer to distinct terms of the corresponding equation.

The diagram shown in Figure 5.1 illustrates the causal relationships, as described by the nine laws of electromagnetism. This image highlights the fundamental EM phenomena and their intertwining, thus providing a physical meaning to the laws. The dotted arrows represent phenomena occurring only in time-varying regimes, whereas the thick, solid arrows represent phenomena occurring in both dynamic and stationary regimes.

A different graphic representation of fundamental relations of electromagnetism, this time from a mathematical perspective as opposed to the physical/causal viewpoint, is the Tonti diagram, also known as “Maxwell’s House” [9], as shown in Figure 5.2 (left). The four pillars in this diagram are perfect De Rham sequences (Figure 5.2

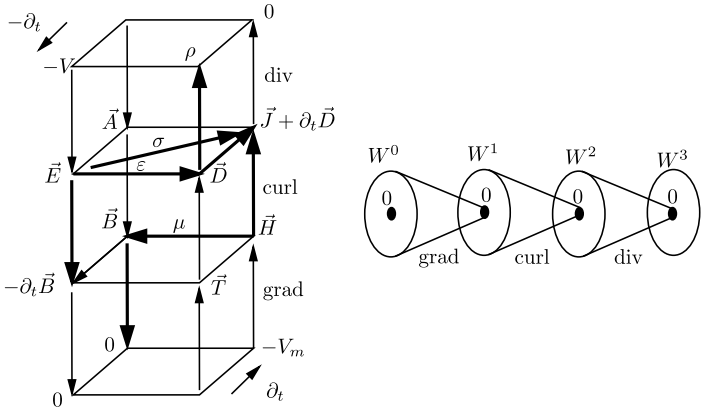


Figure 5.2: Left: Maxwell’s house. Right: De Rham sequence = chain containing spaces of differential forms, with increasing order. The sequence is perfect: the kernel of an operator is in the domain of the previous operator.

(right)), each one being associated with an EM field primitive quantity ($\vec{E}, \vec{D}, \vec{B}, \vec{H}$). The arrows representing Maxwell's relations are emphasized.

On the basis of these nine laws, two important theorems can be proved [33].

1. The *electric charge conservation theorem* highlights the strong link existent between charge and current:

$$i_\Sigma = -\frac{\mathrm{d}q_{\mathcal{D}_\Sigma}}{\mathrm{d}t} \quad \forall\, \mathcal{D}_\Sigma. \tag{5.18}$$

2. The *EM energy conservation theorem* has the following form in motionless media:

$$P_\Sigma = P_{c\mathcal{D}_\Sigma} + \frac{\mathrm{d}W_{\mathcal{D}_\Sigma}}{\mathrm{d}t} \quad \forall\, \mathcal{D}_\Sigma, \tag{5.19}$$

where $P_\Sigma = -\oint_\Sigma (\vec{E}\times\vec{H})\cdot\vec{n}\,\mathrm{dA}$ is the power transferred inward through the boundary $\Sigma = \partial\mathcal{D}_\Sigma$ of the considered domain, $P_{c\mathcal{D}_\Sigma} = \int_{\mathcal{D}_\Sigma} p\,\mathrm{dv}$ is the power transferred to the conducting substance inside the domain, and, in the case of linear media, $W_{\mathcal{D}_\Sigma} = \int_{\mathcal{D}_\Sigma} (\vec{E}\cdot\vec{D} + \vec{H}\cdot\vec{B})/2\,\mathrm{dv}$ is the EM energy inside the domain.

The *fundamental problem of EM field analysis* consists of the computation of the EM field, as a solution to Maxwell's equations, and of the constitutive relations for materials. In the general transient case, this problem can be formulated as follows. **Given**

- the shape and dimensions of the *computing domain* $\mathcal{D}_\Sigma$;
- the values of σ, ε, μ in each point of $\mathcal{D}_\Sigma$ or these *material constants* in each assumed linear or affine and homogenous subdomain;
- the *internal field sources* at each point of the computing domain, provided by $\vec{E}_i$, $\vec{P}_p$, and $\vec{M}_p$ vectors;
- *boundary conditions*: at each moment in time $0 \le t < T$, at each point on the domain's boundary $\Sigma = \partial\mathcal{D}_\Sigma$ the tangential component, either of the electric field $\vec{E}_t$ or of the magnetic field $\vec{H}_t$;
- *initial conditions*: at each point in $\mathcal{D}_\Sigma$ the electric flux density $\vec{D}$ and the magnetic flux density $\vec{B}$ at the initial moment $t = 0$, with $\mathrm{div}\,\vec{B}(\vec{r}, 0) = 0$.

Find the solution represented by the fields $\vec{E}, \vec{D}, \vec{B}, \vec{H}, \rho, \vec{J}$, defined on the computing domain $\mathcal{D}$ and on the time interval $0 \le t < T$.

Such a PDE problem is mathematically correctly formulated ("well-posed" in the Hadamard sense) if a solution exists, is unique, and is continuously dependent on the problem data (more precise, if the problem is well-conditioned). The mathematical formulation of the field problem requires a selection of the most adequate functional framework and the reformulation of the EM field problem within that framework.

The *theorem of the solution uniqueness* states that if $\sigma > 0$, $\varepsilon > 0$, $\mu > 0$, the solution of the Maxwell equations (5.1)–(5.4) together with the constitutive relations in affine media (5.13)–(5.15) is unique if the following conditions are given:

– (CD): internal sources of the field in domain $\mathcal{D}_\Sigma$:

$$\vec{E}_i(\vec{r},t),\vec{P}_p(\vec{r},t),\vec{M}_p(\vec{r},t), \quad \vec{r}\in\mathcal{D}_\Sigma,\ 0\le t<T, \tag{5.20}$$

– (CΣ): boundary conditions on Σ:

$$\vec{n}\times\vec{E}(\vec{r},t)\times\vec{n}=\vec{E}_t, \quad \vec{r}\in S_E\subset\partial\mathcal{D}_\Sigma, \tag{5.21}$$

$$\vec{n}\times\vec{H}(\vec{r},t)\times\vec{n}=\vec{H}_t, \quad \vec{r}\in S_H=\partial\mathcal{D}_\Sigma-S_E, \quad 0\le t<T, \tag{5.22}$$

– (C0): initial conditions:

$$\vec{D}(\vec{r},0)=\vec{D}_0(\vec{r}),\vec{B}(\vec{r},0)=\vec{B}_0(\vec{r}), \quad \vec{r}\in\mathcal{D}_\Sigma. \tag{5.23}$$

The proof is based on the EM energy theorem and on the null-solution lemma, which states that a linear equation has a unique solution if the associated homogenous equation (i. e., a field problem with null uniqueness conditions) has only the null solution.

Most often, the boundary conditions are null. The condition $\vec{E}_t=\vec{0}$, called perfect electric conductor, relates to a perfect conductor (superconductor with $1/\sigma=0$) located on the boundary.

If, to facilitate the understanding, we assume that $\vec{E}_t$ is given over the entire boundary, then the result of the field analysis is a unique distribution of the EM field in the computing domain. This result is uniquely determined including the component $\vec{H}_t$ on the boundary. Thus, a correct formulation of the problem of EM field analysis guarantees a proper definition of an operator which links the two input and output vector fields $\vec{E}_t$ and $\vec{H}_t$, respectively, defined on the boundary $\Sigma=\partial\mathcal{D}_\Sigma$ and on the time interval $0\le t<T$. This transfer operator defines a *dynamic input-output (I/O) system*, which describes the response of the entire domain $\mathcal{D}_\Sigma$ under several excitations.

Although from the perspective of system theory the transfer function refers only to the boundary quantities, treating the studied object as a black box, in fact the transfer function depends on the internal structure and material behavior. Thus, to extract this transfer operator the fundamental EM field problem has to be solved in the entire $\mathcal{D}_\Sigma$. If the medium of this domain is linear, then this I/O system is also linear and the I/O transfer operator is a linear one, since the solved equations are linear. From the perspective of system theory, this I/O dynamical system has the Maxwell equations as state equations. Its state variables (those describing the initial conditions) are the electric and magnetic flux densities $\vec{D}$, $\vec{B}$ in the entire domain $\mathcal{D}_\Sigma$. The I/O system thus defined, which is an EM system, has input and output signals, as well as state variables, all of infinite dimensions, as they are functions defined on Σ and $\mathcal{D}_\Sigma$, respectively.

These mathematical objects have only theoretical value, since in practice, in the computer representations their finite approximations are used. For example, in the case of uniform cubic grid with n nodes along every direction, the system has 2 components × 6 faces × n^2 nodes = $12n^2$ I/O signals and 3 components × 2 vectors × n^3 nodes = $6n^3$ state variables. These numbers of order $O(n^2)$, $O(n^3)$ tend towards infinity when

$n \to \infty$. It can be said that this EM field system is of "infinite-inputs, infinite-outputs" (IIIO) type. It would be multiple-inputs multiple-outputs (MIMO) if the number of input and output signals would be greater than one, but finite. Such a MIMO EM system is discussed in the next section.

In the classical system theory, when they have a finite number of state variables, these kinds of systems are finite, with lumped parameters. In the EM field systems, since they have an infinite number of state variables, distributed in space, we say that these systems are with distributed parameters.

5.3 Coupled field-circuit problems and EMCE boundary conditions

In this section we consider the reduction of an EM system from an infinite number of I/O signals to one with a finite number of I/O signals. This approach is necessary when EM devices with distributed parameters (which include field effects such as eddy currents) are coupled with external circuits.

By definition, electric circuits have a finite number of components interconnected at nodes, characterized by their electric potentials. A subcircuit with m terminals is characterized at each terminal node by a pair of scalar quantities: current and potential. Depending on the excitation mode, one of these quantities is the input signal, the other one being the output. Consequently, an m-polar circuit is an I/O dynamic system of MIMO type, with $m-1$ inputs and $m-1$ outputs, the m-th node being the reference potential.

In order to couple circuits and EM devices, the latter need boundary conditions compatible with external circuits. Specifically, on the boundary of the EM device model, a finite number m of equipotential patches represent the *terminals* connected to the nodes of the exterior circuit. Usually, these terminals are very good conductors (copper, silver, or even gold-plated), which may be modeled as perfect conductive parts without a significant variation of electric potential over their surfaces. Moreover, the boundary conditions need to ensure that for each terminal the associated potential and current can be correctly defined.

The *multipolar electric circuit element (ECE)* is a domain $\mathcal{D}$ (Figure 5.3) with boundary conditions that ensure the compatibility with the ECEs [42, 31].

The boundary $\Sigma = \partial\mathcal{D}$ of the domain comprises m disjoint parts $S_1, S_2, \dots S_m$, with $S = \bigcup_{k=1}^{m} S_k$, called *electric terminals* on which:

$$\vec{n} \cdot \operatorname{curl} \vec{E}(P,t) = 0 \quad (\forall)\, P \in \Sigma, \tag{5.24}$$

$$\vec{n} \cdot \operatorname{curl} \vec{H}(P,t) = 0 \quad (\forall)\, P \in \Sigma - S, \tag{5.25}$$

$$\vec{n} \times \vec{E}(P,t) = \mathbf{0}, \quad (\forall)\, P \in S, \tag{5.26}$$

where $\vec{n}$ is the normal unitary vector in P.

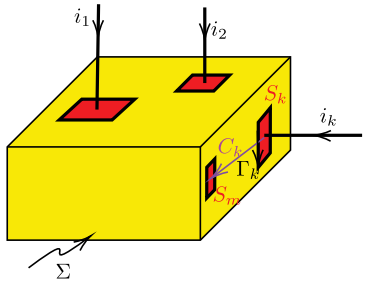


Figure 5.3: The multipolar electric circuit element (ECE).

These boundary conditions ensure: the absence of magnetic coupling with the exterior (5.24); electric coupling only through terminals (5.25); and equipotentiality of each terminal (5.26). The *electric current of a terminal k* is defined as the total current (conduction and displacement) flowing through it: $i_k(t) = \oint_{\Gamma_k} \vec{H} \cdot \mathrm{d}\vec{r}$, where $\Gamma_k = \partial S_k$ is the boundary of the terminal surface S_k. We assume that Γ_k are oriented so that the associated normal of S_k is inwards oriented. Due to (5.25), the sum of all terminal currents is zero and the Kirchhoff current law is a consequence. The *electric voltage of the terminal k* is defined as the integral $v_k(t) = \int_{C_k} \vec{E} \cdot \mathrm{d}\vec{r}$ along an arbitrary curve C_k included in Σ, which is a path between a point on S_k and a point on a reference terminal, say, S_n. Condition (5.24) ensures the consistent definition of the terminal voltage, its independence of the shape of C_k, and the Kirchhoff voltage law as a consequence.

A uniqueness theorem can be stated and the power transferred by any ECE through its boundary, from outside to inside, is given by

$$P = -\oint_{\Sigma} (\vec{E} \times \vec{H}) \cdot \mathrm{d}\vec{A} = \oint_{\Sigma} (-\mathrm{grad}\, v \times \vec{H}) \cdot (-\mathrm{d}\vec{A}) = \oint_{\Sigma} (v\, \mathrm{curl}\vec{H}) \cdot (-\mathrm{d}\vec{A})$$
$$= v_k \sum_{k=1}^{m} \int_{S_k} \mathrm{curl}\, \vec{H} \cdot (-\mathrm{d}\vec{A}) = v_k \sum_{k=1}^{m} \int_{\Gamma_k} \vec{H} \cdot \mathrm{d}\vec{r} = \sum_{k=1}^{m-1} v_k i_k. \tag{5.27}$$

If the terminals are excited by known potentials, then the problem of EM field analysis in a linear domain with ECE boundary conditions has a unique solution as it can be probed by using the lemma of the null solution,

Consequently, the terminals' currents are output signals univocally defined by solving the field problem. As the domain is linear, the equations are linear, and the device with ECE boundary conditions is a linear, MIMO dynamic system with $m-1$ input signals and $m-1$ outputs. The field problem is also well formulated, if only $p < m$ terminals have known voltages, one is grounded, and the remaining $m - p - 1$ have known currents. This is the case of the hybrid excitation. The ECE is *voltage-excited* if $p = m - 1$ and *current-excited* if $p = 0$. Under null initial conditions, due to the linearity of the field equations, and by applying the Laplace transform, the output vector $\mathbf{v}(s) = [v_1(s), v_2(s), \ldots, v_{m-1}(s)]^T$ of terminal potentials of a current-excited ECE will be

linearly dependent on the input current vector $\mathbf{i}(s) = [i_1(s), i_2(s), \ldots, i_{m-1}(s)]^T$:

$$\mathbf{v}(s) = \mathbf{Z}(s)\mathbf{i}(s), \tag{5.28}$$

where $\mathbf{Z}(s)$ is the operational impedances matrix. Since the field problem has a state space of infinite dimension, it is expected that the operational impedances have an infinite number of poles, although it is a finite $(m-1)\times(m-1)$ matrix. In addition, it is expected that the complex impedances matrix, resulting from replacement of the complex frequency s with $j\omega$, will be positive real, since the complex power $\underline{S}$ has the real part P positive for any excitation since

$$\underline{S} = P + jQ = \sum_{k=1}^{m} \underline{v}_k \underline{i}_k^* = \underline{\mathbf{i}}^H \underline{\mathbf{v}} = \underline{\mathbf{i}}^H \underline{\mathbf{Z}}\, \underline{\mathbf{i}}, \tag{5.29}$$

where $\underline{S} = -\oint_\Sigma (\underline{\vec{E}} \times \underline{\vec{H}}^*) \cdot \vec{dA}$, $P = \int_{\mathcal{D}_\Sigma} \sigma \underline{\vec{E}} \cdot \underline{\vec{E}}^*\, \mathrm{d}v$, $Q = \omega \int_{\mathcal{D}_\Sigma} (\underline{\vec{B}} \cdot \underline{\vec{H}}^* - \underline{\vec{E}} \cdot \underline{\vec{D}}^*)\, \mathrm{d}v$. Since the material constants are positive scalars, the ECE device is reciprocal (with a symmetrical impedance matrix) and passive (with absorbed active power $P > 0$, regardless of the field distribution). When terminals are voltage-excited, the complex admittance $\underline{\mathbf{Y}} = \underline{\mathbf{Z}}^{-1}$ matrix is defined similarly. In general, some terminals can be current-controlled, and the others voltage-controlled, in which case the device is characterized by a hybrid operational matrix $\mathbf{H}$.

The generalization of the ECE concept is the multipolar EM circuit element (EMCE), which has not only electric terminals, but also magnetic terminals. The power transferred by an EMCE with n electric terminals and m magnetic terminals is

$$P = -\oint_\Sigma (\vec{E} \times \vec{H}) \cdot \vec{dA} = \sum_{k=1}^{n-1} v_k i_k + \sum_{k=1}^{m-1} u_{m_k} \frac{\mathrm{d}\varphi_k}{\mathrm{d}t}, \tag{5.30}$$

where u_{m_k} is the magnetic voltage of the magnetic terminal k and φ_k is the magnetic flux of flowing through the magnetic terminal k. This expression is completely compatible with the power transferred by a multipolar electric circuit connected to the electric terminals and a multipolar magnetic circuit connected to the magnetic terminals of the EMCE.

We used EMCE formulation in applications such as modeling RF passive components or blocks as in [12], RF models of microelectromechanical switches [14, 39], or even modeling of myelinated axonal compartments [27]. Other researchers also used this formulation in magnetoquasi-static (MQS) problems for inductance extraction [38]. Similar conditions, although with a different definition for the terminal voltages, are proposed in [23].

In what follows we will assume a simple connected domain. The case of multiply connected domains have been addressed in the early paper of Timotin in [52]. The very recent comprehensive study of Hiptmair and Ostrowski [24] proves the usefulness of these boundary conditions, currently not available in popular FEM software.

5.4 Simplified models for spatial distributions and transmission lines

The next step is to reduce the dimension of the state space of the EM system. In the general case, the complete description of the EM field state requires six scalar quantities (components of $\vec{D}$ and $\vec{B}$) at every point of the physical three-dimensional space, at any moment. A dramatic reduction, to four or even two scalar quantities per point, can be achieved if a space simplification to two dimensions or even one dimension is possible.

Transmission lines (TLs) are plan-parallel structures which guide waves, encountered for instance in microelectronics, as shown in Chapter 4 of this volume [5]. At low frequencies the transmission is mainly due to conduction, whereas at high frequencies it is much more due to EM induction and displacement current. The aim of this section is to define the I/O dynamic system with distributed parameters associated to multiconductor TLs.

In the simple case of a two-conductor line, the electric field has a two-dimensional transverse distribution similar to that of an electrostatic (ES) field, being determined by the conductors' potentials, with field lines normal to the surfaces of the conductors. At high frequencies, when the penetration depth of the EM field can be neglected, the current is distributed on the conductror's surfaces and the magnetic field is also transverse and perpendicular to the electric field (Figure 5.4). Under these conditions the current $i(z,t)$ carried by the two conductors in opposite directions and the voltage $v(z,t)$ between the two conductors completely describe the EM field distribution for any cross-section of the TLs. That is why these two scalar time-dependent functions are selected as state variables.

The TL equations, derived by Heaviside in 1880, can be obtained either starting from Maxwell's equations, splitting the phenomena according to transverse and longitudinal operators as below, or by using an approach typical to circuits with distributed

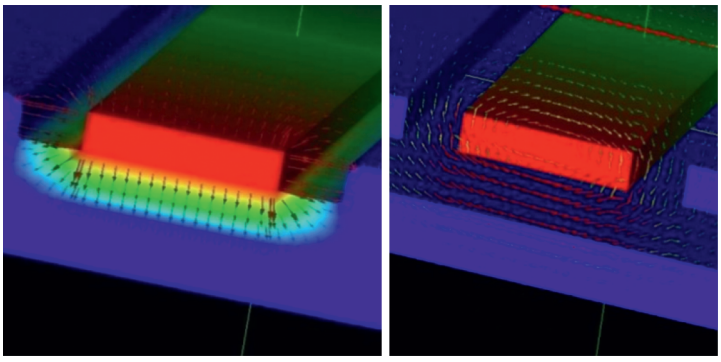


Figure 5.4: Electric (left) and magnetic (right) fields for a two-conductor transmission line.

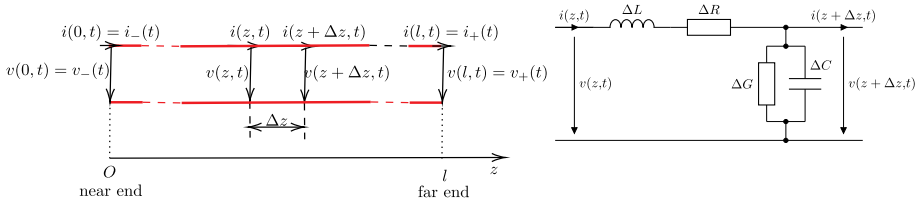


Figure 5.5: TL as an EM device (left) and its circuit model with distributed parameters (right).

parameters (Figure 5.5). The line is characterized by the per unit length (p. u. l.) parameters: R (Ω/m), L (H/m), C (F/m), and G (S/m), which may be extracted by solving four field problems in the electroconductive stationary (EC), magnetostationary (MG), ES, and EC regimes, respectively. In the simplest case of a homogeneous dielectric, only one two-dimensional ES field problem has to be solved, since $G = C\sigma/\varepsilon$ and $L = C\mu/\varepsilon$.

TLs are multipolar elements of electric circuit, their modeling being essential in many RF applications.

In the case of a multiconductor line with n conductors, $2n$ scalar time-dependent signals are required at each space point along the line. For each conductor the state variables are the local current and potential. By solving the PDE equations of the EM field, the equations which describe the propagation along an n-conductor TL are obtained [16]:

$$
\begin{aligned}
-\frac{\partial v_k}{\partial z} &= r_k^0 i_k + \sum_{m=1}^{n}\left(l_{km}^0 \frac{\partial i_m}{\partial t} + \frac{\partial}{\partial t}\int_0^t \left(\frac{\mathrm{d}l_{km}}{\mathrm{d}t}\right)_{t-\tau} i_m(\tau,t)\,\mathrm{d}\tau \right), \\
-\frac{\partial i_k}{\partial z} &= \sum_{m=1}^{n}\left(g_{km} v_m + c_{km}\frac{\partial v_m}{\partial t} \right),
\end{aligned}
\tag{5.31}
$$

where r_k^0 is the p. u. l. DC resistance of the conductor k, l_{km}^0 are p. u. l. external inductances (self inductances for $k = m$ and mutual inductances for $k \neq m$) of the conductors k and m when the return current is distributed on the surface of the substrate, and l_{km} are "transient p. u. l. inductances," defined as averaged values on the cross-section of the conductor k of the vector potential obtained in zero initial conditions by a unity step current injected in conductor m.

For current and voltage null initial conditions, the Laplace transform of the general propagation equations (5.31) are

$$
\begin{aligned}
-\frac{\mathrm{d}\mathbf{v}(z,s)}{\mathrm{d}z} &= \mathbf{Z}(s)\mathbf{i}(z,s), \\
-\frac{\mathrm{d}\mathbf{i}(z,s)}{\mathrm{d}z} &= \mathbf{Y}(s)\mathbf{u}(z,s),
\end{aligned}
\tag{5.32}
$$

with $0 < z < l$, where l is the line length and $\mathbf{v}$ and $\mathbf{i}$ are n-dimensional vectors of voltages and currents, respectively. They are similar to the TL equations (also named

telegrapher's equations) in the frequency domain, obtained from Kirchhoff equations for RLCG lines with distributed parameters. The only major difference is that here resistances and inductances are dependent on the complex frequency s in order to model the skin effect in the line's conductors and the eddy current losses in substrate. To extract the frequency dependence of inductances, an EM field problem in MQS regime has to be solved. The field distribution inside the conductors inserts new state variables (an infinity, if their values are considered for each z). In [28] a method named *of the two fields* is proposed, applied to extract the frequency dependent p. u. l. parameters (admittance $\mathbf{Y}$ and impedance $\mathbf{Z}$) for lossy multiconductor transmission lines, from the field solutions. Based on them a method to extract a parametric reduced-order model for this system is generated. Another approach to include in the TL model the eddy currents effect is based on the substitution of the longitudinal p. u. l. resistance with the approximate ladder model developed in Section 5.6.

If we denote

$$\mathbf{u} = \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & -\mathbf{Z}(s) \\ -\mathbf{Y}(s) & \mathbf{0} \end{bmatrix}, \tag{5.33}$$

then (5.32) can be written as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}z} = \mathbf{A}\mathbf{u}, \tag{5.34}$$

and its solution is

$$\mathbf{u}(z, s) = \exp(\mathbf{A}z)\mathbf{u}(0, s). \tag{5.35}$$

If the quantities of the line ends are denoted by

$$\mathbf{u}_- = \begin{bmatrix} \mathbf{v}_- \\ \mathbf{i}_- \end{bmatrix} = \begin{bmatrix} \mathbf{v}(0, s) \\ \mathbf{i}(0, s) \end{bmatrix}, \quad \mathbf{u}_+ = \begin{bmatrix} \mathbf{v}_+ \\ \mathbf{i}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{v}(l, s) \\ \mathbf{i}(l, s) \end{bmatrix}, \tag{5.36}$$

then

$$\mathbf{u}_+ = \mathbf{T}\mathbf{u}_-, \quad \text{where } \mathbf{T} = \exp(\mathbf{A}l) = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix}. \tag{5.37}$$

The terminal operational admittance matrix $\mathbf{Y}$ can be computed from $\mathbf{T}$ as

$$\begin{bmatrix} \mathbf{i}_- \\ \mathbf{i}_+ \end{bmatrix} = \mathbf{Y} \begin{bmatrix} \mathbf{v}_- \\ \mathbf{v}_+ \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} -\mathbf{T}_{12}^{-1}\mathbf{T}_{11} & \mathbf{T}_{12}^{-1} \\ \mathbf{T}_{22}\mathbf{T}_{12}^{-1}\mathbf{T}_{11} - \mathbf{T}_{21} & -\mathbf{T}_{22}\mathbf{T}_{12}^{-1} \end{bmatrix}. \tag{5.38}$$

Here, the terminals are considered voltage-controlled and, therefore, the multiconductor line with n conductors is a MIMO-type system with $2n$ input signals (potentials of the n close and n distant terminals) and $2n$ output signals (currents through these

terminals). However, being a distributed system (with an infinite number of state variables, one-dimensionally distributed along the line $0 < z < l$), the dimension of state space is infinite. Actually, the multiconductor line meets the ECE boundary conditions (5.24)–(5.26) for an element with $2n$ terminals that are the close and distant extremities of the n conductors. The EM field within the TL is distributed in space and time, but in harmonic state, the information is concentrated in a point in the frequency domain and its one-dimensional space distribution is given by one complex number (voltage or current) related to each line terminal.

For the particular case of a line with a single conductor over the ground, which is a two-port quadrupole described by a system with distributed parameters with two input and two output signals, the global admittance has the following expression:

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \frac{\cosh(\gamma l)}{Z_c \sinh(\gamma l)} & -\frac{1}{Z_c \sinh(\gamma l)} \\ -\frac{1}{Z_c \sinh(\gamma l)} & \frac{\cosh(\gamma l)}{Z_c \sinh(\gamma l)} \end{bmatrix} = \mathbf{Y}^T, \tag{5.39}$$

where $Z_c = \sqrt{(R + sL)/(G + sC)}$, $\gamma = \sqrt{(R + sL)(G + sC)}$ are the characteristic impedance and the complex propagation constant, respectively, depending on p. u. l. parameters R, L, C, and G and frequency. The result of the presented approach related to this particular case has an analytic expression, and thus illustrates the AAM approach.

The TL approach is essential for the MOR of on-chip interconnect lines (see also Chapter 4 of this volume [5]). Although geometric reduction is very effective, this approach is not the final step in the complexity reduction; it needs to be sustained by an additional order reduction process, in order to obtain the desired reduction to a minimal finite order instead of an infinite order [30]. This section uncovered only one step in the process of complexity reduction of EM systems, that based on simplifications of geometrical structure, in which for adequate geometrical modeling it is assumed that each component of the EM field is dependent only on certain coordinates from a proper selected coordinates system. This choice, called geometrical modeling, reduces the complexity of particular EM systems in a dramatic manner.

5.5 Regimes of the electromagnetic field and models with lumped parameters

The complexity reduction of general EM systems can be done by simplifying the physics, i. e., by disregarding some phenomena. This can be acceptable under certain conditions, specific to each particular studied case. In practice such simplifications, called *a field regime*, are based on a series of hypotheses so that the theory remains coherently and rigorously mathematically formulated. This is the main advantage of using a field regime, even if the obtained model does not perfectly reflect the reality.

The EM field regime described by Maxwell's equations (5.5)–(5.8) and (5.13)–(5.15) is known as the *general electrodynamic (ED) regime* or the *full-wave regime*.

Since the magnetic flux density $\vec{B}$ is divergence-free, this field is solenoidal and thus a *magnetic vector potential* $\vec{A}$ can be defined so that

$$\vec{B} = \nabla \times \vec{A}. \tag{5.40}$$

In order to define a unique $\vec{A}$ a gauge condition is needed, which imposes a constraint on the divergence of $\vec{A}$. The simplest condition is the Coulomb gauge: $\nabla \cdot \vec{A} = 0$. By substituting (5.40) in the EM induction law (5.7) it follows that

$$\nabla \times \left(\vec{E} + \frac{\partial \vec{A}}{\partial t} \right) = \vec{0}. \tag{5.41}$$

Consequently, a *scalar potential* V can be defined so that

$$\vec{E} + \frac{\partial \vec{A}}{\partial t} = -\nabla V \quad \Rightarrow \quad \vec{E} = -\nabla V - \frac{\partial \vec{A}}{\partial t}. \tag{5.42}$$

The scalar potential V and the magnetic vector potential $\vec{A}$ are called *electrodynamic potentials* in the ED regime of the EM field. In terms of differentials, with a simplified language, the scalar potential is a 0-form $V \in W^0$ and the vector potential is a 1-form $\vec{A} \in W^1$. The perfect character of the De Rham sequence (Figure 5.2) guarantees the existence of these two potentials.

In linear media with imposed current $\vec{J}_i$, Maxwell equations conduce to second-order PDEs, expressed in potentials which are state variables [33]:

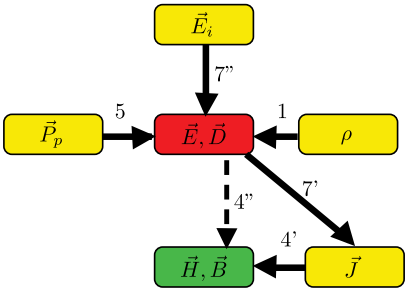
$$\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t} \Rightarrow \nabla \times (\nu \nabla \times \vec{A}) + \sigma \frac{\partial \vec{A}}{\partial t} + \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2} + \sigma \nabla V + \varepsilon \nabla \frac{\partial V}{\partial t} = \vec{J}_i, \tag{5.43}$$

$$\nabla \cdot \vec{J} = -\frac{\partial \rho}{\partial t} \Rightarrow \nabla \cdot (\sigma \nabla V) = \frac{\partial \rho}{\partial t} + \nabla \cdot \vec{J}_i, \tag{5.44}$$

where $\nu = 1/\mu$ and the gauge $\nabla \cdot (\sigma \vec{A}) = 0$ was used. Their solution is an EM field that propagates with a finite speed.

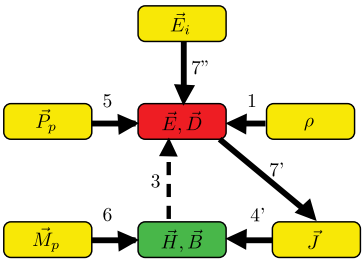
A complexity reduction of general EM systems can be achieved by eliminating some quantities that are irrelevant in certain conditions. For example, if the EM field varies sufficiently slow so that wave propagation can be ignored, one of the two field regimes known as *quasi-static* can be used [8].

The *electroquasi-static* (EQS) regime is intended especially for the study of capacitive effects, where the phenomenon of EM induction is neglected (Figure 5.6). Formally, $\mu = 0$ and thus $\vec{B} = \vec{0}$. This choice reduces by half the dimension of the state space, since the only state quantity is $\vec{D}$. An example of usage is an RCG transmission line (with null line inductance), used to model one-dimensionally distributed resistive-capacitive combined effects.



1. $\nabla \cdot \vec{D} = \rho$
3. $\nabla \times \vec{E} = 0$
4. $\Rightarrow \nabla \cdot \vec{J} = -\frac{\partial \rho}{\partial t}$
5. $\vec{D} = \varepsilon \vec{E} + \vec{P}_p$
7. $\vec{J} = \sigma(\vec{E} + \vec{E}_i)$

Figure 5.6: Causal diagram of the EQS regime.



1. $\nabla \cdot \vec{D} = \rho$
2. $\nabla \cdot \vec{B} = 0$
3. $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$
4. $\nabla \times \vec{H} = \vec{J} \Rightarrow \nabla \cdot \vec{J} = 0$
5. $\vec{D} = \varepsilon \vec{E} + \vec{P}_p$
6. $\vec{B} = \mu \vec{H} + \mu_0 \vec{M}_p$
7. $\vec{J} = \sigma(\vec{E} + \vec{E}_i)$

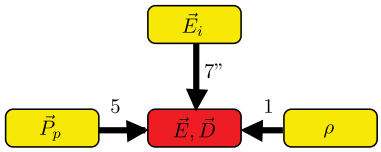
Figure 5.7: Causal diagram of the MQS regime.

The *MQS* regime is intended especially for the study of inductive effects, where the displacement current is neglected (Figure 5.7). Formally, $\varepsilon = 0$ and thus $\vec{D} = \vec{0}$. This choice reduces by half the dimension of the state space, since the only state quantity is $\vec{B}$. An example of usage is an RLG transmission line (with null line capacitance), used to model one-dimensionally distributed resistive-inductive combined effects.

The electrostatic regime, capacitance extraction

A very efficient method to reduce the complexity of EM systems is by modeling them with circuits having lumped elements (resistors, inductors, capacitors). In these models, the current and the electric and magnetic fields are segregated, and thus their energies are concentrated: electric energy in capacitors, magnetic energy in inductors, whereas the resistors do not concentrate EM energy. The fields interact only through the circuit.

The extraction of the corresponding lumped parameters (R, L, C) is carried out by assuming stationary field distributions, even if the EM field is variable in time and, consequently, the circuit will be simulated in a dynamic regime. Thus, the ES regime is used to extract capacities (C), the MG regime for extracting inductances (L), and the EC regime for extracting resistances (R).



1. $\nabla \cdot \vec{D} = \rho$
3. $\nabla \times \vec{E} = 0$
5. $\vec{D} = \varepsilon \vec{E} + \vec{P}_p$
7. $\vec{J} = \sigma(\vec{E} + \vec{E}_i) = \vec{0}$

Figure 5.8: Causal diagram of the ES regime.

In a static regime there is no time variation, no motion, and no energy transfer, so that power and currents are zero. The causal diagram and the first-order fundamental differential ES equations are shown in Figure 5.8. The nonrotational character of the electric field allows definition of the *electrostatic scalar potential*

$$\vec{E} = -\nabla V, \tag{5.45}$$

which is the solution of the *elliptical second-order PDE of Poisson type*, with a *div-grad operator* [33]:

$$-\nabla \cdot (\varepsilon \nabla V) = \rho_t, \tag{5.46}$$

where $\rho_t = \rho - \nabla \cdot \vec{P}_p$. In homogeneous media, without charges and permanent polarization, the equation becomes of Laplace type, $\triangle V = 0$.

On discontinuity surfaces between two bodies with different dielectric characteristics, the interface conditions are

$$\vec{n}_{12} \cdot (\vec{D}_2 - \vec{D}_1) = \rho_s \Rightarrow \varepsilon_1 \frac{\mathrm{d}V_1}{\mathrm{d}n} - \varepsilon_2 \frac{\mathrm{d}V_2}{\mathrm{d}n} = \rho_s - \vec{n}_{12} \cdot (\vec{P}_{p2} - \vec{P}_{p1}), \tag{5.47}$$

$$\vec{n}_{12} \times (\vec{E}_2 - \vec{E}_1) = \vec{0} \Rightarrow V_1 = V_2. \tag{5.48}$$

These interface conditions highlight the continuity of potential due to the continuity of the tangential component of the electric field strength and the continuity of the normal component of the electric induction when there is no permanent polarization and the interface is not charged.

The boundary conditions that ensure the potential uniqueness are [8]

$$\text{Dirichlet:} \quad V(P) = f_D(P), \quad P \in S_D \neq \emptyset, \quad S_D \subset \Sigma = \partial \mathcal{D}, \tag{5.49}$$

$$\text{Neumann:} \quad \frac{\partial V}{\partial n} = f_N(P), \quad P \in S_N = \Sigma - S_D. \tag{5.50}$$

The capacitive interaction between $n+1$ conductors located in a linear dielectric (Figure 5.9 (left)) is described by Maxwell's equations for capacitances:

$$\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & & \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \quad \Leftrightarrow \quad \mathbf{q} = \mathbf{CV}, \tag{5.51}$$

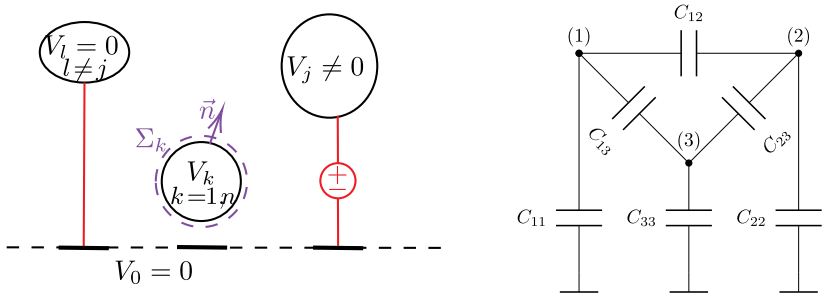


Figure 5.9: Capacitance extraction. Left: Excitation of conductors needed in the ES problem from which column j of the *nodal capacitance* matrix is computed. Right: Equivalent circuit for $n = 3$. C_{kj} are *partial capacitances* that are computed from the nodal capacitances.

where $\mathbf{q}, \mathbf{V} \in \mathbb{R}^n$ are the vectors of charges and potentials, respectively, and $\mathbf{C} = (c_{ij})_{i,j=1,n} \in \mathbb{R}^{n\times n}$ is the matrix of conductors' capacitance, also named *matrix of nodal capacitances*. The column j of the matrix $\mathbf{C}$ is extracted from the solution of a fundamental ES problem in the dielectric, with Dirichlet boundary conditions, null on the surface of all the conductors except for conductor j for which $V = V_j$ (Figure 5.9 (left)). The capacitance $c_{kj} = q_k/V_j|_{V_l=0,l\neq j}$, where $k = 1, n$ is computed from the charge

$$q_k = \oint_{\Sigma_k} \vec{D} \cdot \mathrm{d}\vec{\mathrm{A}} = -\oint_{\Sigma_k} \varepsilon \frac{\mathrm{d}V}{\mathrm{d}n}\,\mathrm{d}A, \tag{5.52}$$

which is linearly dependent on V_j. The computation of the matrix $\mathbf{C}$ needs the solving of n distinct ES field problems, obtained by exciting successively only one conductor, the other conductors being grounded.

Due to passivity and reciprocity, $\mathbf{C}$ is symmetrical, positively defined, and diagonal dominant. Its inverse $\mathbf{S} = \mathbf{C}^{-1}$ is called the *matrix of potential coefficients*. The electric energy stored in this system is

$$W_e = \frac{1}{2}\mathbf{V}^T\mathbf{q} = \frac{1}{2}\mathbf{V}^T\mathbf{C}\mathbf{V} = \frac{1}{2}\mathbf{q}^T\mathbf{V} = \frac{1}{2}\mathbf{q}^T\mathbf{S}\mathbf{q} > 0. \tag{5.53}$$

Relation (5.51) can be modeled with a capacitive circuit, having the topology of a complete polygon (Figure 5.9 (right)). By applying the nodal method, it follows that the capacitances of this circuit (which have positive values), called *partial capacitances*, can be computed from the nodal capacitances as

$$C_{kj} = -c_{kj} \quad \forall\, k, j = 1, \ldots, n, \quad k \neq j, \tag{5.54}$$

$$C_{kk} = \sum_{j=1}^{n} c_{kj} \quad \forall\, k = 1, \ldots, n. \tag{5.55}$$

This procedure is widely applied in practice to extract parasitic RC parameters in integrated circuits. The ES regime results provide the C parameters, whereas the

EC regime is similar to ES. If ε is replaced with σ, then the conductance values are obtained.

The advantage of having a model described as an equivalent capacitive circuit comes from the fact that its *complexity reduction* can be efficiently achieved by removing negligible equivalent capacitances connected between floating nodes (e. g., capacitances smaller than a threshold of 0.1% relative to capacitances to ground). This leads to a *sparse nodal capacitance matrix*, which does not lose its symmetry and positive definite, thus passive, characteristic.

Since the capacitance between two conductors is inversely proportional with the distance between them, it follows that around each conductor there is a spherical "window" containing coupled conductors. Outside this window, the capacitive coupling may be ignored. This observation simplifies the ES field problem that has to be solved to extract a (sparse) column of the nodal capacitances matrix. Even more efficient methods can be conceived if the pattern of the sparse nodal capacitance matrix is known, e. g., it is banded [35]. Such strategies are important and compulsory for integrated circuits, where the number of conductors is huge, over one million.

The magnetic stationary regime, inductance extraction

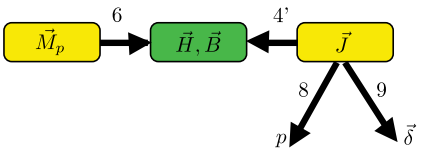
In a stationary regime there is no time variation and no motion, but energy transfer is allowed, so that power and currents are nonzero. The causal diagram and the first-order fundamental differential equations of the magnetic stationary (MG) regime are shown in Figure 5.10. In the particular case where there are no currents, this becomes the magnetostatic (MS) regime. The rotational character of $\vec{H}$ does not allow a natural definition of a scalar potential, this being the main difference between MG and MS. The solenoidal character of $\vec{B}$ allows however the definition of the *magnetic vector potential*

$$\vec{B} = \nabla \times \vec{A}, \tag{5.56}$$

which is the solution of the *second-order PDE* with a *curl-curl operator* [33]:

$$\nabla \times (\nu \nabla \times \vec{A}) = \vec{J}_t, \tag{5.57}$$

where $\vec{J}_t = \vec{J} + \nabla \times (\nu \vec{B}_r)$. As discussed in the general ED case, a gauge condition is required to ensure the uniqueness of the magnetic vector potential. In homogeneous



2. $\nabla \cdot \vec{B} = 0$
4. $\nabla \times \vec{H} = \vec{J}$
6. $\vec{B} = \mu \vec{H} + \mu_0 \vec{M}_p$

Figure 5.10: Causal diagram of the MG regime.

media, with Coulomb gauge $\nabla \cdot \vec{A} = 0$, the magnetic vector potential satisfies a Poisson vector equation:[1]

$$-\triangle \vec{A} = \mu \vec{J}_t. \tag{5.58}$$

If there are no sources, i. e., $\vec{J}_t = \vec{0}$, then the equation becomes Laplace, $\triangle \vec{A} = \vec{0}$.

On discontinuity surfaces between two bodies with different magnetic linear characteristics, the interface conditions are

$$\vec{n}_{12} \times (\vec{H}_2 - \vec{H}_1) = \vec{J}_s \Rightarrow \vec{n}_{12} \times (\nu_2 \nabla \times \vec{A}_2 - \nu_1 \nabla \times \vec{A}_1) = \vec{J}_s, \tag{5.59}$$

$$\vec{n}_{12} \cdot (\vec{B}_2 - \vec{B}_1) = 0 \Rightarrow \vec{n}_{12} \cdot (\nabla \times \vec{A}_2 - \nabla \times \vec{A}_1) = 0. \tag{5.60}$$

If there is no current sheet at the interface, from (5.59) it follows that $\vec{H}_{t1} = \vec{H}_{t2}$, and if $\nu_1 = \nu_2$, then $(\nabla \times \vec{A}_1)_t = (\nabla \times \vec{A}_2)_t$. From the Coulomb gauge it can be derived that $A_{n1} = A_{n2}$ and from (5.60) $\vec{A}_{t1} = \vec{A}_{t2}$. Consequently, the magnetic vector potential is continuous, $\vec{A}_1 = \vec{A}_2$, across interfaces.

An important practical problem is the quantitative characterization of the inductive interaction between n conductive wires. This interaction is described by Maxwell's equations for inductivities, which give the magnetic fluxes produced by these n conductors placed in a linear magnetic medium:

$$\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & & & \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{\varphi} = \mathbf{L}\mathbf{i}, \tag{5.61}$$

where $\boldsymbol{\varphi}, \mathbf{i} \in \mathbb{R}^n$ are the vectors of magnetic fluxes and currents, respectively, and $\mathbf{L} = (L_{ij})_{i,j=1,n} \in \mathbb{R}^{n \times n}$ is the matrix of inductances, holding *self- and mutual inductances*. The column j of the matrix $\mathbf{L}$ is extracted from the solution of a fundamental MG problem in the magnetic medium, where all the currents are zero, except for conductor j for which $i_j \neq 0$. The mutual inductance between wires k and j is $L_{kj} = \varphi_k / i_j|_{i_l = 0, l \neq j}$, where $k = 1, n$ is computed from the magnetic flux

$$\varphi_k = \int\limits_{S_{\Gamma_k}} \vec{B} \cdot \mathrm{d}\vec{\mathrm{A}} = \oint\limits_{\Gamma_k} \vec{A} \cdot \mathrm{d}\vec{r}, \tag{5.62}$$

which is linearly dependent on i_j.

The computation of the matrix $\mathbf{L}$ needs the solving of n distinct MG field problems, obtained by current exciting successively only one conductor, the other conductors being open.

1 $\nabla \times (\nabla \times \vec{A}) = \nabla(\nabla \cdot \vec{A}) - \triangle A = -\triangle A$, due to the Coulomb gauge.

Due to passivity and reciprocity, the matrix $\mathbf{L}$ is symmetrical, positively defined, and diagonal dominant. Its inverse is denoted by $\mathbf{K} = \mathbf{L}^{-1}$. The magnetic energy stored in this system is

$$W_m = \frac{1}{2}\mathbf{i}^T\boldsymbol{\varphi} = \frac{1}{2}\mathbf{i}^T\mathbf{L}\mathbf{i} = \frac{1}{2}\boldsymbol{\varphi}^T\mathbf{i} = \frac{1}{2}\boldsymbol{\varphi}^T\mathbf{K}\boldsymbol{\varphi} > 0. \tag{5.63}$$

In the case of wire conductors, (5.62) can be applied only for the computation of mutual inductances, since the integral is not convergent for self-inductances. The self-inductances can be computed from (5.63), where $W_m = \int_{\mathbb{R}^3} \vec{A} \cdot \vec{J}\, \mathrm{d}v$.

Relation (5.61) can be modeled with an inductive circuit with n coupled coils.

The modeling of inductive effects in integrated circuits is becoming increasingly important as their clock speeds increase. However, on-chip wiring is a large structure, which requires without doubt complexity reduction of its models. The inductance matrix is a full one, and its size is huge, in particular if wire segmentation is applied as in partial electric equivalent circuits [47]. The complexity reduction of the equivalent circuit and the matrix sparsification is more difficult for inductances than for capacitances. Sparsification of a dense or even full matrix means their approximation with a sparse matrix, which has very few nonzero elements. One straight approach to make the inductance matrix sparse is simply to discard those mutual coupling terms of $\mathbf{L}$, which are below a certain threshold. This approach, however, does not guarantee the positive definiteness of the resulting inductance matrix and therefore the model passivity is not preserved. A series of alternative methods for robust sparsification of the inductance matrix which preserve the passivity were proposed, such as the K-method [34], the vector potential equivalent circuit method [54], and the magneto-electric equivalent circuit method [29].

Reduced circuit models with lumped parameters

The simplest model of a system with n conductors in an ED regime, in which there are EM induction phenomena, and capacitive effects are neglected, can be modeled by an n-port circuit (Figure 5.11 (left)) characterized by the equations

$$\mathbf{u} = \mathbf{L}\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t} + \mathbf{R}\mathbf{i}, \tag{5.64}$$

where $\mathbf{R} = \mathrm{diag}(R_1, \ldots, R_n) \in \mathbb{R}^{n\times n}$, with R_k being the resistance of the conductor k; $\mathbf{i}, \mathbf{u} \in \mathbb{R}^n$ are the vectors of currents and voltages along the wires, respectively. These matrices are extracted from the MG and EC field solutions. The number of state variables is equal to the number of inductors, i. e., n.

Similarly, the simplest model of a system with $n + 1$ conductors in an ED regime placed in an imperfect insulator, but where the EM induction phenomena are neglected, can be modeled by a multiport circuit (Figure 5.11 (right)), characterized by the equations

$$\mathbf{i} = \mathbf{C}\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} + \mathbf{G}\mathbf{v}, \tag{5.65}$$

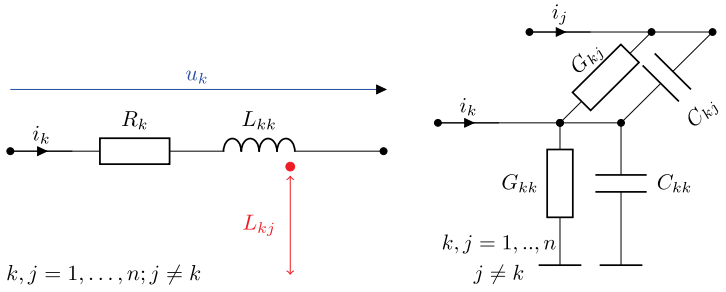


Figure 5.11: The simplest circuit model for n conductors in MQS (left – RL circuit) and EQS (right – GC circuit).

where $\mathbf{C}$ is the capacitance matrix and $\mathbf{G}$ is the matrix of conductances for resistive losses in the dielectric ($\mathbf{G} = \mathbf{C}\sigma/\varepsilon$ for homogeneous media). These matrices are extracted from the ES and EC field solutions. The number of state variables is also equal to n, being less than the number of capacitors since they are in excess, as can be easily seen (there are loops consisting solely of capacitors).

In the two models discussed above, which can be used in MQS or EQS regimes, the lumped parameters are extracted from static or stationary regimes. The results they give may be acceptable at some frequency ranges, but they might not be accurate enough at high frequencies, e. g., when eddy currents and skin effects become relevant. As will be explained in the next section, in this case circuit models with lumped parameters extracted from MQS and EQS fields can be used. Even so, neither these models are accurate enough beyond a certain frequency, where both inductive and capacitive effects have to be considered simultaneously.

It is evident that both inductive and capacitive effects are considered if the field regime used for extraction is the general ED. However, combined inductive and capacitive effects can be more easily modeled if the RL and CG circuits above are combined in T or Π models, without solving the ED field problem (Figure 5.12). These models have the order of complexity equal to $3n$.

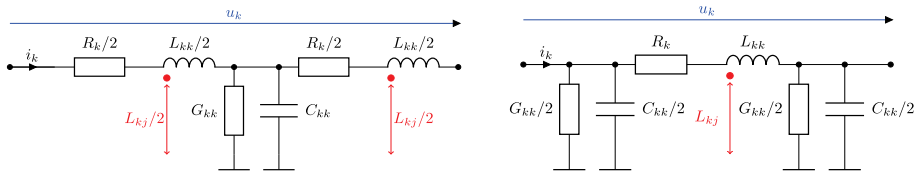


Figure 5.12: The simplest RLC circuit model for conductor k, where $k = 1, \ldots, n$: T (left) and Π (right). To simplify the figure, partial capacitances between conductors are not shown.

Even if they have inductive, capacitive, and conductive effects, the T or Π models are not able to describe the field propagation. If this effect is important, in the case of wire-shaped conductors, the line of length l may be decomposed in p segments. If each segment has a length $l/p \ll \lambda$, where λ is the wavelength (the rule of thumb is $p = 4l/\lambda$), then the propagation along a segment can be neglected. Consequently, each segment can be modeled with a T or Π schematic. Then, by connecting all segments' models, a reduced system with $(2p + 1)n$ state variables is obtained.

The great advantage of modeling with lumped parameters is that the original field problem (EQS, MQS, or ED) with an infinite state space is reduced to an I/O dynamical system with a finite number (from n up to $(2p + 1)n$) of state variables. Moreover, the number of I/O signals is also finite ($2n$ inputs and $2n$ outputs for n conductors). In three-dimensional devices, all three directions have to be "segmented," but this is carried out with numerical approaches, e. g., as discussed in Section 5.7.

Identification of the appropriate field regime

The discussion above revealed that the complexity reduction or EM system can be achieved by choosing an appropriate field regime. For this, the *characteristic times*, which depend on the material constants ε, μ, σ and the characteristic length l of the device under study, are computed and compared [44].

There are three characteristic times of an EM device: (1) *charge relaxation time* $\tau_e = \varepsilon/\sigma$ is the relaxation time of an electric field in a conductor; (2) *magnetic field diffusion time* $\tau_m = \mu\sigma l^2$ is the diffusion time of the magnetic field in a conductor; and (3) *propagation time* $\tau_{em} = l/c$, with $c = 1/\sqrt{\varepsilon\mu}$, is the time in which a wave propagating with speed c travels a distance l.

It follows that $\tau_{em}^2 = \tau_e\tau_m$. In very good conductors $\tau_e \ll \tau_{em} \ll \tau_m$, i. e., charge relaxation time is negligible and magnetic inductive effects are preponderant. In poor conductors, $\tau_m \ll \tau_{em} \ll \tau_e$, the time for magnetic field diffusion is insignificant, and the charge relaxation is important.

If we denote by τ the *characteristic time*, defined as the duration period or time constant of the phenomenon under study, then the appropriate regime can be decided by comparing τ with τ_{em}, τ_e, τ_m. The graphical representation shown in Figure 5.13 illustrates this comparison. A point in this diagram has as abscissa the decimal logarithm of τ/τ_{em} and the ordinate the decimal logarithm of τ_m/τ_{em}.

If $\tau_e < \tau_m$ (i. e., $\tau_e < \tau_{em} < \tau_m$), then the corresponding point on the map will be in the first "quadrant" of the map ($x > 1$, $y > 1$), and if $\tau < \tau_m$, an MQS regime has to be considered (e. g., the case of very good conductors), otherwise a static/stationary magnetic regime is appropriate.

If $\tau_m < \tau_e$ (i. e., $\tau_m < \tau_{em} < \tau_e$), then the corresponding point on the map will be in the fourth "quadrant" of the map ($x > 1$, $y < 1$), and if $\tau < \tau_e$, an EQS regime has to be considered (e. g., the case of poor conductors), otherwise a static/stationary electric regime is appropriate. The lines draw on the map are not very strict. In fact,

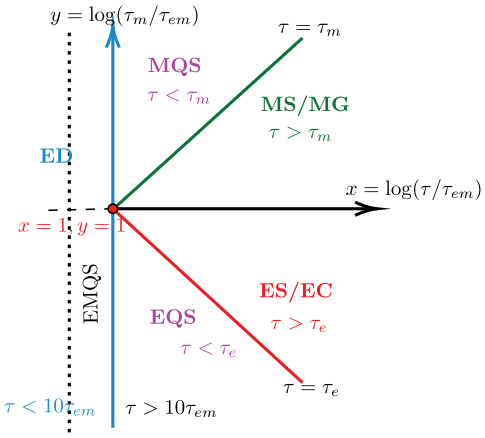


Figure 5.13: Map of EM field regimes, as a function of characteristic times.

if the phenomena are very slow ($\tau \gg \tau_e > \tau_m$ or $\tau \gg \tau_m > \tau_e$), then the stationary/static regimes will be definitely used for modeling; otherwise (for intermediary speeds) quasi-stationary regimes may be chosen. In the case of fast phenomena, there is a small band region on the left side of the $x = 1$ axis, where both charge relaxation and EM induction phenomena are considered, but the propagation is ignored since the displacement current is neglected. This is called the EM quasi-stationary (EMQS) regime.

When the phenomena are extremely fast ($\tau \ll \tau_{em}$, usually $\tau < 10\tau_{em}$ is assumed, i. e., the left semi-space with respect to the $x = 1$ line), then the wavelength is much smaller than the dimensions of the device, and therefore the ED regime must be used to describe the field propagation.

If $\tau > 10\tau_{em}$, then the discussion depends on the ordering between τ_e and τ_m, τ_{em} having a value in-between these two characteristic times being their geometric mean.

5.6 Equivalent infinite circuits of devices with distributed parameters and their finite approximations

The models with lumped parameters shown in Figure 5.11 and Figure 5.12 are valid not only for wire-shaped conductors, but also for massive conductors if the operating frequency is low enough. However, with the frequency increase, the penetration depth of the EM field diminishes, making the field regime to move progressively to MQS, EMQS, and ED, eventually. This phenomenon, called "skin effect," would make the lumped parameters of Figures 5.11 and 5.12 dependent on the frequency. However, equivalent circuits with constant lumped parameters are preferred, so that their be-

havior seen from the terminals is able to catch the high-frequency phenomena. This section shows how such an equivalent circuit can be obtained.

The infinite electric circuit equivalent to the two-pole ECE

Let us consider the ECE shown in Figure 5.3 with two terminals ($m = 2$). We will assume that the domain $\mathcal{D}$ is linear and the charge relaxation time is much smaller than the magnetic field diffusion time ($\tau_e \ll \tau_m$). Consequently, the appropriate field regime is MQS and satisfies

$$\frac{1}{\sigma}\nabla \times \left(\frac{1}{\mu}\nabla \times \vec{E}(\vec{r},t)\right) = -\frac{\partial \vec{E}(\vec{r},t)}{\partial t}, \tag{5.66}$$

with the ECE boundary conditions (5.24), (5.25), (5.26) expressed only in $\vec{E}$:

$$\vec{n} \cdot \left(\nabla \times \vec{E}(\vec{r},t)\right) = 0 \quad (\forall)\, \vec{r} \in \Sigma, \tag{5.67}$$

$$\vec{n} \cdot \vec{E}(\vec{r},t) = 0 \quad (\forall)\, P \in \Sigma - S_1 - S_2, \tag{5.68}$$

$$\vec{n} \times \vec{E}(\vec{r},t) = \vec{0} \quad (\forall)\, \vec{r} \in S_1 \cup S_2. \tag{5.69}$$

The input signal is a known voltage imposed between the two terminals:

$$\int_{C_{AB} \subset \Sigma} \vec{E} \cdot \mathrm{d}\vec{r} = u(t), \tag{5.70}$$

where $A \in S_1$ and $B \in S_2$. The output signal is the current through the terminal

$$i(t) = \int_{S_2} \sigma \vec{E} \cdot \mathrm{d}\vec{\mathrm{A}}. \tag{5.71}$$

We will use an approach specific to functional analysis, based on the *modal analysis* of this device. Let us define a linear operator $\mathcal{A}$

$$\mathcal{A}(\vec{E}) = \frac{1}{\sigma}\nabla \times \left(\frac{1}{\mu}\nabla \times \vec{E}(\vec{r},t)\right). \tag{5.72}$$

Thus the equation to be solved is

$$\mathcal{A}(\vec{E}) = -\frac{\partial \vec{E}(\vec{r},t)}{\partial t}. \tag{5.73}$$

The operator $\mathcal{A}$ is a kind of avatar of the differential curl-curl operator, densely defined on a Hilbert space $\mathcal{H}$, with compact resolvent [25]. If the domain under study is bounded and sufficiently smooth, the operator $\mathcal{A}$ with $\sigma > 0$ and $\mu > 0$ is symmetrical and positively defined in the space of solenoidal functions with null-type ECE boundary conditions. Consequently, its eigenvalues equation

$$\mathcal{A}(\vec{e}_k) = \lambda_k \vec{e}_k \tag{5.74}$$

has an infinite number of solutions $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \leq \cdots \to \infty$. These solutions are the spectrum of differential operator $\mathcal{A}$, and the corresponding eigenfunctions make an orthonormal basis of $\mathcal{H}$

$$(\vec{e}_i, \vec{e}_j) = \delta_{ij}, \quad \text{where } (\vec{e}_i, \vec{e}_j) = \int_{\mathcal{D}} \vec{e}_i \cdot \vec{e}_j \, \mathrm{d}v. \tag{5.75}$$

Thus, the solution of (5.73) for a step excitation $u(t) = U_0\, h(t)$, where $h(t)$ is the unit step, can be expressed as a generalized Fourier series of eigenfunctions:

$$\vec{E}(\vec{r}, t) = \left[\vec{E}_0(\vec{r}) + \sum_{k=1}^{\infty} c_k(t)\, \vec{e}_k(\vec{r}) \right] U_0. \tag{5.76}$$

This is a convergent series, with separate spatial and temporal variables. The term $\vec{E}_0(\vec{r})$ is the stationary field in $\mathcal{D}$, generated by the voltage $u = U_0 = 1$, with $\mathcal{A}(\vec{E}_0) = 0$. From (5.73), (5.74), and (5.76) it follows that

$$\lambda_k c_k(t) + \frac{\mathrm{d}c_k}{\mathrm{d}t} = 0 \quad \Rightarrow \quad c_k(t) = -b_k\, e^{-t/\tau_k} \quad \text{with } \tau_k = 1/\lambda_k. \tag{5.77}$$

Consequently, (5.76) becomes

$$\vec{E}(\vec{r}, t) = \left[\vec{E}_0(\vec{r}) - \sum_{k=1}^{\infty} b_k\, e^{-t/\tau_k}\, \vec{e}_k(\vec{r}) \right] U_0. \tag{5.78}$$

The initial null condition $\vec{E}(\vec{r}, 0) = \vec{0}$ imposes that

$$\vec{E}_0(\vec{r}) = \sum_{k=1}^{\infty} b_k\, \vec{e}_k(\vec{r}). \tag{5.79}$$

This means that b_k are the Fourier coefficients of the solution for the stationary regime. Finally, the ECE current is derived from (5.71), (5.78), and (5.79):

$$i(t) = \left[\sum_{k=1}^{\infty} b_k (1 - e^{-t/\tau_k}) \int_{S_2} \sigma \vec{e}_k(\vec{r}) \cdot \mathrm{d}\vec{\mathrm{A}} \right] U_0. \tag{5.80}$$

If we denote

$$R_k = 1/\left(b_k \int_{S_2} \sigma \vec{e}_k(\vec{r}) \cdot \mathrm{d}\vec{\mathrm{A}} \right) \quad \text{and} \quad L_k = R_k/\lambda_k, \tag{5.81}$$

then (5.80) can be written as

$$i(t) = \sum_{k=1}^{\infty} \frac{U_0}{R_k} (1 - e^{-tR_k/L_k}), \tag{5.82}$$

which represents the current of a ladder circuit with an infinite number of steps consisting of RL series circuits with the parameters given by (5.81).

Due to the linearity of the analyzed system, the obtained result is valid with null initial conditions for any excitation, not only for the step voltage. The infinite-ladder circuit (Figure 5.14 (left)) is therefore equivalent to the distributed parameter element $\mathcal{D}$ with MQS field and ECE boundary conditions, regardless of its excitation.

The finite approximation of infinite equivalent circuits

The modal analysis explained above is based on the computation of the problem's eigenvalues and eigenfunctions. The spectrum of the operator $\mathcal{A}$, which in theory can by continuous, is in reality a discrete one, being a numerable infinite set. Since the series (5.78) and (5.79) are convergent, the partial sums converge to the infinite series, the higher-order terms are less and less important.

The complexity of this system can be reduced by truncating the ladder to a limited number of steps, thus ignoring the irrelevant effect of the upper steps. This generates an error in the stationary regime, which can be eliminated by adding a resistance which, paralleled with those of the inferior steps, produces the stationary regime resistance R_0 (Figure 5.14 (right)): $R_{n+1} = 1/(1/R_0 - \sum_{k=1}^{n} 1/R_k)$.

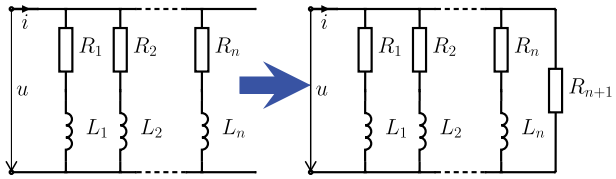


Figure 5.14: Infinite ECE equivalent circuit (left) and its corrected truncation (right).

Thus a sequence of finite circuits is obtained, which converges towards the exact infinite model, all having a correct stationary behavior. Although simple, this reduction method based on truncation is not optimal. The applied compensation corrects the value of the admittance in the stationary regime, but it affects the limit to infinity, which becomes $1/R_{n+1}$ instead of the correct null value which occurs in the absence of this modification. Since R_k increases to ∞, the error goes to zero when $k \to \infty$.

For the particular case when the element of the ECE circuit is made of a homogeneous material and has a cylindrical shape of arbitrary cross-section, the eigenvalues and eigenfunctions are precisely those of the two-dimensional Laplace operator in the cross-section with null Dirichlet boundary values. The truncation errors have been estimated for two special cases of practical significance, for which the analytic solutions are known – the plate (Figure 5.15) and the circular cylinder.

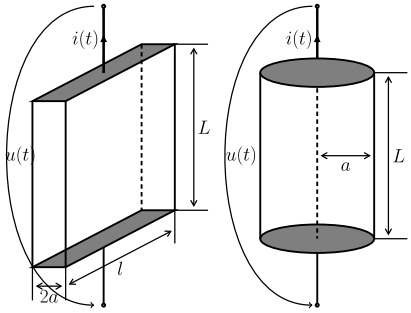


Figure 5.15: Dipolar ECE.

The expressions obtained for the lumped parameters of the infinite equivalent circuit for the plate are

$$R_0 = L/(2\sigma a l), \tag{5.83}$$

$$R_k = R_0\beta_k^2/2, \quad \text{where} \quad \beta_k = (2k-1)\pi/2, \quad k = 1, 2, \ldots, \tag{5.84}$$

$$L_k = \mu a L/(4l), \quad k = 1, 2, \ldots. \tag{5.85}$$

The inductances L_k have a constant value, while the resistances R_k increase approximately proportional with the square of k. That means a truncation error of about 1 %, if only 10 steps are retained in the finite ladder approximation.

The admittance of the plate $Y(s) = \tanh(\sqrt{s\tau})/(R_0\sqrt{s\tau})$, with $\tau = \mu\sigma a^2$, is a transcendental function versus s, with an infinite number of poles $s_k = -R_k/L_k$. In the MQS regime, the poles and zeros of the admittance (impedance) are real and negative, in a series which tends to infinity. Thus, the system is passive and stable. Figure 5.16 represents with dots the Nyquist diagram of a plate admittance $\underline{Y}(j\omega)R_0$, for $\omega\tau = 10^{-3}, 10^{-2}, \ldots, 10^{3}$.

The result of the presented approach is an analytical solution; therefore it is an illustration of the AAM. The ladder truncation is not the best method to reduce the infinite circuit to a finite one. In [32] another procedure is proposed, able to identify

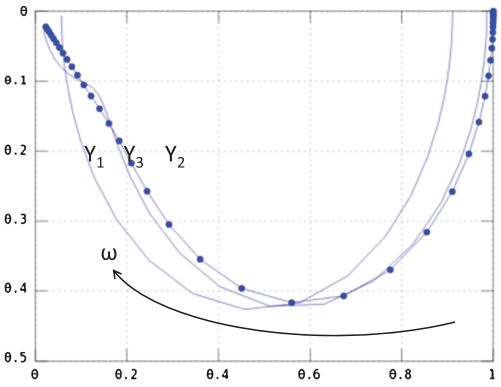


Figure 5.16: Nyquist plot of the plate admittance.

reduced-order models for various orders, which ensures an error of minimum Hankel norm. It is determined that the errors of this optimal procedure are about 10 times smaller than those of models of the same order obtained through simple truncation and compensation. Hence, the circuits from Figures 5.11 and 5.12 can model also the skin effects in conductors, if ladder circuits with four steps, three RL series, and one a simple R substitute initial RL series groups.

5.7 FEMs and their reduction

Analytical approaches are not appropriate for EM devices with complex geometries encountered in practice, in which cases numerical methods have to be used. The FEM is the most popular one, due to its ability to adapt for the solving of intricate geometries and various types of PDE that correspond to any EM field regime, or to other multiphysics fields.

The theoretical foundation of FEM is the reformulation of the problem in weak form. In this section we will show how the EM equations for various field regimes can be reformulated in weak form. Then, we will review how a corresponding discrete model can be obtained and how its order can be reduced.

The weak form of the fundamental electrostatic problem

The ES potential V defined by (5.45) satisfies the generalized Poisson equation (5.46) in a linear domain $\mathcal{D}$, with $\varepsilon > 0$, to which boundary conditions (5.49)–(5.50) are added, to ensure the potential's uniqueness. Relations (5.46), (5.49), and (5.50) represent the *strong form* of the fundamental ES problem and V is its strong solution. The weak form is obtained by projecting (5.46) on a set of independent directions of a Hilbert space $\mathcal{H}$ of "test" functions u:

$$-\int_{\mathcal{D}} \nabla \cdot (\varepsilon \nabla v) u \, \mathrm{d}x = \int_{\mathcal{D}} \rho u \mathrm{d}x, \tag{5.86}$$

where the weak solution is denoted by v and ρ_t was renamed ρ. By applying the Gauss–Ostrogradsky formula[2] it follows from (5.86) that

$$\int_{\mathcal{D}} \varepsilon (\nabla v) \cdot (\nabla u) \, \mathrm{d}x = \int_{\mathcal{D}} \rho u \mathrm{d}x + \int_{\partial \mathcal{D}} \varepsilon u \frac{\mathrm{d}v}{\mathrm{d}n} \, \mathrm{dA}. \tag{5.87}$$

This form of equation is also called *Galerkin projection*, and the procedure for obtaining it *method of moments*, since the moments are defined by the products between the residual and test functions, that have to be null.

---

2 $\nabla \cdot ((\varepsilon \nabla v) u) = \nabla \cdot (\varepsilon \nabla v) u + \varepsilon (\nabla v) \cdot (\nabla u)$, then apply $\int_{\mathcal{D}}$ and use Gauss–Ostrogradsky: $\int_{\mathcal{D}} \nabla \cdot ((\varepsilon \nabla v) u) \, \mathrm{d}x = \int_{\partial \mathcal{D}} ((\varepsilon \nabla v) u) \cdot \vec{n} \, \mathrm{d}A$.

By defining a bilinear functional (form) $a(\cdot,\cdot)$ and a linear functional $f(\cdot)$, relation (5.87) imposes the equality between these two functionals for any test function u having zero values on the Dirichlet boundary:

$$a(u,v) = f(u) \quad \forall u \in \mathcal{H}, \tag{5.88}$$

where

$$a(u,v) = \int_{\mathcal{D}} \varepsilon(\nabla v)\cdot(\nabla u)\,\mathrm{d}x, \tag{5.89}$$

$$f(u) = \int_{\mathcal{D}} \rho u \mathrm{d}x + \int_{S_N} \varepsilon u f_N\,\mathrm{dA}, \quad u(S_D) = 0. \tag{5.90}$$

The test functions u satisfy a null Dirichlet condition on S_D, while the solution v satisfies a nonnull condition on S_D. Therefore, in $\mathcal{H}$ there are two subspaces: one of *test* functions $\mathcal{H}_0$ with null Dirichlet conditions, which is a linear space, and the other one of *trial* functions $\mathcal{H}_D$, in which the solution is searched for; $\mathcal{H}_D$ is an affine (and not a linear) subspace of $\mathcal{H}$ containing the functions that satisfy the nonnull Dirichlet condition f_D in (5.49). For a correct definition of the bilinear functional $a(\cdot,\cdot)$, the space $\mathcal{H}$ will be considered a Sobolev space of square Lebesgue integrable functions on $\mathcal{D}$: $\mathcal{L}^2(\mathcal{D})$, which has generalized, square integrable derivatives (gradient), isomorphic with $\mathcal{W}^0$, the domain of the grad operator. The *inner product* in $\mathcal{H}$, denoted by $(u,v)_{\mathcal{H}}$, defines the norm in that space:

$$\begin{aligned} &\mathcal{H} \equiv \mathcal{H}^1 = \{u \in \mathcal{L}^2(\mathcal{D}) \mid \nabla u \in \left(\mathcal{L}^2(\mathcal{D})\right)^3\};\\ &(u,v)_{\mathcal{H}} = \int_{\mathcal{D}} uv\,\mathrm{d}x + \int_{\mathcal{D}} (\nabla u)\cdot(\nabla v)\,\mathrm{d}x \quad \Rightarrow \quad \|v\|_{\mathcal{H}} = \sqrt{(v,v)_{\mathcal{H}}}. \end{aligned} \tag{5.91}$$

The weak form of the ES problem is (5.88). It addresses the two boundary conditions Dirichlet and Neumann in a very different manner. The Dirichlet condition is "strongly" imposed from the start, such that the solution will be searched for in a set of functions, which satisfy this condition. On the other hand, the Neumann boundary condition, which is explicitly included in the expression of the linear functional (5.90), is "weakly" satisfied as well as possible. For this reason we say that the Dirichlet condition (strongly accomplished) is *essential*, while the Neumann condition (weakly accomplished) is *natural*. In the particular case of null Dirichlet conditions ($f_D = 0$), the two spaces, that of possible solutions (trial function space, named Ansatz in German publications) and that of test functions

$$\mathcal{H}_0^1 = \{u \in \mathcal{L}^2(\mathcal{D}) \mid \nabla u \in \left(\mathcal{L}^2(\mathcal{D})\right)^3, u(S_D) = 0\} \subset \mathcal{H}^1, \tag{5.92}$$

$$\mathcal{H}_D^1 = \{u \in \mathcal{L}^2(\mathcal{D}) \mid \nabla u \in \left(\mathcal{L}^2(\mathcal{D})\right)^3, u(S_D) = f_D\} \subset \mathcal{H}^1, \tag{5.93}$$

are identical, $\mathcal{H}_D = \mathcal{H}_0$. By comparing the two forms, strong and weak, for the ES equations, we observe the following *advantages of the weak form*:

- In the strong form there are second-order differentials, while in the weak form the differentials are only of the first order, which impose fewer restrictions to functions.
- In the strong form, the boundary conditions are separately written, while in the weak form they are included in the equation, in the definition of the functionals, explicitly for the natural condition and implicitly for the essential one.
- The strong equation is verified for each point in the computing domain, while in the weak form the equation is globally verified (for each basis test function).
- In the weak form, the interface conditions on discontinuity surfaces are automatically fulfilled and they do not need to be explicitly imposed, which is the case for the strong form of the equations.
- The material parameter may have an arbitrary spatial change in the weak format, since it does not need to be derivable.

These advantages make the use of functional analysis instead of classical calculus worthwhile.

Well-posed fundamental electrostatic problems

For the functional framework given by (5.88)–(5.93), the conditions required by the Lax–Milgram theorem (discussed in Chapter 1 of [4]) are satisfied [8]. This theorem states that if the bilinear functional $a(u, v)$ defined on the Hilbert space $\mathcal{H}$ is *bounded*, i. e., a positive constant C exists so that

$$\left|a(u, v)\right| \leq C\|u\|\|v\|, \tag{5.94}$$

and *coercive*, i. e., it a real positive constant c exists so that

$$\left|a(u, u)\right| \geq c\|u\|^2 \tag{5.95}$$

for any elements u and v in $\mathcal{H}$, then equation $a(u, v) = f(u)$ has a solution in $\mathcal{H}$ for any f from $\mathcal{H}'$ (dual space of $\mathcal{H}$, defined as the space of linear functionals over $\mathcal{H}$), which is unique and bounded, i. e., $\|v\| \leq \|f\|/c$, where c is the coercivity constant.

This theorem has a crucial role, since it guarantees proper expression of the ES problem in its weak form, providing the *existence*, *uniqueness*, and *well-conditioning* with the Lipschitz constant $L = c$ of the solution. Consequently, this result concludes the mathematical modeling of ES problems.

Since the bilinear functional $a(\cdot, \cdot)$ is symmetrical and positive, the solution of the weak form minimizes the *energy* convex functional $F(v)$ given by (5.96), a statement known as the *Dirichlet principle*:

$$F(v) = \frac{1}{2}a(v, v) - f(v) \quad \Rightarrow \quad \mathrm{d}F = \frac{1}{2}\big(a(\mathrm{d}v, v) + a(v, \mathrm{d}v)\big) - f(\mathrm{d}v), \tag{5.96}$$

$$\min F(v) \Leftrightarrow \mathrm{d}F = 0 \quad \Leftrightarrow \quad a(u, v) = f(u), \quad \forall u = dv \in \mathcal{H}. \tag{5.97}$$

This is the proof that, in the case of elliptical div-grad PDE equations (e. g., Laplace and Poisson equations), the Galerkin projective formulation (5.88) and Ritz minimizing formulation (5.96) are equivalent. For this reason, the solution of the weak form is also called the *variational approach*.

Finite element method principles

The central concept of the FEM is the use of the weak form given by (5.88) for a finite-dimensional subspace of $\mathcal{H}$, denoted as $\mathcal{H}_h$, suitable for a computing system. The goal is to find a numeric, discrete solution v_h, characterized by a finite number of degrees of freedom. The numerical solution satisfies

$$a(u_h, v_h) = f(u_h), \quad \forall u_h \in \mathcal{H}_h \subset \mathcal{H}, \tag{5.98}$$

where $a(\cdot,\cdot)$ and $f(\cdot)$ are defined by (5.89) and (5.90). Therefore, the conditions in the Lax–Milgram theorem are still fulfilled, so the approximate discrete problem (5.98) is also well-posed, having a solution that is unique and is continuously dependent on data. This is yet another advantage of the weak reframing of the field problems.

The finite-dimensional space $\mathcal{H}_h$ is generated by a finite set of basis functions also named *shape functions* or *test functions*; $\mathcal{H}_h$ consists of the set of linear combinations of these basis functions, so to identify the numerical solution in the case of the null Dirichlet boundary conditions, it is enough to find a finite number of degrees of freedom, coordinates of the solution in this space $\mathcal{H}_h$. In FEM, the basis functions are defined starting with a disjoint (nonoverlapping) partitioning of the computing domain in a mesh of cells having simple geometrical shapes (the most frequent are triangles or quadrilaterals in two dimensions and tetrahedrons or hexahedrals in three dimensions), called finite elements, with maximum size h. The basis functions are complete polynomials of degree p, i. e., they contain all terms $x^i y^j (z^k)$ with respect to the Cartesian coordinates x and y (and z in three dimensions, compared to two dimensions, where $k = 0$), with degree $i + j + k < p + 1$. In each cell, the polynomial interpolates the values of the potential in a minimum number of nodes, points located on the element boundary. This choice ensures the continuity of the potential at the transition through interfaces between elements, this being an essential FEM condition, called *inter-element compatibility*.

The floating nodes are located in the interior of the computing domain and on the Neumann boundary of the domain, while those located on the Dirichlet boundary are not floating, since they have an imposed, fixed known potential. The degrees of freedom resulting from solving a system of linear algebraic equations are those potential values in the floating nodes which ensure the best global approximation of the exact solution. The number of floating nodes N is the number of degrees of freedom of the problem, as each floating node defines a basis function which is a polynomial of degree p for every element, having a unitary value for that node and zero value for all

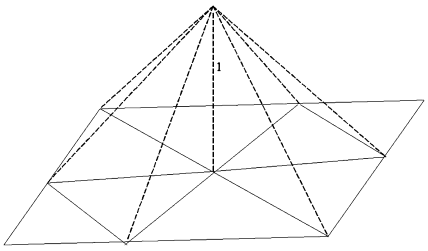


Figure 5.17: Basis function in linear two-dimensional FEM.

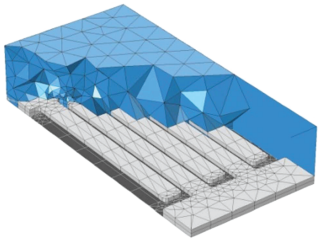


Figure 5.18: Example of a three-dimensional mesh (Comsol).

other floating nodes. For the simplest case, when $p = 1$, called *linear (first-order) elements*, the nodes are located in the vertices of the triangles/tetrahedrons. The basis function φ_k associated with the node k is null over the entire computing domain except for the elements which have that node in common. For the two-dimensional case with $p = 1$, the k basis function has a pyramidal graph with unity height for node k and the base formed by the assembly of triangles which have this node in common (Figure 5.17).

The maximum size for all edges, denoted by h, gives the mesh norm, and describes the level of mesh refining. The degree p of basis polynomials gives the finite element order. The most frequent orders are linear ($p = 1$), quadratic ($p = 2$), and cubic ($p = 3$). We say that the shape functions constitute a *nodal base* if the degrees of freedom are the values of the numerical solution in the nodes, i. e., the basis functions have the values $\varphi_k \in \mathcal{H}_h$, $\varphi_k(n_j) = \delta_{kj}$, $j = 1, \ldots, N$, in triangulation nodes $n_1, n_2, \ldots, n_N$. For each cell, the number of polynomial coefficients is equal to the number of nodes. For example, in two dimensions, the linear triangular finite elements have three nodes placed in vertices, and the quadratic elements have six nodes, three nodes being added in the middle of the edges. The coefficients are associated to $(1, x, y)$ in the linear two-dimensional case and to $(1, x, y, x^2, y^2, xy)$ in the quadratic two-dimensional case. In three dimensions (Figure 5.18), the linear finite elements have four nodes placed in tetrahedron vertices, the polynomial being a linear combination of $(1, x, y, z)$. The quadratic tetrahedral elements have 10 nodes, three nodes in vertices, and six in the middle of the edges, the polynomial being generated by $(1, x, y, z, x^2, y^2, z^2, yz, xz, xy)$. In practice quadrilateral (hexahedral) meshes are used, having only nodes on the boundaries of elements, internal nodes being removed, to-

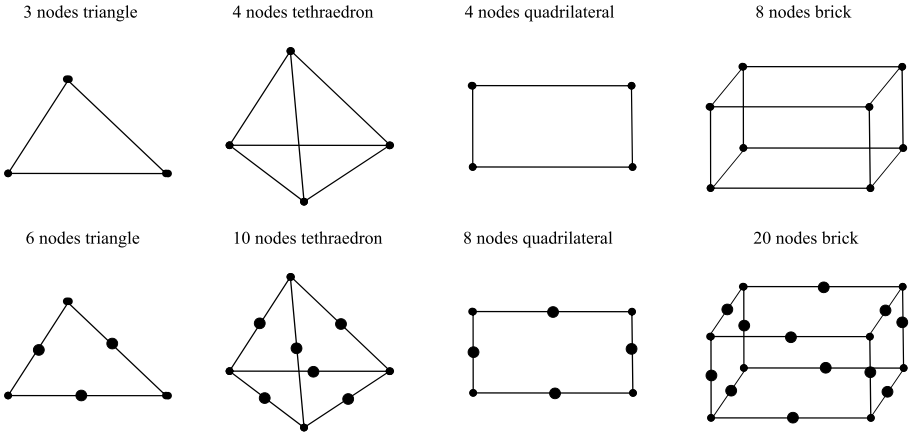


Figure 5.19: Typical elements and their nodes.

gether with highest-order terms of the shape functions. They are called "serendipity elements" and have only eight (20) nodes (Figure 5.19).

In FEM a frequently used concept is that of *barycentric coordinates*, which in triangles and tetrahedrons give indication about the distances to the vertices. In the case of a triangle, there are three barycentric coordinates: λ_i, associated with vertices $i = 1, 2, 3$, which have affine variation inside the triangle. The coordinate λ_i has unitary value in the vertex i, and it is zero in the other two nodes. They identify the position of any point inside the triangle, although actually only two of them is enough, since the sum of all barycentric coordinates is unitary ($\lambda_1 + \lambda_2 + \lambda_3 = 1$). In the same manner the four barycentric coordinates of a tetrahedron are defined. The nodal shape function of first order associated with the node i has in each adjacent triangle/tetrahedron just the value $\varphi_i = \lambda_i$. With these selections, any element of the solutions' space with null Dirichlet conditions has the following expression:

$$u_h = \sum_{j=1}^{N} u_j^h \varphi_j \quad \forall u_h \in \mathcal{H}_h \subset \mathcal{H}_0. \tag{5.99}$$

By replacing (5.99) in (5.98), a system of N linear algebraic equations is obtained:

$$\mathbf{Au} = \mathbf{b}, \tag{5.100}$$

where

$$\mathbf{A} = [a(\varphi_i, \varphi_j)]; \quad \mathbf{b} = [f(\varphi_i)]; \quad \mathbf{u} = [u_j^h]. \tag{5.101}$$

The solution of this system is the vector of potential values in the floating nodes. The matrix **A**, called the *rigidity (stiffness) matrix*, and the right-hand side term **f** have the

following elements:

$$a_{ij} = a(\varphi_i, \varphi_j) = \int_{\mathcal{D}} \varepsilon(\nabla\varphi_i) \cdot (\nabla\varphi_j)\, \mathrm{d}x,$$
$$b_{ki} = f(\varphi_i) = \int_{\mathcal{D}} \rho\varphi_i\, \mathrm{d}x + \int_{S_N} \varepsilon f_N \varphi_i\, \mathrm{dA}. \tag{5.102}$$

The matrix **A** is symmetrical, positive definite, and sparse due to the limited support of the basis functions. The integrals in (5.102) are estimated analytically (Holland–Bell) or, more frequently, numerically with Gaussian quadrature [58]. Since this quadrature is exact for low-degree polynomials, it gives exact values for the integrals (5.102) if appropriate quadrature nodes are used.

Considering the properties of matrix **A**, the system of linear equations (5.100) can be solved with direct methods (such as Cholesky factorization type LLT, LDLT, or SVD) or iterative methods (such as conjugate gradient, without or with various preconditioning). Regardless of the solving method, the condition number of the rigidity matrix dictates the accuracy of the solution and the efficiency of computations. There are many preconditioning techniques, which avoid the ill-conditioning. In particular, multimesh preconditioning (called also multigrid) is an extremely efficient one, giving a strong acceleration of the iterative methods [10]. The essential part of the rigidity matrix **A** can be identified by truncating SVD factorization of that matrix. It is an efficient technique for the order reduction of the FEM model (see also Chapter 2 of [4]).

The linear system (5.100) is just the nodal equation of a capacitive circuit, connected according to the FEM mesh. In the EC case, the equivalent circuit contains resistances. These circuits may be reduced by star-polygon transforms which remove the internal nodes. This process corresponds to the Gaussian elimination applied to the internal nodes in (5.100). The resulting circuit has fewer nodes, but usually, the number of branches is increased, because the matrix is no longer a sparse one. To further reduce the resulted circuit, several techniques to sparsify the matrix can be applied, by clustering the nodes which have equal or almost equal potentials.

The nodes on the Dirichlet boundary have known potentials, which in the general case may be nonnull, and need to be carefully addressed in the numerical approach, i. e., to preserve the symmetry of the matrix. If we separate the nodes on the Dirichlet boundary, we obtain for floating nodes a system of linear equations with a form similar to that of the problem with null Dirichlet conditions:

$$u_h = \sum_{k\in S_D} f_{Dk}\varphi_k + \sum_{j=1}^{N} u_j^h \varphi_j \quad \forall\varphi_k \in \mathcal{H}_h \subset \mathcal{H}; f_{Dk} = f_D(P_k), P_k \in S_D, \tag{5.103}$$

$$a(u_h, \varphi_i) = f(\varphi_i) \quad \Rightarrow \quad a\left(\sum_{j=1}^{N} u_j^h \varphi_j, \varphi_i\right) = f(\varphi_i) - a\left(\sum_{k\in S_D} f_{Dk}\varphi_i, \varphi_k\right), \tag{5.104}$$

$$\mathbf{Au} = \mathbf{b}, \tag{5.105}$$

$$\mathbf{A} = [a(\varphi_i, \varphi_j)], \quad \mathbf{b} = [f(\varphi_i) + f_{Di}], \quad \mathbf{u} = [u_j^h], \tag{5.106}$$

$$a_{ij} = a(\varphi_i, \varphi_j) = \int_{\mathcal{D}} \varepsilon(\nabla\varphi_i) \cdot (\nabla\varphi_j)\, \mathrm{d}x, \tag{5.107}$$

$$b_i = f(\varphi_i) + f_{Di} = \int_{\mathcal{D}} \rho\varphi_i\, \mathrm{d}x + \int_{S_N} \varepsilon f_N \varphi_i\, \mathrm{dA} - \sum_{k \in S_D} a_{ik} f_{Dk}. \tag{5.108}$$

This time, the right-hand side term has three contributions: internal field sources (charge density inside the cells); boundary natural conditions (type Neumann, addressed similarly with a superficial charge density); and the boundary essential condition (type Dirichlet, addressed similarly with the floating nodes potentials, but imposed forcefully). In this manner, the numerical solution will satisfy exactly the essential Dirichlet boundary conditions, while the Neumann, natural boundary conditions are satisfied as exactly as possible. Consequently, these results conclude the step of numerical modeling.

FEM, from theory to code

If we suppose that the problem was previously detailed and well formulated, knowing the computational domain, the solved equation, values of the material constant, and the field sources, including the boundary conditions, the general FEM has the following steps of the computational modeling stage:

1. *Domain discretization* selects the shape of the finite elements and generates the mesh which covers the computational domain, determining the mesh norm h.
2. *Selection of the element type* consists of choosing the shape functions, in particular the elements order p and node placement.
3. The *weak formulation step* identifies the bilinear and linear functionals of the weak form, considering the equation to be solved and the boundary condition.
4. The *derivation of element matrices* step identifies the coordinate transformation which maps the local coordinates of a common reference element to global ones, and thus its Jacobian matrix, and selects the variant of the Gaussian quadrature method used to compute element contributions to both functionals.
5. *Assemblage of element equations* computes and localizes the contribution of each element to the system matrix and to the right-hand side term, evaluating both bilinear and linear functionals, respectively, treating with a different technique in particular the Dirichlet boundary conditions. The use of sparse matrix techniques is highly recommended since it produces superior performances in execution. In most software packages, steps 4 and 5 are encapsulated, being hidden from the user.
6. *Solution of equations* is a step in which the linear system generated previously is solved with a direct or an iterative method. As will be seen, in the case of transient

problems, the discretized equations to be solved are of ODE type, which require a numerical integration in time. A different approach is used to find the eigenvalues in the case of modal analysis, when the problem is reduced to finding the eigenvectors and eigenvalues of a matrix.

7. *Order reduction* may be realized by iterative cycling of this procedure and/or by processing simulation results or the state equations generated with the FEM.

Examples of simple FEM codes illustrating these steps can be found in [20].

In FEM programs, most often, coefficients (5.102) are not computed directly, but through a bijective transformation from triangular, local coordinates to physical, global coordinates. This transformation maps a standard *reference element* (e. g., a right angle triangle with unity legs, a unitary square, a unitary tetrahedron with orthogonal, unity legs, or a unitary cube, with *natural, local coordinates*), to elements in the physical space (called *global coordinates*), being determined by the correspondence of the nodes in these two spaces. These elements are called *isoparametric*. As the main advantage, this approach allows the encapsulation and separate development of the reference element library. When implemented on a computer, a difference is acknowledged between the basis functions, defined on the reference cell, and the shape functions, defined in the real physical space, such that each shape function is associated to a degree of freedom from the global grid.

The derivation of element matrices and the assemblage of element equations are the core steps of any FEM program. The combination of two ideas, isoparametric elements and Gaussian numerical quadrature, allows a simple and uniform treatment of the elemental assembly procedure, even for higher-order elements and elements with curved boundaries [18].

Convergence of the finite element method

The convergence of the FEM guarantees that the numerical solutions will tend toward the exact one when the discretization mesh is uniformly refined and thus the number of degrees of freedom tends to infinity.

The convergence towards the exact solution assumes that the discrete, finite-dimensional space $\mathcal{H}_h$ tends to an infinite-dimensional space, dense in $\mathcal{H}$. This convergence condition, named *completeness*, is obtained if the shape functions are continuous (ensuring the inter-element compatibility) and they have the unity partition property. Under the conditions of the Lax–Milgram theorem, the completeness guarantees the convergence [18].

A theorem which provides the rate of convergence of the numerical solution v_h in FEM assumes that the shape functions are polynomials of p degree over each element of a mesh having the norm h, satisfying the inequality [55]

$$\|v - v_h\|_{\mathcal{H}^m} < C\, h^{p+1-m} \|v\|_{\mathcal{H}^{p+1}} \tag{5.109}$$

if the exact solution v, continuous on the entire domain, is sufficiently smooth.

This relation known as *a priori error estimators* guarantees the global convergence of the FEM, showing that if the norm h of the discretization mesh tends to zero (which implies that the number of degrees of freedom N tends to infinity), then the numerical solution v_h tends to the exact one v. We observe that (5.109) ensures a global, "average" convergence of the numerical solution and its m derivatives towards the exact solution on the entire domain, as opposed to a local, pointwise convergence, characteristic of strong formulations of problems. If $m = 0$, it refers to potential, and if $m = 1$, it refers to the field strength.

The numerical solution is trusted if it is relatively independent of the mesh, i. e., h is sufficiently low, so that the error (5.109) has an acceptable level. The mesh independence may be checked by refining the current mesh, e. g., by halving h. If the numerical solution does not change too much, the mesh is adequate; otherwise the refining process continues. This approach is named uniform mesh (or "h") refining. If new nodes are added in the middle of each edge, face, or cell, the number of degrees of freedom increases just as when passing from the first-order to the second-order FEM. Although the rigidity matrices have the same size, in the first-order refined case it has fewer nonzero elements. The additional computational effort to generate and solve the system in the case of the quadratic FEM order is rewarded by a more accurate and smoother numerical solution, since the order of convergence for the field is just the order p of the elements.

Figure 5.20 (left) illustrates the convergence of the FEM in the first- and second-order cases. The faster convergence of the second-order elements ($p = 2$), related to that of the first order ($p = 1$), is obvious. According to this figure, we observe that in practice, the numerical error cannot be made lower than a limit, which is about 10^{-8} to 10^{-10}. If the individual mesh elements start to get very small, we run into the limits of numerical precision. That is, the numbers in our model are smaller than can be accurately represented on a computer. This is an inherent problem with all computational

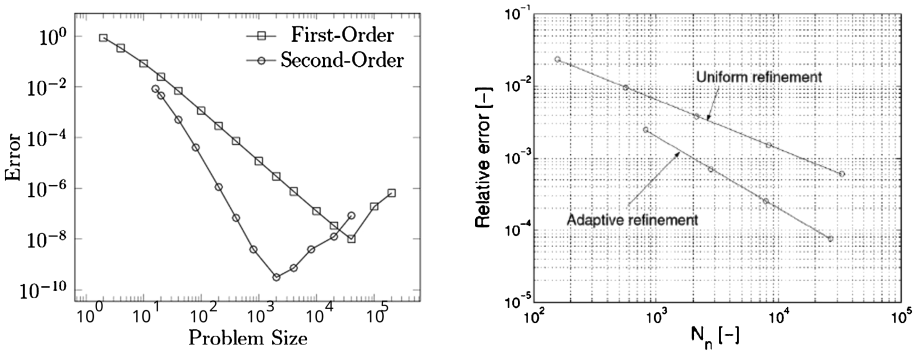


Figure 5.20: Left: Relative error vs. number of degrees of freedom (https://www.comsol.com/). Right: Relative AMR error vs. number of nodes [6].

methods, not just the FEM; computers cannot represent all real numbers accurately. Moreover, the condition number of the FEM matrix grows with mesh refinement at least linearly (or faster, depending on the problem). Therefore, it is important for the users to be aware of this rapidly growing accumulated round-off error as they refine meshes, especially with very large models. One million degrees of freedom in three dimensions is a usual limit for mesh refinement.

Adaptive mesh refinement

The nonsmooth problems have a lower convergence rate. One method to recover the depreciation of convergence caused by singularities is to use a self-adaptive algorithm to refine the discretization mesh. This technique is called *adaptive mesh refinement* (AMR) or the *adaptive FEM* [6]. This iterative algorithm begins from a coarse mesh and solves the problem on a series of meshes, adding successively new nodes, only when and where necessary. Consequently, the number of elements and the number N of degrees of freedom are progressively increased, but not so fast as in the uniform refinement. At each iteration, certain elements, specifically those for which the solution has an unacceptable error, are refined through segmentation. For example, in two dimensions, a triangle can be divided in two, three, or four subtriangles, depending on how supplemental nodes (zero or one) are appended in the middle of each edge.

Figure 5.20 (right) shows the mode in which the relative error decreases with respect to the number of nodes N (equal to the number of degrees of freedom), in the case of a uniform refinement of the mesh and in the case of an adaptive refinement. In the first case the convergence rate is approximately 0.7, two times lower than $q = 1.4 \approx 4/3 < 2$ (which is the ideal value in the absence of the singularity). These results were obtained for the example shown in Figure 5.21, where the discretization mesh is generated by AMR. It can be seen that the mesh is selectively refined, especially near the vertex that generates the singularity.

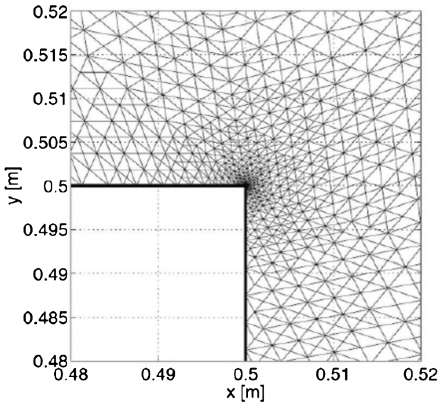


Figure 5.21: Multimesh FEM with adaptive refinement [6].

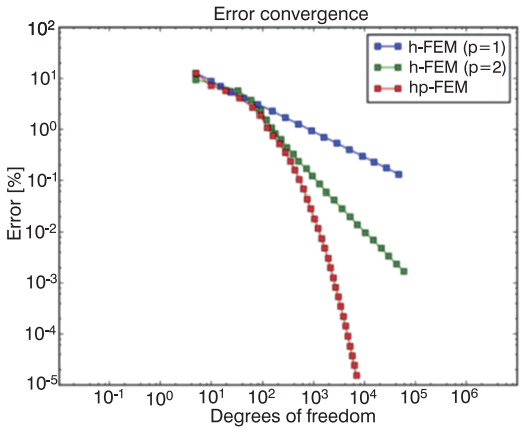


Figure 5.22: FEM convergence h for $p = 1$ (blue), h for $p = 2$ (red), and hp (green) [22].

Finite element with *hp*-refinement

As previously observed, another method for error reduction of numerical solutions is to increase the degree of the polynomials, which define the shape functions. The refinement by successively increasing of the degree p provides very good results for the case of extremely smooth solutions (which are analytical functions, indefinitely differentiable functions, harmonic functions, solutions to the Laplace equation, etc.), for which the error diminishes exponentially with the increase in the degree p. However, it is useless in the case of nonsmooth problems. Combination of h- and p-adaptive refinements regains the accelerated convergence for problems with singularities. Its justification is based on the observation that in zones where the solution is smooth, it is more efficient to use high values for the degree p, while in zones adjacent to the singularities, the spatial refinement through reduction of the h parameter is more efficient [41]. The hp algorithms are super-convergent, ensuring an exponential decrease of the error, which cannot be achieved with h or p strategies alone. As can be seen in Figure 5.22, the L-shape problem is solved through h-refinement at a convergence rate of 2/3; through p-refinement at a rate of 5/4, and through hp-refinement with an exponential convergence, for which real numbers C, c, and δ exist so that

$$\|v - v_h\| < C\, e^{-cN^{\delta}}. \tag{5.110}$$

Through hp-refinement field relative errors of the order of 10^{-6} are obtained, i. e., numerical solutions with six correct significant digits, with fewer than 10^4 degrees of freedom. Since uniform h-refinement requires a much larger number of degrees of freedom to achieve a similar accuracy, the hp-refinement may be considered as being the most efficient method for complexity reduction of numerical FEM solutions.

AMR and hp-refinement are techniques which allow the reduction of the number of degrees of freedom, for a given accuracy. Therefore, they are improvements of the numerical model, which provides *on-the-fly* MOR, the latter being the most efficient known technique of this category.

The weak form of curl-curl equations

Static and stationary magnetic regimes (MS, EC) are similar to the ES regime for non-charged bodies, and their scalar potential satisfies the same generalized Poisson div-grad equation, which can be solved as discussed above. The magnetostationary (MG) regime is an exception, which needs a magnetic vector potential, since in this case a native scalar potential does not exist. Essentially, a PDE of curl-curl type (5.57) is solved to determine the magnetic field, whereas for all other static and stationary regimes, a PDE equation of div-grad type (5.46) is solved.

The weak form of MG regime equations with mixed boundary conditions is obtained by projecting the residual on the space of test functions, which are now vectors. This set is actually the Sobolev space of the vector square integrable functions, which have a curl of integrable square:

$$\boldsymbol{\mathcal{H}}_0(\mathrm{curl}, \mathcal{D}) = \{\vec{u} \in \left(\mathcal{L}^2(\mathcal{D})\right)^3 \mid \nabla \times \vec{u} \in \left(\mathcal{L}^2(\mathcal{D})\right)^3, \vec{n} \times \vec{u} = \vec{0} \text{ on } S_B\}. \tag{5.111}$$

In this functional framework, the variational formulation of the problem searches for the solution $\vec{v}$, such that

$$a(\vec{u}, \vec{v}) = f(\vec{u}) \quad \forall \vec{u} \in \boldsymbol{\mathcal{H}}_0(\mathrm{curl}, \mathcal{D}), \tag{5.112}$$

where

$$a(\vec{u}, \vec{v}) = \int_{\mathcal{D}} \nu(\nabla \times \vec{v}) \cdot (\nabla \times \vec{u}) \, \mathrm{d}x, \tag{5.113}$$

$$f(\vec{u}) = \int_{\mathcal{D}} \vec{J} \cdot \vec{u} \, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{u} \, \mathrm{dA}. \tag{5.114}$$

We will proceed as in the scalar case, where it was found that the solution v with null essential boundary conditions was sought in the linear space $\mathcal{H}_0$, as a function with such essential boundary conditions. If the boundary condition was nonnull, the solution v was sought in the affine space resulting from translating $\mathcal{H}_0$, with elements which meet that nonnull condition. This time, according to (5.114) the essential boundary condition $\vec{A}_t = (\vec{n} \times \vec{A}) \times \vec{n}$ is that on $S_B \subset \partial\mathcal{D}$, while on $S_H = \partial\mathcal{D} - S_B$ natural boundary conditions $\vec{H}_t = (\vec{n} \times \nu \nabla \times \vec{A}) \times \vec{n} = \vec{J}_s$ are imposed, and the vector solution must be sought in a curl-conform subspace, in which the vector fields conserve their tangential components when crossing discontinuity interfaces.

However, regrettably in this instance of three-dimensional MG, the functional $a(\vec{u}, \vec{v})$ is not coercive, and the Lax–Milgram theorem cannot be directly applied. Consequently, the nongauged magnetic potential vector, solution of equation (5.112), is not unique. An exception is the two-dimensional plan-parallel case, with two magnetic field components in the problem's plane and a sole component for current, and vector potential (orthogonal to that plane), which is automatically gauged, having a null divergence. Moreover, the sole component of vector $\vec{A}$ satisfies a curl-curl-type

equation which devolves in one of type div-grad. To surmount this standoff, there are several solutions:

- **Impose gauge.** For the magnetic potential vector, this is most simply achieved with the Coulomb condition ($\nabla \cdot \vec{A} = 0$) and by adding the boundary condition $A_n = 0$ on S_B.
- **Do regularization.** Another approach, called regularization, consists in adding a coercive term to the bilinear functional, meaning that a *mass matrix* is added to the stiffness matrix, weighted with a parameter k:

$$a(\vec{u}, \vec{v}) = \int_{\mathcal{D}} \nu(\nabla \times \vec{v}) \cdot (\nabla \times \vec{u}) \, \mathrm{d}x + k \int_{\mathcal{D}} \vec{v} \cdot \vec{u} \, \mathrm{d}x. \tag{5.115}$$

  Due to this term, the coercivity condition is met again, and the problem is well formulated, no matter how small is the parameter k. Then, the solution is moved towards the limit for $k \to 0$.
- **Use Lagrange multipliers.** In the case of the weak form of curl-curl type, another possible approach consists of extending the Lax–Milgram theorem such that it can be applied to variational problems with Lagrange multipliers, which do not have minimum points, but have critical points of saddle type. Such an extension is achieved by the Brezzi theorem, which highlights the conditions to have a well-posed weak formulation of the equation for vector potential [55].
- **Use reduced scalar potential.** The idea behind this method is to extract a field with a curl from the solution, which is precisely the given current density, such that what is left is a nonrotating field, solution of an MS problem. Thus, the problem accepts a scalar magnetic potential, called *reduced*. After its numerical determination by solving a div-grad equation, this field is added to the field extracted initially, resulting in the numerical solution of the MG problem. Other expressions with different potentials of the magnetic field in MG and MQS regimes are proposed in [37].
- **No gauging.** This is a very interesting approach in which the potential vector $\vec{A}$ is left without gauge and its indeterminate equation is solved numerically with an iterative method of Krylov type, which has native minimization properties. In addition to surprisingly reaching a unique magnetic field, the iterations are faster convergent than in the gauging case [26].

Analysis of the MG field with edge elements

When computing resonance frequencies for cavities, false resonant modes, called *spurious modes*, emerge when nodal finite elements are used. This phenomenon, called *spectral pollution*, disappears when the vector solution is sought in a curl-conform space. The reason is that in $\mathcal{H}_1(\mathcal{D})$ the kernel of the curl-curl operator is not properly represented [6]. The response is to use shape functions appropriate for the vector problem, called *edge elements* [8]. In three dimensions a vector field is represented by the

three coupled scalar fields, components of the vector field:

$$\vec{T} = T_x\vec{e}_x + T_y\vec{e}_y + T_z\vec{e}_z, \tag{5.116}$$

where $\vec{e}_x$, $\vec{e}_y$, and $\vec{e}_z$ are the unit vectors of the Cartesian system of coordinates. In principle, each component T_x, T_y, and T_z could be represented by combinations of nodal shape functions, but as mentioned before, this approach should be avoided. It does not guarantee the conservation of the normal component of the field flux density (or equivalently, the tangential component of the magnetic vector) either when passing from an element to another, or on the discontinuity surfaces. An alternative is to seek numerical solutions resulting from the expression

$$\vec{T}(x,y,z) = \sum_{i=1}^{E} t_i\vec{w}_i(x,y,z), \tag{5.117}$$

for which the degree of freedom t_i is the integral of the vector field $\vec{T}$ along the edge l_i from the assessed triangulation with E floating edges. Therefore, *vector shape functions* $\vec{w}_i$ are defined by

$$\int_{l_j} \vec{w}_i(x,y,z)\,\mathrm{ds} = \delta_{ij}. \tag{5.118}$$

Consequently, the vector shape function $\vec{w}_i$ is tangentially oriented along the edge l_i and is normally oriented along the other edges of element K which includes l_i. For this reason, such basis functions are called *edge elements*. Furthermore, in these circumstances the vector shape function $\vec{w}_i$ has in three dimensions the tangential component null on any face which does not include the edge i. The encountered restrictions cause two triangles in two dimensions with a common edge to have on the interface through this edge the same tangential component of the shape functions, but a different normal component. In three dimensions the vector shape functions conserve the tangential component where passing from a tetrahedron to another by way of their common face. As a result, the tangential component of these "edge" vector shape functions is continuous, in all computing domains, for which reason they are also called *vector shape functions with continuous tangent*. This curl-conformance makes them appropriate for representing the electric or magnetic field strength, but also for the magnetic potential vector (in which case the conservation of the normal component of magnetic induction is automatically ensured, where passing through any surface, hence the solenoidal character of this quantity).

The vector functions in barycentric coordinates are

$$\vec{w}_{ij} = \lambda_i\nabla\lambda_j - \lambda_j\nabla\lambda_i, \tag{5.119}$$

valid both in two dimensions, where $i,j = 1,2,3$, and in three dimensions, where $i,j = 1,2,3,4$, and where ij is the edge connecting the nodes i and j. These functions,

called Whitney elements [55], are appropriate to be selected as shape functions for the magnetic potential vector, since they have a constant tangential component of $1/l_{ij}$ on the ij edge and null on the other edges, divergence zero on the entire K element, and constant curl for that element: $\nabla \times \vec{w}_{ij} = 2\nabla\lambda_j \times \nabla\lambda_i$. The degrees of freedom associated with these functions are the integral values on characteristic edges [8].

FEM of MG field

The numerical solution $\vec{v}_h$ of the MG problem, described by the vector $\mathbf{t} = [t_i]$, is found by solving the following system of linear equations:

$$\mathbf{At} = \mathbf{b}, \tag{5.120}$$

where

$$\mathbf{A} = [a(\vec{w}_i, \vec{w}_j)]; \quad \mathbf{b} = [f(\vec{w}_i)]; \quad \mathbf{t} = [t_j^h], \tag{5.121}$$

with

$$\begin{aligned} a_{ij} &= \int_{\mathcal{D}} \nu(\nabla \times \vec{w}_i) \cdot (\nabla \times \vec{w}_j)\, \mathrm{d}x, \\ b_i &= \int_{\mathcal{D}} \vec{J} \cdot \vec{w}_i\, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{w}_i\, \mathrm{dA}, \end{aligned} \tag{5.122}$$

where $\nu = 1/\mu$, $\vec{J}$ is the current density, and $\vec{J}_s$ is related to $\vec{H}_t$ on the boundary.

Again, the central part of the FEM algorithm consists of assembling the matrix $\mathbf{A}$ of the linear system and of the vector $\mathbf{f}$ of the right-hand terms. To do this, elements of triangulation are sequentially traversed and the contribution (5.122) of each element K is added to the rigidity matrix and to the right-hand side term. Then, this linear system is solved with direct or iterative methods and the degrees of freedom t_j are determined, one for each floating edge. The numerical solution is the linear combination of the basis functions (of the edge), with these degrees of freedom as coefficients

$$\vec{v}_h = \vec{A}(x, y, z) = \sum_{j=1}^{E} t_j \vec{w}_j(x, y, z) = \mathbf{t}^T \vec{\mathbf{w}}. \tag{5.123}$$

The linear system (5.120) is similar to the mesh equations of the system of partial inductances, connected according the FEM mesh. Its robust reduction is discussed in [17].

FEM of the MQS field in the time domain

The established functional framework finds its application in the MQS regime as well, in which there are two approaches: in the time domain and in the frequency domain.

The weak form of the MQS equation in the time domain with interface conditions and boundary conditions identical with those in the MG regime and with the initial condition for the transient potential vector $\vec{A}$ is obtained by the Galerkin projection

$$a(\vec{u},\vec{v}) = f(\vec{u}) \quad \forall \vec{u} \in \boldsymbol{\mathcal{H}}_0(\text{curl},\mathcal{D}), \tag{5.124}$$

where

$$a(\vec{u},\vec{v}) = \int_{\mathcal{D}} \nu(\nabla\times\vec{v})\cdot(\nabla\times\vec{u})\,\mathrm{d}x, \tag{5.125}$$

$$f(\vec{u}) = \int_{\mathcal{D}} \left(\vec{J}_t - \sigma\left(\frac{\partial\vec{v}}{\partial t} + \nabla V\right)\right)\cdot\vec{u}\,\mathrm{d}x + \int_{S_H} \vec{J}_s\cdot\vec{u}\,\mathrm{dA}. \tag{5.126}$$

Actually, the difference in MQS is that, to the current density, the field source in MG, is added the induced current density. In additions, the electroconduction (EC) equation $\nabla\cdot(\sigma\nabla V) = 0$ needs to be solved for the scalar potential V, with Dirichlet boundary conditions (the potential value $V = f_D$) or Neumann boundary conditions (the value of the normal derivative of the potential $\mathrm{d}V/\mathrm{d}n = f_N$, given by the normal component of the current density J_n of the injected in the surface). In the weak form, the scalar V solution is sought in $\mathcal{H}^1_D$ given by (5.93) such that

$$b(u,v) = g(u), \quad \forall u \in \mathcal{H}^1_0, \tag{5.127}$$

where $\mathcal{H}^1_0$ is given by (5.92) and

$$b(u,v) = \int_{\mathcal{D}} \sigma(\nabla v)\cdot(\nabla u)\,\mathrm{d}x, \tag{5.128}$$

$$g(u) = \int_{S_N} \sigma u f_N\,\mathrm{dA}. \tag{5.129}$$

The dependency between the scalar potential V and the input signals (boundary conditions – boundary potential and the currents injected in the boundary) is instantaneous, without any delay, since V is the solution of a scalar-elliptical problem without time variable. Using a discretization with edge elements, the vector potential is expressed as $\vec{A} = \mathbf{t}^T\vec{\mathbf{w}}$, where $\mathbf{t}$ is the vector of degrees of freedom and $\vec{\mathbf{w}}$ is the vector of shape (edge) functions. A system of first-order ODEs is obtained by performing the Galerkin projection on this finite-dimensional space of test functions:

$$\mathbf{E}\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}t} = \mathbf{S}\mathbf{t} + \mathbf{f}, \tag{5.130}$$

where

$$\mathbf{S} = [s_{ij}] = \int_{\mathcal{D}} \nu(\nabla\times\vec{w}_i)\cdot(\nabla\times\vec{w}_j)\,\mathrm{d}x, \tag{5.131}$$

$$\mathbf{f} = [f_i] = \int_{\mathcal{D}} (\vec{J}_t - \sigma \nabla V) \cdot \vec{w}_i \, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{w}_i \, \mathrm{dA}, \tag{5.132}$$

$$\mathbf{E} = [e_{ij}] = \int_{\mathcal{D}} \sigma (\nabla \vec{w}_i) \cdot (\nabla \vec{w}_j) \, \mathrm{d}x. \tag{5.133}$$

Thus, if ECE type boundary conditions are used, an I/O linear time-invariant dynamic system is defined, which has as state vector the degrees of freedom of the edge elements (circulations of the vector potential $\vec{A}$ along the floating edges of the discretization mesh). Its state matrix is actually the rigidity matrix **S**, **E** (representing damping) is the descriptor matrix, and the input signals are the injected currents or the applied potentials. Due to reciprocity and passivity, matrices **S** and **E** are symmetrical and positive definite, and the real and negative eigenvalues λ_k of the matrix $\mathbf{E}^{-1}\mathbf{S}$ give the system's time constants $\tau_k = -1/\lambda_k$. The system (5.130) is similar to the system of state equations in descriptor form of the RL circuits, which have the topology described by the FEM mesh. The reduction of such circuits is treated in [19].

Now it becomes apparent that the application of the *hp*-refinement, which ensures a minimal number of degrees of freedom for an acceptable accuracy, is an *on-the-fly* MOR approach. However, there is a supplementary restriction: in order to ensure a good accuracy, the cell size h must be smaller than the field penetration depth ($h < \delta = \sqrt{2/(\omega\mu\sigma)}$). By applying *a posteriori MOR* classic methods (e. g., Krylov, balanced truncation, modal analysis of data, or circuit reduction) to the system (5.130), a model with a more reduced order is obtained eventually. A posteriori MOR is a feature encountered in all important FEM software packages (see Chapter 13 of this volume [5]), such as ANSYS or COMSOL [46].

FEM of the ED field in the time domain and MOR

In the case of the ED regime, the magnetic vector potential satisfies the hyperbolic PDE:

$$\nabla \times (\nu \nabla \times \vec{A}) + \sigma \frac{\partial \vec{A}}{\partial t} + \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2} + \sigma \nabla V + \varepsilon \nabla \left(\frac{\partial V}{\partial t} \right) = \vec{J}_i, \tag{5.134}$$

with boundary conditions (5.21) and (5.22) and initial conditions for $\vec{A}$, $\partial\vec{A}/\partial t$, and V to have a unique solution if $\nabla \cdot \vec{A} = 0$. The weak form of (5.134) is obtained by Galerkin projection:

$$a(\vec{u}, \vec{v}) = f(\vec{u}) \quad \forall \vec{u} \in \mathcal{H}_0(\mathrm{curl}, \mathcal{D}), \tag{5.135}$$

where

$$a(\vec{u}, \vec{v}) = \int_{\mathcal{D}} \nu (\nabla \times \vec{v}) \cdot (\nabla \times \vec{u}) \, \mathrm{d}x, \tag{5.136}$$

$$f(\vec{u}) = \int_{\mathcal{D}} \left(\vec{J}_t - \sigma \left(\frac{\partial \vec{v}}{\partial t} + \nabla V \right) - \varepsilon \left(\frac{\partial^2 V}{\partial t^2} + \nabla \left(\frac{\partial V}{\partial t} \right) \right) \right) \cdot \vec{u} \, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{u} \, \mathrm{dA}, \tag{5.137}$$

to which the weak form of the EC equation for the scalar potential V is added, as in the MQS case.

By using a discretization with edge elements, the potential vector is $\vec{A} = \mathbf{t}^T \vec{\mathbf{w}}$, where $\mathbf{t}$ is the vector of degrees of freedom and $\mathbf{w}$ is the vector of shape-edge functions. The weak form becomes the system of second-order ODEs

$$\mathbf{M} \frac{\mathrm{d}^2 \mathbf{t}}{\mathrm{d}t^2} + \mathbf{E} \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}t} - \mathbf{S}\mathbf{t} = \mathbf{f}, \tag{5.138}$$

where $\mathbf{S}$ is given by (5.131), $\mathbf{E}$ is given by (5.133), and

$$\mathbf{f} = [f_i] = \int_{\mathcal{D}} \left(\vec{J}_t - \sigma \nabla V - \varepsilon \frac{\partial \nabla V}{\partial t} \right) \cdot \vec{w}_i \, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{w}_i \, \mathrm{dA}, \tag{5.139}$$

$$\mathbf{M} = [m_{ij}] = \int_{\mathcal{D}} \varepsilon (\nabla \vec{w}_i) \cdot (\nabla \vec{w}_j) \, \mathrm{d}x. \tag{5.140}$$

This time, the system is characterized by three matrices – of mass $\mathbf{M}$, of damping $\mathbf{E}$, and of stiffness $\mathbf{S}$. The elements of these matrices are the inner products of shape functions (or their derivatives) weighted with several material constants. For instance, in $\mathbf{M}$ (characteristic to the inertial phenomena in mechanical systems), the dielectric permittivity describes the capacitive effects; in $\mathbf{E}$ (characteristic to the damping phenomena), the conductivity describes the conduction effect; and in $\mathbf{S}$ the magnetic permeability describes the inductive effects, similar to elastic stiffness/rigidity.

The system (5.138) is similar to the equations of the RLC circuits which have the topology described by the FEM mesh. The reduction of such circuits is treated extensively in the literature [40, 48, 49]. Several methods for order reduction of second-order ODE systems of (5.138) type are also proposed in the literature [2]. The similarity to mechanical systems (see also Chapter 2 of this volume [5]), also suggested by the names of the three matrices, allows the use of reduction methods from structural analysis in electrodynamics, including the static condensation of matrices $\mathbf{S}$ and $\mathbf{M}$, called Guyan reduction. This reduction consists of the selection of a reduced number of master degrees of freedom and the removal of the other state variables from the stationary regime equation, in which the field sources associated with the omitted variables are ignored. Retaining for the dynamic regime as well only the most important, master variables, the state equation reduces correspondingly its dimensions. It is expected that the first eigenvalues will not be changed substantially and, therefore, its dynamic behavior will not be fundamentally affected. The *hp*-refinement ensures *on-the-fly* MOR. However, this time, for a good accuracy, the cells dimension needs to be much smaller than the wavelength of the field $h \ll \lambda$. The numerical integration

over time for this system can be made also by discretizing the time derivative with finite differences. The Courant–Friedrichs–Lewy (CFL) condition compels the time step to be smaller than the time necessary for the EM wave to travel through one cell. It ensures the numerical stability of the computation, limiting the size of the time step used in the numerical integration over time ($\Delta t < h/c$). It is observed that, as for the explicit scheme, the factorization of the mass matrix $\mathbf{M}$ becomes necessary. An approach which removes this effort consists of approximating the matrix $\mathbf{M}$ with a diagonal one, an operation called mass lumping, which has sufficient precision only for rectangular cells (squares, hexahedrals), but not for simplex cells (triangles, tetrahedrons). For the case of triangles/tetrahedrons it is preferable to use the implicit Newmark scheme:

$$\mathbf{M}\frac{\mathbf{t}_{k+1}-2\mathbf{t}_k+\mathbf{t}_{k-1}}{h^2} = -\mathbf{E}\frac{\mathbf{t}_k-\mathbf{t}_{k-1}}{h} + \mathbf{S}(\theta\mathbf{t}_{k+1}+(1-2\theta)\mathbf{t}_k+\theta\mathbf{t}_{k-1}) + \mathbf{f}(\theta\mathbf{t}_{k+1}+(1-2\theta)\mathbf{t}_k+\theta\mathbf{t}_{k-1}). \tag{5.141}$$

This scheme is stable for any time step if $\theta \geq 1/4$. However, the solution loses accuracy if the time step exceeds the limit imposed by the CFL condition [6]. If $\theta = 0$, the scheme becomes that of explicit integration over time with finite differences.

Both simulation and order reduction are more difficult in the ED regime than in the MQS and EQS regimes, since now the system eigenvalues are complex. For this case there is a large collection of methods for a posteriori MOR of the system generated by FEM [11, 56, 50, 57, 45, 36, 53, 7, 1, 51]. They are based on truncated balanced models (see Chapter 2 of [3]), moment matching (see Chapter 3 of [3]), or data-driven and interpolation (see Chapter 6 of [3]). The research in the area of MOR of EM devices functioning with ED or MQS fields continues, since finding the best technique to solve this problem is still an open problem. The passivity enforcement of the reduced model is discussed in Chapter 5 of [3].

FEM of the ED field in the frequency domain and MOR

In the time-harmonic ED regime, the complex (operational) magnetic potential satisfies a complex equation of Helmholtz type, obtained after applying the Laplace transform to (5.134) with zero initial conditions:

$$\begin{aligned} &\nabla\times\big(\nu\nabla\times\vec{A}(s)\big)+(\sigma+\varepsilon s)\big(s\vec{A}(s)+\nabla V(s)\big)=\vec{J}_i(s),\\ &\nabla\cdot\big(\sigma\nabla V(s)\big)=s\rho. \end{aligned} \tag{5.142}$$

The weak forms of these equation are obtained by Galerkin projection:

$$a(\vec{u},\vec{v}) = f(\vec{u}) \quad \forall\vec{u}\in\boldsymbol{\mathcal{H}}_0(\mathrm{curl},\mathcal{D}), \tag{5.143}$$

$$b(u,v) = g(u) \quad \forall u\in\mathcal{H}_0^1(\mathcal{D}), \tag{5.144}$$

where

$$a(\vec{u},\vec{v}) = \int_{\mathcal{D}} \nu(\nabla\times\vec{v})\cdot(\nabla\times\vec{u})\,\mathrm{d}x + s\int_{\mathcal{D}}(\sigma+\varepsilon s)\vec{v}\cdot\vec{u}\,\mathrm{d}x, \tag{5.145}$$

$$f(\vec{u}) = \int_{\mathcal{D}} (\vec{J}_t - (\sigma + \varepsilon s)\nabla V) \cdot \vec{u}\, \mathrm{d}x + \int_{S_H} \vec{J}_s \cdot \vec{u}\, \mathrm{dA}, \tag{5.146}$$

$b(u, v)$ is given by (5.128), and

$$g(u) = \int_{\mathcal{D}} s\rho u\, \mathrm{d}x + \int_{S_N} \sigma u f_N\, \mathrm{dA}. \tag{5.147}$$

Here the vector potential $\vec{A}$ is labeled as $\vec{v}$ and the scalar potential V is labeled as v. In the case of the ECE, with null initial conditions, S_B means $A_t = 0$ on the entire boundary, whereas S_D means that the scalar potential is given on the terminals. In this case

$$f(\vec{u}) = -\int_{\mathcal{D}} (\sigma + \varepsilon s)(\nabla V) \cdot \vec{u}\, \mathrm{d}x, \tag{5.148}$$

$$g(u) = \int_{S_N} \sigma u f_N\, \mathrm{dA} = 0. \tag{5.149}$$

We need to note that, as opposed to the case of elliptical equations, for which the Galerkin projection formulation is equivalent to the minimizing Ritz formulation, in the case of hyperbolic equations this equivalence disappears. Now the solution in the variational formulation corresponds to a critical point which, however, is not anymore a minimum of the energy functional, since the energy functional is not a convex one, having a saddle shape in the critical point, of the solution.

As for the MQS case, the numerical ED solution is sought with nodal elements for the scalar electric potential and with edge elements for the magnetic vector potential, by solving the following systems of linear complex equations:

$$a(\vec{u}_h, \vec{v}_h) = f(\vec{u}_h) \quad \forall \vec{u}_h \in \mathcal{H}_{0h}(\mathrm{curl}, \mathcal{D}), \tag{5.150}$$

$$b(u_h, v_h) = g(u_h) \quad \forall u_h \in \mathcal{H}^1_{0h}(\mathcal{D}), \tag{5.151}$$

where f and g are linear complex functionals, which generate the right-hand side term of the system of equations, while a and b are bilinear functionals (a is complex and b is real), which generate after the discretization the system matrix. It is the complex transform of (5.138):

$$(s^2\mathbf{M} + s\mathbf{E} - \mathbf{S})\mathbf{t}(s) = \mathbf{f}(s) \quad \Rightarrow \quad \mathbf{t}(s), \tag{5.152}$$

$$\vec{A}(x, y, z, s) = \mathbf{t}^T(s)\vec{\mathbf{w}}(x, y, z). \tag{5.153}$$

The total number of degrees of freedom is equal to the number of floating edges, but before solving a linear system with size equal to the number of floating nodes, to find the scalar potential. These complex systems may be solved with direct or iterative methods. Unfortunately, this time, the multimesh approach is not applicable because

it is not convergent, but the methods of Krylov type are efficient, especially with preconditioning.

The solution of system (5.152) depends linearly on the right-hand side term and, therefore, in the case of the ECE, with null initial conditions, in the absence of internal sources, the solution vector and, implicitly, the output signals depend linearly on the excitations of the terminals. The linearity allows the correct definition of the admittance **Y** and, implicitly, computation of the scattering, complex operational matrix **S**. Again, in this instance the numerical solution to the problem for a series of frequencies allows the determination of the frequency characteristics of the EM systems $\underline{Y}(j\omega)$. On this base, applying MOR techniques based on data, such as vector fitting (see Chapter 8 of [3]) very efficiently reduced models of these systems can be extracted [13].

FEM of the ED field in the frequency domain with ECE boundary conditions

The previous formulation used the pair $(\vec{A}, V)$ in every point of the computational domain. Alternatively, a formulation in $(\vec{E}, V)$ can be used, as described in [21], where V is defined also inside the domain, or a formulation in $(\vec{E}, V)$, where the electric field $\vec{E}$ is defined strictly inside the domain and the electric scalar potential V is described solely on $\partial\mathcal{D}$. The following weak equation for $\vec{E}$ is obtained [15]:

$$\int_{\mathcal{D}} \left[(\nu\nabla\times\vec{E})\cdot(\nabla\times\vec{E}') + \mathrm{j}\omega(\sigma+\mathrm{j}\omega\varepsilon)\vec{E}\cdot\vec{E}'\right]\mathrm{d}x = \mathrm{j}\omega\sum_{k\in\mathcal{I}_c} V_k' I_k, \tag{5.154}$$

where $\mathcal{I}_c$ is the set of indices of current excited terminals. The equation for the electric potential on the boundary is obtained by projecting the normal component of the total current density $J_n \overset{\text{not}}{=} (\nabla\times\underline{\mathbf{H}})\cdot\mathbf{n}$ onto a set of scalar test functions $\underline{V}'$:

$$\oint_{\partial\mathcal{D}} (\nabla\times\vec{H})\cdot\vec{n}\,V'\,\mathrm{d}s = \oint_{\partial\mathcal{D}} J_n V'\,\mathrm{d}s = \sum_{k=1}^{m}\int_{S_k} J_n V'\,\mathrm{d}s = \sum_{k\in\mathcal{I}_c} V_k' I_k,$$

where $\vec{E}\in\mathcal{H}_E$, $V\in\mathcal{H}_V$, $\vec{E}'\in\mathcal{H}_{E,0}$, $V'\in\mathcal{H}_{V,0}$, and

$$\mathcal{H}_E = \Big\{\mathbf{u}\in\mathcal{H}(\mathrm{curl},\Omega)\mid \mathbf{n}\times(\mathbf{u}\times\mathbf{n}) = -\nabla_2\underline{V}' \text{ on } \partial\Omega,\ \underline{V}'\in\mathcal{H}_V,$$
$$\mathbf{n}\times(\mathbf{u}\times\mathbf{n}) = \mathbf{0} \text{ on } \bigcup_{k=1}^{m} S_k\Big\},$$

$$\mathcal{H}_{E,0} = \Big\{\mathbf{u}\in\mathcal{H}(\mathrm{curl},\Omega)\mid \mathbf{n}\times(\mathbf{u}\times\mathbf{n}) = -\nabla_2\underline{V}' \text{ on } \partial\Omega,\ \underline{V}'\in\mathcal{H}_{V,0},$$
$$\mathbf{n}\times(\mathbf{u}\times\mathbf{n}) = \mathbf{0} \text{ on } \bigcup_{k=1}^{m} S_k\Big\},$$

$$\mathcal{H}_V = \{u\in\mathcal{H}(\mathrm{grad},\partial\Omega)\mid u = \underline{V}_k \text{ on } S_k,\ k\in\mathcal{I}_v,$$
$$u = \text{constant(unkown) on } S_k,\ k\in\mathcal{I}_c\},$$

$$\mathcal{H}_{V,0} = \{u \in \mathcal{H}(\text{grad}, \partial\Omega) \mid u = 0 \text{ on } S_k,\ k \in \mathcal{I}_v,\ u = \text{constant(unkown) on } S_k,\ k \in \mathcal{I}_c\}.$$

The model complexity is reduced since the degrees of freedom are scalar quantities associated to edges strictly inside the domain and nodes on the boundary.

5.8 Conclusions

The optimal design of EM field devices requires the solving of Maxwell PDEs. Consequently, the state of such a device is completely described by local and instantaneous vector quantities, i. e., by a system with an infinite number of degrees of freedom depending on both space and time. Classical MOR approaches can be applied only to finite-dimensional systems, and that is why complexity reduction of Maxwell-based models is needed. Complexity reduction is obtained by discretization of field quantities and conduces to a model described by a finite number of degrees of freedom, expressed as a system of ODEs. The accuracy and order of the reduced EM model depend not only on the chosen *a posteriori* MOR method, but also on the *a priori* and *on-the-fly* complexity reduction approaches. A posteriori reduction, i. e., the reduction after the discretization, can be carried out with one or several methods described in [3, 4], but the complexity reduction can be carried out only with methods specific to electromagnetism, which were described in this chapter: ECE boundary conditions, an appropriate EM field regime, geometrical simplification, and discretization methods and meshes adapted to the problem. This chapter described FEM discretization, which is the most popular, but other approaches are also used in practice, such as the boundary element method and the finite integration technique/finite difference method, which generate systems of ODEs that can be reduced with the same methods as the ones used for the FEM.

Bibliography

[1] M. Ahmadloo and A. Dounavis, Parameterized model order reduction of electromagnetic systems using multiorder Arnoldi, *IEEE Trans. Adv. Packaging*, **33** (4) (2010), 1012–1020.

[2] Z. Bai and Y. Su, Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method, *SIAM J. Sci. Comput.*, **26** (5) (2005), 1692–1709.

[3] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.

[4] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, 2020.

[5] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 3: Applications*, De Gruyter, Berlin, 2020.
[6] A. Bondenson, T. Rylander, and P. Ingelstrom, *Computational Electromagnetics*, Springer, 2005.
[7] R. U. Borner, O. G. Ernst, and K. Spitzer, Fast 3-d simulation of transient electromagnetic fields by model reduction in the frequency domain using Krylov subspace projection, *Geophys. J. Int.*, **173** (3) (2008), 766–780.
[8] A. Bossavit, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*, Academic Press, 1998.
[9] A. Bossavit, *Electromagnétisme en vue de la modélisation*, Math. Appl., vol. 14, Springer-Verlag, 2003.
[10] W. L. Briggs, H. Van, and S. McCormick, *A Multigrid Tutorial*, SIAM, 2000.
[11] A. C. Cangellaris and L. Zhao, Model order reduction techniques for electromagnetic macromodelling based on finite methods, *Int. J. Numer. Model.*, **13** (2–3) (2000), 181–197.
[12] G. Ciuprina, D. Ioan, R. Janssen, and E. van der Heijden, MEEC models for RFIC design based on coupled electric and magnetic circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **34** (3) (2015), 395–408.
[13] G. Ciuprina, D. Ioan, I. A. Lazar, and B. Dita, Vector fitting based adaptive frequency sampling for compact model extraction on hpc systems, *IEEE Trans. Magn.*, **48** (2) (2012), 431–434.
[14] G. Ciuprina, D. Ioan, A. S. Lup, L. M. Silveira, A. Duca, and M. Kraft, Simplification by pruning as a model order reduction approach for RF-MEMS switches, *Compel*, **39** (2) (2019), 511–523.
[15] G. Ciuprina, D. Ioan, M. Popescu, and A. S. Lup, Electric circuit element boundary conditions in the finite element method for full-wave frequency domain passive devices, in W. H. A. Schilders (eds.) *Scientific Computing in Electrical Engineering. Mathematics in Industry*, pp. 1–8, under review.
[16] G. Ciuprina, J. Fernández Villena, D. Ioan, Z. Ilievski, S. Kula, E. J. W. ter Maten, K. Mohaghegh, R. Pulch, W. H. A. Schilders, L. Miguel Silveira, A. Stefanescu, and M. Striebel, *Parameterized Model Order Reduction*, vol. 21, pp. 267-359, Springer, 2015.
[17] A. Devgan et al., How to efficiently capture on-chip inductance effects: Introducing a new circuit element K, in *IEEE/ACM Int. Conf. CAD*, 2000.
[18] C. A. Felippa, *Introduction to Finite Element Methods*. Department of Aerospace Engineering Sciences and Center for Aerospace Structures University of Colorado Boulder, Colorado 80309-0429, USA Last updated 2004.
[19] R. W. Freund and P. Feldman, Reduced-order modeling of large passive linear circuits by means of the SYPVL algorithm, in *Proc. of 1996 IEEE/ACM Int. Conf. on Computer-aided Design*, 1997.
[20] S. Funken et al., Efficient implementation of adaptive P1-FEM in Matlab, *Comput. Methods Appl. Math.*, **11** (4) (2011), 460–490.
[21] I. F. Hantila and D. Ioan, Voltage-current relation of circuit elements with field effects, *Rev. Roum. Sci. Tech., Sér. Électrotech. Énerg.*, **39** (3) (1994), 405–416.
[22] H. Tutorial, Adaptive low-order FEM and hp-FEM. http://hpfem.org.
[23] R. Hiptmair, F. Kramer, and J. Ostrowski, A robust Maxwell formulation for all frequencies, *IEEE Trans. Magn.*, **44** (6) (2008), 682–685.
[24] R. Hiptmair and J. Ostrowski, Electromagnetic Port Boundary Conditions: Topological and Variational Perspective, *Technical Report* 2020-27, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2020.
[25] V. Hutson, J. S. Pym, and M. J. Cloud, *Applications of Functional Analysis and Operator Theory*, Elsevier Science, 2005.
[26] H. Igarashi, On the property of the curl-curl matrix in finite element analysis with edge elements, *IEEE Trans. Magn.*, **37** (5) (2001), 3129–3132.

[27] D. Ioan, R. Barbulescu, L. M. Silveira, and G. Ciuprina, Reduced order models of myelinated axonal compartments, *J. Comput. Neurosci.*, **47** (2) (2019), 141–166.
[28] D. Ioan, G. Ciuprina, and S. Kula, Reduced order models for hf interconnect over lossy semiconductor substrate. *IEEE Workshop on Signal Propagation on Interconnects*, pp. 233–236, 2007.
[29] D. Ioan, G. Ciuprina, and M. Radulescu, Algebraic sparsefied partial equivalent electric circuit (ASPEEC), in *Scientific Computing in Electrical Engineering, volume 9*, pp. 45–50, Springer, 2006.
[30] D. Ioan, G. Ciuprina, M. Radulescu, and E. Seebacher, Compact modeling and fast simulation of on-chip interconnect lines, *IEEE Trans. Magn.*, **42** (4) (2006), 547–550.
[31] D. Ioan and I. Munteanu, Missing link rediscovered: The electromagnetic circuit element concept, *JSAEM Stud. Appl. Electromagn. Mech.*, **8** (1999), 302–320.
[32] D. Ioan, I. Munteanu, and C. G. Constantin, The best approximation of the field effects in electric circuit coupled problems, *IEEE Trans. Magn.*, **34** (5) (1998), 3210–3213.
[33] J. D. Jackson, *Electrodynamics*, Wiley, 1975.
[34] H. Ji, A. Devgan, and W. Dai, KSim: a stable and efficient RKC simulator for capturing on-chip inductance effect, in *Proc. of 2001 Asia and South Pacific Design Automation Conference*, 2001.
[35] J. Kanapka, J. Phillips, and J. White, Fast methods for extraction and sparsification of substrate coupling, in *Proc. of the 37th annual Design Automation Conference*, 2000.
[36] T. J. Klemas, *Full-wave Algorithms for Model Order Reduction and Electromagnetic Analysis of Impedance and Scattering*. Diss. MIT, 2005.
[37] M. Kuczmann, *Potential Formulations in Magnetic Applying the Finite Element Method*, Lecture notes Laboratory of Electromagnetic Fields, "Szechenyi Istvan" University, Gyor, Hungary, 2009.
[38] S. Kurz, Some remarks about flux linkage and inductance, *Adv. Radio Sci.*, **2** (2004), 39–44.
[39] A. S. Lup, G. Ciuprina, D. Ioan, A. Duca, A. Nicoloiu, and D. Vasilache, Physics-aware macromodels for MEMS switches, *Compel*, **39** (2) (2020), 497–509.
[40] A. Odabasioglu, M. Celik, and L. T. Pileggi, Prima: Passive reduced-order interconnect macromodeling algorithm, in *Proc. of IEEE/ACM Int. Conf. on Computer-aided design*, 1997.
[41] S. Polstyanko and J.-F. Lee, Adaptive finite element electrostatic solver, *IEEE Trans. Magn.*, **37** (5) (2001), 3120–3124.
[42] R. Radulet et al., Introduction des parametres transitoires dans l'etude des circuits electrique lineaires ayant des elements non filiformes et avec pertes suplimen-taires, *Rev. Roum. Sci. Tech., Sér. Électrotech. Énerg.*, **11** (4) (1966), 565–639.
[43] N. N. Rao, *Fundamentals of Electromagnetics for Electrical and Computer Engineering*, Pearson Prentice Hall, 2009.
[44] F. Rapetti and G. Rousseaux, On quasi-static models hidden in Maxwell's equations, *Appl. Numer. Math.*, (2012), 1–15.
[45] R. F. Remis, Low-frequency model-order reduction of electromagnetic fields without matrix factorization, *IEEE Trans. Microw. Theory Tech.*, **52** (9) (2004), 2298–2304.
[46] E. B. Rudnyi et al., mor4ansys: Generating compact models directly from ansys models, in *Proc. of 2004 Nanotechnology Conf. and Trade Show, Nanotech.*, 2004.
[47] A. E. Ruehli, Inductance calculations in a complex integrated circuit environment, *IBM J. Res. Dev.*, **16** (5) (1972), 470–481.
[48] B. N. Sheehan, ENOR: Model order reduction of RLC circuits using nodal equations for efficient factorization, in *Proc. of IEEE 36th Design Automation Conference*, 1999.
[49] L. M. Silveira et al., A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits, *Comput. Methods Appl. Mech. Eng.*, **169** (3) (1999), 377–389.

[50] R. D. Slone et al., Multipoint Galerkin asymptotic waveform evaluation for model order reduction of frequency domain FEM electromagnetic radiation problems, *IEEE Trans. Antennas Propag.*, **49** (10) (2000), 1504–1513.
[51] P. Sumant et al., Reduced-order models of finite element approximations of electromagnetic devices exhibiting statistical variability, *IEEE Trans. Antennas Propag.*, **60** (1) (2012), 301–309.
[52] A. Timotin, The passive electromagnetic circuit element (in Romanian), *Rev. Roum. Sci. Tech., Sér. Électrotech. Énerg.*, **21** (2) (1971), 347–362.
[53] H. Wu and A. C. Cangellaris, Krylov model order reduction of finite element approximations of electromagnetic devices with frequency-dependent material properties, *Int. J. Numer. Model.*, **20** (5) (2007), 217–235.
[54] H. Yu and L. He, Vector potential equivalent circuit based on PEEC inversion, in *Proc. of 40th annual Design Automation Conference*, 2003.
[55] S. Zaglmayr, *High Order Finite Element Methods for Electromagnetic Field Computation*, Thesis – Linz Univ, 2006.
[56] L. Zhao and A. C. Cangellaris, Reduced-order modeling of electromagnetic field interactions in unbounded domains truncated by perfectly matched layers, *Microw. Opt. Technol. Lett.*, **17** (1) (1998), 62–66.
[57] Y. Zhu and A. C. Cangellaris, Finite element-based model order reduction of electromagnetic devices, *Int. J. Numer. Model.*, **15** (1) (2002), 73–92.
[58] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method: Its Basis and Fundamentals*, Elsevier, 2005.

Masayuki Yano

6 Model reduction in computational aerodynamics

Abstract: Computational aerodynamics has become an indispensable tool in the design and analysis of modern aircraft. However, traditional high-fidelity aerodynamics simulations can be computationally too expensive for scenarios that require responses in real time (e. g., flow control) and/or predictions for many different configurations (e. g., design-space exploration and flight-parameter sweep). The goal of model reduction is to accelerate the solution of unsteady and/or parameterized aerodynamics problems in real-time and/or many-query scenarios. In this chapter, we survey model reduction techniques for linearized and nonlinear aerodynamics problems that have been developed in the past two decades. We discuss essential ingredients of model reduction: stable and efficient projection methods, generation of the reduced basis tailored for the specific solution manifold, and offline-online computational decomposition. We focus on techniques that are designed to address challenges in aerodynamics – nonlinearity, limited stability, limited regularity, and wide range of scales – and have been demonstrated for multidimensional aerodynamic flows. We highlight successful applications of model reduction for large-scale aerodynamics problems.

Keywords: aerodynamics, model reduction, parameterized partial differential equations, (Petrov–)Galerkin projection, reduced basis

MSC 2010: 65N30, 65N35, 35Q30, 35Q35, 76G25

6.1 Introduction

6.1.1 Motivation

With advances in both computational algorithms and hardware, computational fluid dynamics (CFD) has become an indispensable tool in the analysis and design of aerospace vehicles. Today's CFD tools can accurately predict aerodynamics of aircraft in cruise conditions and complement wind-tunnel and flight tests in the aircraft design process; in fact, with the advances in CFD, the number of wings tested in the design of a typical commercial aircraft has decreased by an order of magnitude from the late 1970s to the early 2000s [38].

However, there are computational challenges that still remain out of reach for traditional CFD solvers. To motivate the model reduction work reviewed in this chapter, we name a few "grand challenges" outlined in vision papers [51, 61]. First is high-

Masayuki Yano, University of Toronto, Toronto, Ontario, Canada

https://doi.org/10.1515/9783110499001-006

fidelity aerodynamic database generation; the task requires accurate prediction of aerodynamic forces for the entire range of flight conditions with variations in, e. g., the free stream Mach number and angle of attack. Second is real-time dynamic flight simulation; the task requires aerodynamic or aeroelastic simulation of maneuvering aircraft with the control input specified in real-time. Third is probabilistic design of cooled turbine blades; the task requires accurate characterization of the turbine blade performance under geometric uncertainties due to manufacturing variabilities. These tasks are challenging for traditional CFD solvers because they require (i) predictions for a large number of configurations (i. e., many-query) and/or (ii) real-time predictions of transient phenomena. Completing these tasks, especially in the time scale and computational resources available in typical engineering settings, can be prohibitive with traditional CFD tools. The objective of this chapter is to survey the state of the art in model reduction for many-query and/or real-time problems in aerodynamics.

6.1.2 Real-time and many-query scenarios

We now provide examples of many-query and/or real-time engineering scenarios to which model reduction has been applied. We restrict ourselves to problems in aerodynamics, rather than more general fluid dynamics; we refer to Chapter 9 of this volume for the latter. We do not attempt to provide a comprehensive review; we merely present a few representative works.

S1. *Aerodynamic shape optimization.* One of the many-query applications of model reduction in aerodynamics is shape optimization. Reduced-order models (ROMs) are used to accelerate aerodynamics analysis under parametric geometry changes and to optimize the geometry. The task consists of three steps: parameterization of the geometry; construction of a ROM; and identification of the optimal geometry. ROMs have been used in many-query analysis [6, 69] and inverse design, where the objective is to identify airfoil geometry that yields the prescribed pressure distribution [43, 44, 45, 78].

S2. *Flight-parameter sweep*. Another many-query application of parametric model reduction in aerodynamics is flight-parameter sweep. ROMs are used to accelerate the prediction of aerodynamic forces and moments for a range of flight conditions described in terms of the angle of attack and Mach number [80, 79, 66, 68, 75, 76].

S3. *Aeroelasticity*. One of the classical real-time applications of model reduction in aerodynamics is aeroelasticity. The goal is to analyze the interaction between aerodynamics forces and elastic structure and to detect, for instance, the onset of flutter. Aeroelasticity saw one of the earliest uses of model reduction, with works appearing in at least as early as the mid-1990s for nonparameterized problems [33, 55, 42, 34, 64]. More recently, techniques have been extended to parameterized aeroelasticity problems, with the angle of attack and Mach number as parameters [47, 46, 4, 2, 5]. We also note that there are nonprojection-based

approaches to model reduction, e. g., by the Volterra series; however, given the focus of this handbook, we do not cover these works and refer interested readers to review papers [26, 49, 25].

S4. *Model predictive control.* Another real-time application of model reduction is the control of aerodynamic systems using model predictive control (MPC). Without model reduction, MPC is infeasible for large-scale systems, as it requires real-time solution of optimization problems. ROMs have been incorporated in MPC to control shock location in a supersonic diffuser [36] and to optimize flight path under fuel consumption and aeroelastic constraints [3].

S5. *Uncertainty quantification and state estimation.* Model reduction has also been used for uncertainty quantification, in which the effect of geometry or flow-condition uncertainties are propagated to quantities of interests. ROMs have been used for probabilistic analysis of turbine blades, in which simulation is carried out for thousands of different configurations [17]. Model reduction has also been applied to state estimation, where the aerodynamic flow field is inferred from surface pressure tap data [16, 71].

6.1.3 Scope and outline

We make four disclaimers regarding the scope of this chapter. First, we restrict our presentation to works on aerodynamics rather than more general fluid mechanics, and in particular to works on compressible flow rather than incompressible flow. We refer to Chapter 9 of this volume for more general coverage of model reduction in CFD. Second, given the emphasis of this handbook, we focus on formulation, rather than theoretical, aspects of model reduction. We however note that mathematical theories have played important roles in the development of model reduction approaches for aerodynamics problems; we refer to references provided throughout the chapter for further theoretical discussions. Third, the model reduction literature for aerodynamics problems is vast, with development from both engineering and applied mathematics communities; we attempt to cover representative works but admit the coverage is not exhaustive and there are inevitable omissions. Fourth, we note that (i) precise requirements for an ROM depend on the particular engineering scenario and there is no universal formulation suitable for all scenarios; (ii) even for a given scenario there are many different approaches; and (iii) there are relatively few comparative studies due to the recentness of some of the techniques and the shear cost of performing such studies for large-scale aerodynamics problems. We hence do not attempt to make definitive recommendations and focus on surveying existing approaches, with a hope that the chapter will still serve as a guide to construct an ROM that works for the problem of interest.

This chapter is organized as follows. In Section 6.2, we review full-order discretizations for aerodynamics problems. In Section 6.3, we review model reduction tech-

niques for linearized aerodynamics problems; the linearized problem is relevant for small perturbation analysis, which arises in applications including aeroelasticity, flow control, and uncertainty quantification. In Section 6.4, we review model reduction techniques for nonlinear aerodynamics equations; the full nonlinear analysis is often required for aerodynamic shape optimization and flight-parameter sweep.

6.2 Full-order models

In this section we review full-order models (FOMs) for aerodynamics problems. We consider both the linearized and full nonlinear FOMs; the associated ROMs will be constructed in Sections 6.3 and 6.4, respectively. We describe FOMs in abstract forms to accommodate various governing equations and discretizations under a unified framework.

6.2.1 Conservation laws of aerodynamics

We introduce the general form of aerodynamics partial differential equations (PDEs) considered throughout this chapter. We introduce a P-dimensional parameter domain $\mathcal{P} \in \mathbb{R}^P$, a d-dimensional spatial domain $\Omega \subset \mathbb{R}^d$, the associated boundary $\partial\Omega$, and a time interval $\mathcal{I} \equiv (0, T] \subset \mathbb{R}$. Aerodynamic flow in Ω over $\mathcal{I}$ is modeled by a system of N_c nonlinear conservation laws of the form

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot \left(f^{\text{inv}}(u) + f^{\text{visc}}(u, \nabla u) \right) &= f^{\text{src}}(u, \nabla u) \quad \text{in } \Omega \times \mathcal{I}, \\ b\left(u, n \cdot f^{\text{visc}}(u, \nabla u)\right) &= 0 \quad \text{on } \partial\Omega \times \mathcal{I}, \\ u|_{t=0} &= u^0 \quad \text{in } \Omega, \end{aligned} \tag{6.1}$$

where u is the conservative state, f^{inv} is the inviscid flux function, f^{visc} is the viscous flux function, f^{src} is the source function, b is the boundary condition function, and u^0 is the initial state. While the exact forms of the flux, source, and boundary functions depend on the specific governing equation – the Euler, Navier–Stokes, or Reynolds-averaged Navier–Stokes (RANS) equations – and flow conditions, all conservation laws in aerodynamics can be cast in the general form (6.1). We also emphasize that, although omitted here for brevity, all functions in general depend on the parameter $\boldsymbol{\mu} \in \mathcal{P}$ for parameterized problems and the time $t \in \mathcal{I}$ for unsteady problems.

In many aerodynamics problems, our interest is not necessarily in the entire state field u but in few quantities of interest (i. e., output). Arguably the most common output in aerodynamics are lift and drag, which can be expressed as a surface integral of

the form

$$s \equiv \int_{\Gamma_{\text{body}}} f^{\text{out}}(u, n \cdot f^{\text{visc}}(u, \nabla u); n) ds,$$

where $\Gamma_{\text{body}} \subset \partial\Omega$ is the aerodynamic surface of interest, n denotes the unit vector normal to Γ_{body}, and the function f^{out} maps the surface state and viscous flux to aerodynamic forces.

We make a few remarks about the governing equations in aerodynamics. First, inviscid flows are modeled by the Euler equations, which are purely hyperbolic. Second, viscous flows are modeled by the Navier–Stokes equations which, for Reynolds number relevant to aerodynamics, are convection-dominated. Third, for turbulent flow simulations based on the RANS equations, the Navier–Stokes equations are augmented with additional empirical PDEs that model the turbulence behavior; most turbulence models are highly nonlinear, including the one-equation Spalart–Allmaras (SA) turbulence model [62] used in most of the works reviewed in this chapter. Fourth, nonconservative variables, such as the entropy variables [35], may be used as the working state variables; the entropy variables are of particular interest for stability analysis of Galerkin methods [12] and in particular ROMs [9, 39].

6.2.2 Semi-discrete form

We now consider a full-order approximation of the conservation law (6.1). While there is a number of different discretizations for (6.1), they must provide stability for hyperbolic and convection-dominated PDEs. As a result, most works on model reduction for aerodynamics use one the three full-order discretizations: a finite volume method [65], a stabilized finite element method [15, 37], or a discontinuous Galerkin (DG) method [24, 7]. We refer to the references above for details of the discretizations, and here describe FOMs in an abstract form.

To introduce an FOM, we first introduce a triangulation $\mathcal{T}_h \equiv \{\kappa_1, \ldots, \kappa_{N_e}\}$, where $\{\kappa_i\}_{i=1}^{N_e}$ is a set of N_e nonoverlapping elements such that $\overline{\Omega} = \cup_{\kappa \in \mathcal{T}_h} \overline{\kappa}$ and $\kappa_i \cap \kappa_j = \emptyset$, $i \neq j$. We next introduce an N_h-dimensional approximation space $V_h \subset V$ associated with $\mathcal{T}_h$; the associated dual space is denoted by V_h' with the duality pairing $\langle \cdot, \cdot \rangle : V_h' \times V \to \mathbb{R}$. We then introduce an FOM spatial residual operator $r_h : V_h \times \mathcal{P} \to V_h'$; the particular form of the residual depends on the conservation laws and discretization. A semi-discrete form of our FOM problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_h(t; \boldsymbol{\mu}) \in V_h$, $t \in \mathcal{I}$, such that

$$\frac{\partial u_h(t; \boldsymbol{\mu})}{\partial t} + r_h(u_h(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = 0 \quad \text{in } V_h', \tag{6.2}$$

and $u_h(t = 0; \boldsymbol{\mu}) = \Pi_h u^0(\boldsymbol{\mu})$; here $u^0(\boldsymbol{\mu}) \in V$ is the initial condition, and $\Pi_h : V \to V_h$ is a projection operator from V to V_h. Throughout this chapter, for any Hilbert space W

and the associated dual space W', the statement "$g = 0$ in W'" should be interpreted as $\langle g, w\rangle = 0\ \forall w \in W$. We then introduce an FOM output functional $q_h : V_h \times \mathcal{P} \to \mathbb{R}^{N_o}$, so that the set of N_o outputs is given by

$$s_h(t;\boldsymbol{\mu}) = q_h(u_h(t;\boldsymbol{\mu});\boldsymbol{\mu}). \tag{6.3}$$

We assume that the solution and output to the FOM exists and is unique.

We may also consider an "algebraic form" of the problem, i. e., the form of the problem described by matrices and vectors, which is convenient for the computational implementation of the formulation. To this end, we first introduce a basis $\{\varphi^j\}_{j=1}^{N_h}$ of the space V_h. We next associate any function $v_h \in V_h$ with a generalized coordinate $\mathbf{v}_h \in \mathbb{R}^{N_h}$ by $v_h = \mathbf{v}_h^j \varphi^j$, where $\mathbf{v}_h^j$ denotes the j-th component of $\mathbf{v}_h$ and the summation on the repeated indices is implied. We then introduce algebraic forms of the FOM residual operator $\mathbf{r}_h : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{N_h}$, the output functional $\mathbf{q}_h : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}$, and the mass matrix $\mathbf{M}_h \in \mathbb{R}^{N_h \times N_h}$ given by

$$\begin{aligned}
\mathbf{r}_h(\mathbf{w}_h;\boldsymbol{\mu})_i &\equiv \langle r_h(\mathbf{w}_h^j \varphi^j;\boldsymbol{\mu}), \varphi^i\rangle, \quad i = 1,\dots,N_h,\\
\mathbf{q}_h(\mathbf{w}_h;\boldsymbol{\mu}) &\equiv q_h(\mathbf{w}_h^j \varphi^j;\boldsymbol{\mu}),\\
\mathbf{M}_{h,ij} &\equiv (\varphi^j, \varphi^i)_{L^2(\Omega)}, \quad i,j = 1,\dots,N_h.
\end{aligned}$$

The algebraic form of the FOM problem (6.2) and (6.3) is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_h(t;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, $t \in \mathcal{I}$, such that

$$\mathbf{M}_h \frac{d\mathbf{u}_h(t;\boldsymbol{\mu})}{dt} + \mathbf{r}_h(\mathbf{u}_h(t;\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } \mathbb{R}^{N_h} \tag{6.4}$$

for $\mathbf{u}_h(t=0;\boldsymbol{\mu}) = \mathbf{u}_h^0(\boldsymbol{\mu})$, and then evaluate

$$s_h(t;\boldsymbol{\mu}) = \mathbf{q}_h(\mathbf{u}_h(t;\boldsymbol{\mu});\boldsymbol{\mu}). \tag{6.5}$$

This algebraic form of the problem is equivalent to the operator form (6.2) and (6.3); in particular, $u_h(t;\boldsymbol{\mu}) = \mathbf{u}_h^j(t;\boldsymbol{\mu})\varphi^j$. The solution to (6.4) is typically obtained using a Newton-like method.

We make a few remarks. First, for a typical aerodynamics problem, $P = \mathcal{O}(1-10)$, $N_h = \mathcal{O}(10^5 - 10^7)$, and $N_o = \mathcal{O}(1)$. Second, for steady problems, the time derivative term vanishes and we seek $u(\boldsymbol{\mu}) \in V_h$ such that

$$r_h(u(\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } V_h', \tag{6.6}$$

or, equivalently, $\mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that $\mathbf{r}_h(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) = 0$ in $\mathbb{R}^{N_h}$. Third, for problems with shape deformations, the spatial domain Ω depends on the parameter $\boldsymbol{\mu} \in \mathcal{D}$; we refer to a review [57] for the treatment of parameter-dependent domains by a reference-domain formulation, which provides an equivalent problem in a parameter-independent reference domain. Fourth, while finite volume methods are typically not

presented as a weak formulation (6.2), the form encompasses (in general high-order) finite volume methods, as the methods can be recast as a DG method with an appropriate state reconstruction function; see, e. g., [13]. Fifth, in any event, all FOM discretizations can be expressed in the algebraic form (6.4) and (6.5). Hence, in Sections 6.3 and 6.4, we describe all model reduction techniques using this abstract framework.

6.2.3 Full-discrete form

We now introduce a full-discrete form of the FOM (6.4). We first introduce time instances $0 = t^0 \leq t^1 \leq \cdots \leq t^K = T$, and the associated sequence of functions $\{u_h^k(\boldsymbol{\mu})\}_{k=1}^K = \{\mathbf{u}_h^{k,j}(\boldsymbol{\mu})\varphi^j\}_{k=1}^K$ such that $u_h(t^k;\boldsymbol{\mu}) \approx u_h^k(\boldsymbol{\mu})$, $k = 1,\ldots,K$. We then discretize the semi-discrete equation (6.4) using a multistep or multistage scheme. For instance, if the backward Euler method is used, the full-discrete FOM problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\{\mathbf{u}_h^k(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}\}_{k=1}^K$ such that

$$\mathbf{r}_{h,\Delta t}^k(\mathbf{u}_h^k(\boldsymbol{\mu}); \mathbf{u}_h^{k-1}(\boldsymbol{\mu}); \boldsymbol{\mu}) \equiv \frac{1}{\Delta t}\mathbf{M}_h(\mathbf{u}_h^k(\boldsymbol{\mu}) - \mathbf{u}_h^{k-1}(\boldsymbol{\mu})) + \mathbf{r}_h(\mathbf{u}_h^k(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0$$

for $k = 1,\ldots,K$, and $\mathbf{u}_h^{k=0}(\boldsymbol{\mu}) = \mathbf{u}^0(\boldsymbol{\mu})$. Here, $\mathbf{r}_{h,\Delta t}^k : \mathbb{R}^{N_h} \times \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{N_h}$ is the full-discrete residual operator for the backward Euler method at the time instance k, which depends on the state at the previous time step $\mathbf{u}_h^{k-1}(\boldsymbol{\mu})$. More generally, for a multistep method, the full-discrete residual operator depends on the states at k_{step} previous time instances and takes the form $\mathbf{r}_{h,\Delta t}^k : \mathbb{R}^{N_h} \times \mathbb{R}^{N_h \times k_{\text{step}}} \times \mathcal{P} \to \mathbb{R}^{N_h}$. We assume that an appropriate time marching scheme is chosen such that a sequence of stable solutions exists.

We note that the solution to the steady problem (6.6) is often obtained using a pseudo-transient continuation (PTC) method [41], which solves the unsteady problem using pseudo-time stepping, to improve the convergence of the nonlinear solver. Hence the temporal stability is an important consideration even for steady problems. We refer to [41] for a review of PTC methods.

6.2.4 Linearized equations

While aerodynamic flow is governed by a system of nonlinear conservation laws, as discussed in Section 6.1, time-dependent linearized analysis is also of engineering interest. The goal of linearized analysis is to propagate small input disturbances to output perturbations. Here, the input disturbances may result from small changes in the geometry (e. g., vibrations), boundary conditions (e. g., gust), or initial conditions; our interest is in the associated change in the aerodynamic forces and moments.

Before we proceed, we make one notational change. In the previous section, we introduced the parameter-dependent steady residual operator $r_h : V_h \times \mathcal{P} \to V_h'$; in

this section, to be consistent with literature on linearized aerodynamics analysis, we explicitly separate the parameters subjected to input disturbances from those that are not. Specifically, we introduce a Q-dimensional input space $\mathcal{Q} \subset \mathbb{R}^Q$. We then introduce the steady residual operator $r_h : V_h \times \mathcal{Q} \times \mathcal{P} \to V_h'$, which is a function of the state, input, and parameter. Similarly, we introduce the output operator $q_h : V_h \times \mathcal{Q} \times \mathcal{P} \to \mathbb{R}^{N_o}$.

In linearized analysis, we decompose the solution $u_h \in V_h$ into a base solution $\bar{u}_h$ and perturbation δu_h so that $u_h = \bar{u}_h + \delta u_h$. Similarly, we decompose the input $\nu \in \mathcal{Q}$ into a base input $\bar{\nu}$ and disturbance $\delta\nu$ so that $\nu = \bar{\nu} + \delta\nu$. The perturbation is governed by the following linearized problem: Given $\boldsymbol{\mu} \in \mathcal{P}$ and input $\delta\nu(t) \in \mathcal{Q}$, find $\delta u_h(t; \delta\nu, \boldsymbol{\mu}) \in V_h$, $t \in \mathcal{I}$, such that

$$\frac{\partial \delta u_h(t; \delta\nu, \boldsymbol{\mu})}{\partial t} + J_h(\boldsymbol{\mu})\delta u_h(t; \delta\nu, \boldsymbol{\mu}) + B_h(\boldsymbol{\mu})\delta\nu(t) = 0 \quad \text{in } V_h', \tag{6.7}$$

and $\delta u_h(t = 0; \boldsymbol{\mu}) = \Pi_h \delta u^0(\boldsymbol{\mu})$ for $\delta u^0(\boldsymbol{\mu}) \in V_h$ the initial perturbation. Here, the Jacobian $J_h(\boldsymbol{\mu}) \in \mathcal{L}(V_h, V_h')$ is the Fréchet derivative of $r_h(\cdot, \bar{\nu}; \boldsymbol{\mu}) : V_h \to V_h'$ at $\bar{u}_h$, and the operator $B_h(\boldsymbol{\mu}) \in \mathcal{L}(Q, V_h')$ is the Fréchet derivative of $r_h(\bar{u}_h, \cdot; \boldsymbol{\mu}) : \mathcal{Q} \to V_h'$ at $\bar{\nu}$. Given the perturbed state $\delta u_h(t; \delta\nu, \boldsymbol{\mu}) \in V_h$, we evaluate the associated output perturbation

$$\delta s_h(t; \delta\nu, \boldsymbol{\mu}) = g_h(\boldsymbol{\mu})\delta u_h(t; \delta\nu, \boldsymbol{\mu}),$$

where $g_h(\boldsymbol{\mu}) \in \mathcal{L}(V_h, \mathbb{R}^{N_o})$ is the Fréchet derivative of $q_h(\cdot, \bar{\nu}; \boldsymbol{\mu})$ at $\bar{u}_h$. The goal of the linearized aerodynamics analysis is to map the disturbances in the input $\delta\nu(t) \in \mathcal{Q}$ to the perturbations in the output $\delta s_h(t; \delta\nu, \boldsymbol{\mu}) \in \mathbb{R}^{N_o}$ for any parameter value $\boldsymbol{\mu} \in \mathcal{P}$. In aerodynamics, the linearization state $\bar{u} \in \mathcal{V}_h$ is often the solution to the steady-state nonlinear problem (6.6); i. e., $r_h(\bar{u}_h; \boldsymbol{\mu}) = 0$ in V_h'.

The linearized equations can also be expressed in an algebraic form. To this end, we introduce the Jacobian matrix $\mathbf{J}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$, input matrix $\mathbf{B}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times Q}$, and output gradient vector $\mathbf{g}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_o \times N_h}$ such that

$$\begin{aligned}
\mathbf{J}_h(\boldsymbol{\mu})_{ij} &= \left\langle J_h(\bar{u}_h; \boldsymbol{\mu})\varphi^j, \varphi^i \right\rangle, \quad i, j = 1, \ldots, N_h, \\
\mathbf{B}_h(\boldsymbol{\mu})_{ij} &= \left\langle B_h(\bar{u}_h; \boldsymbol{\mu})e^j, \varphi^i \right\rangle, \quad i = 1, \ldots, N_h,\ j = 1, \ldots, Q, \\
\mathbf{g}_h(\boldsymbol{\mu})_{ij} &= \left\langle g_h(\bar{u}_h; \boldsymbol{\mu})\varphi^j, e^i \right\rangle \quad i = 1, \ldots, N_o,\ j = 1, \ldots, N_h,
\end{aligned}$$

where e^j is the unit vector with the j-th entry equal to 1. The algebraic form of the linearized problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$ and input $\delta\nu(t) \in \mathcal{Q}$, find $\delta\mathbf{u}_h(t; \delta\nu, \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, $t \in \mathcal{I}$, such that

$$\mathbf{M}_h \frac{d\delta\mathbf{u}_h(t; \delta\nu, \boldsymbol{\mu})}{dt} + \mathbf{J}_h(\boldsymbol{\mu})\delta\mathbf{u}_h(t; \delta\nu, \boldsymbol{\mu}) + \mathbf{B}_h(\boldsymbol{\mu})\delta\nu(t) = 0 \quad \text{in } \mathbb{R}^{N_h}, \tag{6.8}$$

and evaluate the output

$$\delta s_h(t; \delta\nu, \boldsymbol{\mu}) = \mathbf{g}_h(\boldsymbol{\mu})\delta\mathbf{u}_h(t; \delta\nu, \boldsymbol{\mu}).$$

We note that, for a fixed parameter $\boldsymbol{\mu} \in \mathcal{P}$, the problem is in the standard linear time-invariant form. The application of a time marching scheme yields a full-discrete form of the linearized equations whose solution $\{\delta\mathbf{u}_h^k\}_{k=1}^K$ satisfies $\delta\mathbf{u}_h^k \approx \delta\mathbf{u}_h(t^k)$, $k = 1,\dots,K$, analogously to the discussion for the nonlinear FOM in Section 6.2.3.

6.3 Model reduction for linearized aerodynamics

In this section we discuss model reduction of linearized aerodynamics problems. As discussed in Section 6.2.4, linearized (i. e., small-perturbation) analysis of unsteady aerodynamics provides significant insights in many engineering scenarios. More pragmatically, model reduction of linearized PDEs requires fewer ingredients than that of nonlinear PDEs, and hence we introduce common ingredients in the linearized context.

6.3.1 Galerkin method

We now consider reduced-order approximations of the FOM (6.7) (or equivalently (6.8)). To this end, we introduce a sequence of reduced basis spaces $V_{N=1} \subset \cdots \subset V_{N=N_{\max}}$, each of which is a subset of $\mathcal{V}_h$; for a typical aerodynamics ROM, $N_{\max} = \mathcal{O}(10\text{–}100)$, which is significantly smaller than $N_h = \mathcal{O}(10^5\text{–}10^7)$. We then introduce the associated hierarchical reduced basis $\{\zeta^n \in \mathcal{V}_h\}_{n=1}^N$ such that $V_N = \text{span}\{\zeta^n\}_{n=1}^N$, $N = 1,\dots,N_{\max}$. We may also express the reduced basis in an algebraic form $\{\boldsymbol{\zeta}^n \in \mathbb{R}^{N_h}\}_{n=1}^N$ such that $\zeta^n = \boldsymbol{\zeta}^{n,j}\varphi^j$, $N = 1,\dots,N_{\max}$; we introduce the associated reduced basis matrix $\mathbf{Z}_N = (\boldsymbol{\zeta}^1,\dots,\boldsymbol{\zeta}^N) \in \mathbb{R}^{N_h \times N}$. We will discuss various methods to construct the reduced basis in Section 6.3.2; for now we assume the basis is given.

Given a reduced basis space V_N, the semi-discrete form of the Galerkin-ROM problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$ and $\delta v(t) \in \mathcal{Q}$, find $\delta u_N(t;\delta v,\boldsymbol{\mu}) \in V_N$, $t \in \mathcal{I}$, such that

$$\frac{\partial \delta u_h(t;\delta v,\boldsymbol{\mu})}{\partial t} + J_h(\boldsymbol{\mu})\delta u_N(t;\delta v,\boldsymbol{\mu}) + B_h(\boldsymbol{\mu})\delta v(t) = 0 \quad \text{in } V_N', \tag{6.9}$$

and $\delta u_N(t=0;\boldsymbol{\mu}) = \Pi_N u^0(\boldsymbol{\mu})$, where $\Pi_N : V \to V_N$ is a projection operator from V to V_N. Again, for $g \in V_h'$, the statement $g = 0$ in V_N' should be interpreted as $\langle g, v\rangle = 0$ $\forall v \in V_N$. We then evaluate the output perturbation $\delta s_h(t;\delta v,\boldsymbol{\mu}) = g_h(\boldsymbol{\mu})\delta u_N(t;\delta v,\boldsymbol{\mu})$. The comparison of the FOM problem (6.7) and the Galerkin-ROM problem (6.9) shows that the latter results from the restriction of the test and trail spaces to the reduced space $V_N \subset V_h$.

The Galerkin-ROM problem (6.9) can also be expressed in an algebraic (or matrix-vector) form. To this end, we associate any function $v_N \in V_N$ with a generalized coordinate $\mathbf{v}_N \in \mathbb{R}^N$ by $v_N = \mathbf{v}_N^j \zeta^j$; we may also express the full-order generalized coordinate

of $v_N \in V_N$ as $\mathbf{v}_h = \mathbf{v}_N^j \boldsymbol{\zeta}^j = \mathbf{Z}_N \mathbf{v}_N \in \mathbb{R}^{N_h}$. Given the reduced basis, we define the ROM operators

$$\begin{aligned}
\mathbf{M}_N &\equiv \mathbf{Z}_N^T \mathbf{M}_h \mathbf{Z}_N = \left((\zeta^j, \zeta^i)_{L^2(\Omega)} \right)_{i,j=1}^N \in \mathbb{R}^{N \times N}, \\
\mathbf{J}_N(\boldsymbol{\mu}) &\equiv \mathbf{Z}_N^T \mathbf{J}_h(\boldsymbol{\mu}) \mathbf{Z}_N = \left(\langle J_h(\boldsymbol{\mu}) \zeta^j, \zeta^i \rangle \right)_{i,j=1}^N \in \mathbb{R}^{N \times N}, \\
\mathbf{B}_N(\boldsymbol{\mu}) &\equiv \mathbf{Z}_N^T \mathbf{J}_h(\boldsymbol{\mu}) = \left(\langle B_h(\boldsymbol{\mu}) e^j, \zeta^i \rangle \right)_{i=1,j=1}^{N,Q} \in \mathbb{R}^{N \times Q}, \\
\mathbf{g}_N(\boldsymbol{\mu}) &\equiv \mathbf{g}_h(\boldsymbol{\mu}) \mathbf{Z}_N = \left(\langle g_h(\boldsymbol{\mu}) \zeta^j, e^i \rangle \right)_{j=1}^N \in \mathbb{R}^{N_o \times N}.
\end{aligned} \tag{6.10}$$

The algebraic form of the linearized problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$ and $\delta v(t; \boldsymbol{\mu}) \in \mathcal{Q}$, find $\delta \mathbf{u}_N(t; \delta v, \boldsymbol{\mu}) \in \mathbb{R}^N$, $t \in \mathcal{I}$, such that

$$\mathbf{M}_N \frac{d \delta \mathbf{u}_N(t; \delta v, \boldsymbol{\mu})}{dt} + \mathbf{J}_N(\boldsymbol{\mu}) \delta \mathbf{u}_N(t; \boldsymbol{\mu}) + \mathbf{B}_N(\boldsymbol{\mu}) \delta v = 0 \quad \text{in } \mathbb{R}^N, \tag{6.11}$$

and evaluate the output $\delta s_N(t; \delta v, \boldsymbol{\mu}) = \mathbf{g}_N(\mathbf{u}_N(\boldsymbol{\mu}); \delta v, \boldsymbol{\mu}) \delta \mathbf{u}_N(t; \boldsymbol{\mu})$. Again, the operator form (6.9) and the algebraic form (6.11) are equivalent and $\delta u_N(t; \boldsymbol{\mu}) = \delta \mathbf{u}_N^j(t; \boldsymbol{\mu}) \zeta^j$. We note that the ROM operators (6.10) are precomputed in the construction stage, so that the ROM (6.11) can be solved in $\mathcal{O}(N^\bullet)$ operations for the exponent $\bullet$ between 1 and 3. In particular the cost to solve the ROM (6.11) is independent of N_h; we recall that $N = \mathcal{O}(10\text{–}100)$ and $N_h = \mathcal{O}(10^5\text{–}10^7)$ for a typical aerodynamics problem. We discuss this offline-online computational decomposition in Section 6.3.4.

6.3.2 Reduced basis for nonparameterized linearized problems

The efficacy of the Galerkin-ROM (6.9) (or (6.11)) depends on the choice of the reduced basis. We now review techniques to identify an effective reduced basis $\{\zeta^j\}_{j=1}^N$ (or reduced basis matrix $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N}$). For practical and historical reasons, we first present procedures for nonparameterized (or fixed-parameter) problems; the model reduction of time-varying but fixed-parameter aerodynamics problems enables fast simulation of complex flows, which is essential for, for instance, MPC. As the problems are nonparameterized, we suppress the argument $\boldsymbol{\mu}$ for all operators throughout this section. In addition, as our primary goal is to provide recipes for implementation, rather than to discuss theory, we present algorithms in algebraic forms.

6.3.2.1 Eigenmodes

A classical approach to identify a reduced basis for linearized aerodynamics problems is eigenanalysis. The approach, first introduced by Hall [33], is as follows:

1. Solve the generalized eigenproblem: Find the eigenvector $\boldsymbol{\zeta}^k \in \mathbb{R}^{N_h}$ and the associated eigenvalue $\lambda^k \in \mathbb{C}$ such that

$$\mathbf{J}_h \boldsymbol{\zeta}^k = \lambda^k \mathbf{M}_h \boldsymbol{\zeta}^k \quad \text{in } \mathbb{R}^{N_h};$$

   without loss of generality, sort the eigenpairs such that $|\lambda^1| \geq \cdots \geq |\lambda^{N_h}|$.
2. Construct the reduced basis matrix $\mathbf{Z}_N = (\boldsymbol{\zeta}^1, \ldots, \boldsymbol{\zeta}^N) \in \mathbb{R}^{N_h \times N}$.

While historically important, eigenanalysis has a major limitation: The reduced basis is based solely on the Jacobian $\mathbf{J}_h$ and does not account for the system input $\mathbf{B}_h$ or output $\mathbf{g}_h$. Hence, the number of eigenmodes N required to achieve a given solution or output accuracy is typically greater than empirical approaches based on proper orthogonal decomposition (POD).

6.3.2.2 Time-domain POD

To address limitations of eigenmodes discussed in Section 6.3.2.1, Romanowski [55] proposes a (time-domain) POD approach for linearized Euler equations. We here present the method of snapshots [60] to efficiently compute a POD basis for large-scale problems in aerodynamics:

1. Choose L time-dependent training inputs $\{\{\delta v^l(t)\}_{t\in\mathcal{I}}\}_{l=1}^L$, where l is the training input index.
2. Solve the full-discrete form of the linearized FOM (6.7) for the training inputs $\{\delta v^l\}_{l=1}^L$ and for K time steps $\{t^k\}_{k=1}^K$ to construct a snapshot matrix $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$, whose columns are $\delta \mathbf{u}_h^k(\delta v^l) \approx \delta \mathbf{u}_h(t^k; \delta v^l)$ for $k = 1, \ldots, K$, $l = 1, \ldots, L$, and $N_s \equiv KL$.
3. Construct the correlation matrix $\mathbf{A} = \mathbf{S}^T \mathbf{X}_h \mathbf{S}$ in $\mathbb{R}^{N_s \times N_s}$. Here, $\mathbf{X}_h \in \mathbb{R}^{N_h \times N_h}$ such that $\mathbf{X}_{h,ij} = (\varphi^j, \varphi^i)_{X_h}$ is associated with an appropriate inner product; a common choice is the $L^2(\Omega)$-inner product.
4. Solve the eigenproblem: Find $(\mathbf{v}^k, \lambda^k) \in \mathbb{R}^{N_s} \times \mathbb{R}$ such that

$$\mathbf{A}\mathbf{v}^k = \lambda^k \mathbf{v}^k \quad \text{in } \mathbb{R}^{N_s};$$

   without loss of generality, sort the eigenpairs such that $|\lambda^1| \geq \cdots \geq |\lambda^{N_s}|$.
5. Set the reduced basis matrix $\mathbf{Z}_N = (\boldsymbol{\zeta}^1, \ldots, \boldsymbol{\zeta}^N) \in \mathbb{R}^{N_h \times N}$, where

$$\boldsymbol{\zeta}^k = \lambda_k^{-1/2} \mathbf{S}\mathbf{v}^k, \quad k = 1, \ldots, N.$$

The resulting basis $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N}$ is orthogonal with respect to the $\mathbf{X}_h$ inner product; i. e., $\mathbf{Z}_N^T \mathbf{X}_h \mathbf{Z}_N = I_N$. In addition, $\mathbf{Z}_N$ minimizes the X_h-projection error for the snapshots; i. e., $\mathbf{Z}_N = \arg\min_{\mathbf{W}_N \in \mathbb{R}^{N_h \times N}} \|\mathbf{S} - \mathbf{W}_N \mathbf{W}_N^T \mathbf{X}_h \mathbf{S}\|_{\mathbf{X}_h}$. In this sense, the POD basis is optimal for the approximation of the state $\delta u_h(t; \delta v)$ associated with the particular system input δv; however, the system output s_h is not accounted for in the POD method.

6.3.2.3 Frequency-domain POD

A variant of the time-domain POD approach above is the frequency-domain POD approach proposed by Kim [42] and Hall et al. [34]. As the name suggests, this approach takes advantage of the linearity of the problem (6.8) and computes snapshots in the frequency domain. Namely, we consider time-harmonic disturbances of the form $\delta v(t) = \delta\hat{v}e^{j\omega t}$ of a frequency $\omega \in \mathbb{R}$ so that the associated time-harmonic perturbations are of the form $\delta\mathbf{u}_h(t;\delta v,\boldsymbol{\mu}) = \delta\hat{\mathbf{u}}_h(\delta\hat{v})e^{j\omega t}$ for $\delta\hat{\mathbf{u}}_h(\delta\hat{v}) = (j\omega\mathbf{M}_h + \mathbf{J}_h)^{-1}\mathbf{B}_h\delta\hat{v}$, where $j \equiv \sqrt{-1}$. The frequency-domain POD approach replaces the first two steps of the time-domain POD approach in Section 6.3.2.2 with the following:

1'. Choose L training inputs $\{\delta\hat{v}^l \in \mathbb{R}^Q\}_{l=1}^L$ and K training frequencies $\{\omega_k \in \mathbb{R}\}_{k=1}^K$.

2'. Solve the frequency-domain equation

$$(j\omega_k\mathbf{M}_h + \mathbf{J}_h)\delta\hat{\mathbf{u}}^k(\delta\hat{v}^l) = \mathbf{B}_h\delta\hat{v}^l \tag{6.12}$$

for $\{\delta\hat{v}^l\}_{l=1}^L$ and $\{\omega_k\}_{k=1}^K$ to construct a snapshot matrix $\mathbf{S} \in \mathbb{R}^{N_h\times N_s}$, whose columns are the real and imaginary parts of the frequency-domain perturbation, $\mathbb{R}(\delta\hat{\mathbf{u}}^k(\delta v^l))$ and $\mathbb{I}(\delta\hat{\mathbf{u}}^k(\delta v^l))$, for $k = 1,\dots,K$, $l = 1,\dots,L$, and $N_s \equiv 2KL$.

The training input modes and frequencies can be chosen based on known characteristics of input disturbances; e. g., for aeroelasticity problems, the modes and frequencies may be chosen to coincide with the resonance modes of the structure. For linearized aerodynamics problems, frequency-domain POD is often more efficient than time-domain POD and hence is preferred; the approach has been successfully applied to the linearized Euler equations in works including [42, 34, 64, 47, 46, 4, 2]. We however make two cautionary remarks: First, implementation must support complex arithmetic; second, just like time-domain POD, while the POD basis is in some sense optimized for the solution field $\delta u_h(t;\delta v) \in V_h$, it is not specialized for the particular system output s_h.

6.3.2.4 Balanced POD

The time- and frequency-domain POD approaches construct a reduced space V_N which is well suited for the approximation of the entire state $\delta u_h(t;\delta v) \in V_h$; however, in aerodynamics, we are often not interested in the entire state but rather only in few outputs (i. e., quantities of interest). In these cases, we can construct a more efficient ROM using the balanced POD (BPOD) method proposed by Willcox and Peraire [73], which approximates balanced truncation [52] for large-scale problems. The key to BPOD is (i) to realize that both the input and output play equally important roles in characterizing the input-output relationship and (ii) to incorporate the dual problem to account for the choice of the output. The dual problem for the linearized aerodynamics problem (6.8) with a single output ($N_o = 1$) is as follows: Given $\delta v(t) \in \mathcal{Q}$, find

$\mathbf{z}_h(t;\delta\nu) \in \mathbb{R}^{N_h}$, $t \in \mathcal{I}$, such that

$$-\frac{d\mathbf{z}_h(t;\delta\nu)}{dt} + \mathbf{J}_h^T\mathbf{z}(t;\delta\nu) + \mathbf{g}^T = 0 \quad \text{in } \mathbb{R}^{N_h}, \tag{6.13}$$

and then evaluate the output

$$\delta s_h(t;\delta\nu) = \delta\nu\mathbf{B}_h^T\mathbf{z}_h(t;\delta\nu).$$

The associated frequency-domain problem seeks $\delta\hat{\mathbf{z}}(\delta\nu) = (-j\omega\mathbf{M}_h + \mathbf{J}_h^T)^{-1}\mathbf{g}_h^T$. The BPOD procedure based on frequency-domain sampling for $N_o = 1$ is as follows:

1. Choose L training inputs $\{\delta\hat{\nu}^l \in \mathbb{R}^Q\}_{l=1}^L$ and K training frequencies $\{\omega_k \in \mathbb{R}\}_{k=1}^K$.
2. Solve the frequency-domain problem (6.12) to collect N_s primal snapshots, and then obtain the POD mode matrix $\mathbf{Z}_p^{\text{pr}} \in \mathbb{R}^{N_h\times p}$ and eigenvalue matrix $\mathbf{\Lambda}_p^{\text{pr}} \in \mathbb{R}^{p\times p}$ for the $p \geq N$ largest eigenvalues.
3. Solve the frequency-domain dual problem (6.13) to collect N_s adjoint snapshots, and then obtain the POD mode matrix $\mathbf{Z}_p^{\text{du}} \in \mathbb{R}^{N_h\times p}$ and eigenvalue matrix $\mathbf{\Lambda}_p^{\text{du}} \in \mathbb{R}^{p\times p}$ for the p largest eigenvalues.
4. Compute the eigenvectors $\mathbf{Z}_N \in \mathbb{R}^{N_h\times N}$ associated with the N largest eigenvalues of the matrix $(\mathbf{Z}_p^{\text{pr}}\mathbf{\Lambda}_p^{\text{pr}}\mathbf{Z}_p^{\text{pr}\,T})(\mathbf{Z}_p^{\text{du}}\mathbf{\Lambda}_p^{\text{du}}\mathbf{Z}_p^{\text{du}\,T})$ using a Krylov subspace method. (Note that the matrix is never explicitly formed.)

The BPOD method produces a reduced basis optimized for the input-output mapping problem and enables goal-oriented reduction of linearized aerodynamics problems [73]; depending on the output, BPOD significantly reduces the dimension of the reduced space required to achieve a given output tolerance compared to the standard POD, as demonstrated for a two-dimensional plunging airfoil [73]. A variant of BPOD modified for a problem with a large number of outputs is developed by Rowley in [56].

6.3.2.5 Other goal-oriented methods

We survey a few other goal-oriented methods to generate reduced bases; we again restrict ourselves to techniques that have been demonstrated for aerodynamics problems. In [74], Willcox et al. propose an Arnoldi-based method, which identifies a reduced basis by matching moments of the FOM input-output transfer function, and apply it to aeroelastic analysis of a transonic turbine cascade with unsteady blade motions. In [72], Willcox and Megretski propose a method which identifies a reduced basis by computing the Fourier expansion of the discrete-frequency transfer function, and apply it to analysis of a supersonic diffuser. In [18], Bui-Thanh et al. propose a more general approach to goal-oriented model reduction that identifies a reduced basis as a solution of a constrained optimization problem and apply it to analysis of a subsonic turbine blade. All three methods are goal-oriented in the sense that they consider both system inputs and outputs to identify an effective reduced basis.

6.3.3 Reduced basis for parameterized linearized problems

We have so far discussed the construction of reduced bases for nonparameterized problems or, equivalently, for one fixed parameter. For parameterized problems, in general a reduced basis constructed for one parameter value does not provide a good approximation for another parameter value, as the associated dynamics can be very different; see, for example, a study for parameterized turbine blades by Epureanu [28]. We here discuss a few different strategies to construct reduced bases for parameterized problems.

6.3.3.1 Global POD

One approach to construct a reduced basis for parameterized problems is to prepare a "global" or "composite" POD basis, which has been trained for a range of parameters, as proposed for aerodynamics problems by Schmit, Taylor, and Glauser [59, 63]. In this approach, we first introduce a training parameter set $\Xi_{N_t} \equiv \{\boldsymbol{\mu}^m\}_{m=1}^{N_t}$, collect the snapshots for all parameter values, and then apply POD to the snapshots. The global POD approach for parameterized problem replaces the first two steps of the time-domain POD approach in Section 6.3.2.2 with the following:

1'. Choose N_t training parameters $\{\boldsymbol{\mu}^n\}_{n=1}^{N_t}$ and L training inputs $\{\delta\nu\}_{l=1}^{L}$.
2'. Solve the full-discrete form of the linearized FOM (6.7) for the training parameters $\{\boldsymbol{\mu}^m\}_{m=1}^{N_t}$, training inputs $\{\delta\nu^l\}_{l=1}^{L}$, and time steps $\{t^k\}_{k=1}^{K}$ to construct a snapshot matrix $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$, whose columns are $\delta\mathbf{u}_h^k(\delta\nu^l;\boldsymbol{\mu}^m) \approx \delta\mathbf{u}_h(t^k;\delta\nu^l)$ for $k = 1,\dots,K$, $l = 1,\dots,L$, $m = 1,\dots,N_t$, and $N_s \equiv KLN_t$.

The global POD approximation works well for problems with a relatively small parameter dimension and extent; however, the method may suffer from two issues if the problem exhibits significant parametric variations. First, the FOM may need to be solved for a large number of training parameters, which results in a high training cost. Second, a large number of POD modes may be required to accurately capture the dynamics. (More precisely, if the Kolmogorov N-width of the parametric manifold $\{u_h(t;\delta\nu,\boldsymbol{\mu})\}_{t\in\mathcal{I},\delta\nu\in\mathcal{Q},\boldsymbol{\mu}\in\mathcal{P}}$ is large, then a large number of modes is required to achieve sufficient accuracy.)

6.3.3.2 The (weak) greedy algorithm

To address the potentially high training cost associated with the global POD, the (weak) greedy algorithm has been developed [67, 57]. The greedy algorithm successively identifies a reduced basis $\{\boldsymbol{\zeta}^j\}_{j=1}^{N}$ based on the behavior of a rapidly computable error estimate $\eta_N(\boldsymbol{\mu})$. The algorithm takes as the input the training parameter

set $\Xi_t \subset \mathcal{D}$ which reasonably covers the domain. Then, in the N-th iteration, given $\mathbf{Z}_{N-1} \in \mathbb{R}^{N_h \times (N-1)}$ the algorithm proceeds as follows:

1. Find the parameter with the largest error estimate: $\boldsymbol{\mu}^N = \arg\max_{\boldsymbol{\mu} \in \Xi_t} \eta_{N-1}(\boldsymbol{\mu})$.
2. Solve the FOM for $\boldsymbol{\mu}^N$ to obtain $\mathbf{u}_h(\boldsymbol{\mu}^N) \in \mathbb{R}^{N_h}$.
3. Augment the reduced basis with the new snapshot: $\mathbf{Z}_N = (\mathbf{Z}_{N-1}, \mathbf{u}_h(\boldsymbol{\mu}^N))$; re-orthonormalize $\mathbf{Z}_N$ using Gram–Schmidt.

The steps are repeated until the user-prescribed error tolerance is met for all $\boldsymbol{\mu} \in \Xi_t$. For unsteady problems, Step 3 incorporates an additional reduction technique (e. g., POD) to compress the multiple temporal snapshots associated with a single unsteady solve; this approach, called POD-greedy algorithm, was proposed and analyzed in [32] and its variant is applied to probabilistic analysis of turbine cascades in [17].

The weak greedy algorithm has two advantages over POD. First, it requires only N FOM solutions compared to $N_t \gg N$ solutions for global POD; hence it reduces the training cost, and a larger Ξ_t can be used for more exhaustive training. Second, in the presence of a goal-oriented error estimate, the ROM trained will meet the error threshold for the engineering quantities of interest at least for $\boldsymbol{\mu} \in \Xi_t$. However, one major limitation of the weak greedy algorithm is that it requires a rapidly computable error estimate; due to the difficulty of constructing such an error estimate for hyperbolic and convection-dominated problems in aerodynamics, the greedy algorithm has seen somewhat limited use in the field. In addition, while the training cost is reduced relative to global POD, the resulting ROM may still require a large N if the problem exhibits significant parametric variations. We refer to a review paper [57] for more detailed description of the weak greedy algorithm.

6.3.3.3 Parameter-domain decomposition

One approach to reduce the ROM size for problems that exhibit a large parameter extent is to decompose the parameter domain $\mathcal{P}$ (or time interval $\mathcal{I}$) into smaller subdomains to limit the parameter extent, which in turn controls the reducibility (i. e., the Kolmogorov N-width) of the parametric manifold. Namely, we first subdivide $\mathcal{P}$ into $N_\mathcal{P}$ subdomains $\{\mathcal{P}^n\}_{n=1}^{N_\mathcal{P}}$ so that $\bigcup_{n=1}^{N_\mathcal{P}} \overline{\mathcal{P}}^n = \overline{\mathcal{P}}$. We then construct a set of $N_\mathcal{P}$ reduced bases $\{\mathbf{Z}^n\}_{n=1}^{N_\mathcal{P}}$ for the parametric manifolds $\{\{\mathbf{u}_h(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{P}^n}\}_{n=1}^{N_\mathcal{P}}$. To make an ROM prediction for a given parameter $\boldsymbol{\mu} \in \mathcal{P}$, we identify the subdomain $\mathcal{P}^n$ such that $\boldsymbol{\mu} \in \mathcal{P}^n$ and then invoke the ROM.

One of the earliest applications of the parameter-domain decomposition approach in aerodynamics is Annonen et al. [6]; the so-called multi-POD approach considers multiple reduced bases associated with different shape deformations. Washabaugh et al. [68] also employ the approach for Mach number sweep of a full aircraft configuration. Some versions of the reduced space interpolation methods [2], which is discussed in Section 6.3.3.4, also incorporates the idea to work with

a database of reduced spaces. We also refer to [27] for detailed analyses of parameter-domain decomposition approaches.

6.3.3.4 Reduced-space interpolation based on Grassmann manifold

Another approach to reduce the ROM size for problems that exhibit a large parameter extent is to "interpolate" a set of reduced spaces computed for several parameter values to construct a new reduced space for the particular parameter value. One simple idea is to interpolate each basis vector $\boldsymbol{\zeta}^j$ as a function of $\boldsymbol{\mu} \in \mathcal{P}$; however, this approach, which works with the vectors and not the space, is shown to work poorly for aeroelasticity problems [48]. To address the problem, Lieu et al. [48, 47, 46] propose the so-called subspace-angle interpolation method to interpolate any two reduced spaces. Subsequently, Amsallem et al. [4, 2] propose a more general approach to interpolate an arbitrary number of reduced spaces associated with $\{\mathbf{Z}_N^i\}_{i=1}^{N_Z}$ to construct a new reduced basis $\mathbf{Z}_N$. The approach builds on the observation that the reduced space V_N spanned by a reduced basis $\mathbf{Z}_N$ is an element of the Grassmann manifold $G(N, N_h)$. To interpolate reduced spaces, the approach (i) invokes a logarithmic map to map reduced spaces onto a tangent space, (ii) performs standard interpolation in the tangent space, and (iii) invokes an exponential map to map back the logarithmic representation of the interpolated basis to identify $\mathbf{Z}_N$. Here we outline the algorithm:

1. Choose parameter values $\{\boldsymbol{\mu}^i\}_{i=0}^{N_Z}$ and construct the associated reduced bases $\{\mathbf{Z}_N^i\}_{i=0}^{N_Z}$; $i = 0$ is the reference point.
2. Compute the logarithms $\{\boldsymbol{\Gamma}_i \in \mathbb{R}^{N_h \times N}\}_{i=1}^{N_Z}$ given by

$$
\begin{aligned}
(\mathbf{I} - \mathbf{Z}_N^0 \mathbf{Z}_N^{0\,T}) \mathbf{Z}_N^i (\mathbf{Z}_N^{0\,T} \mathbf{Z}_N^i)^{-1} &= \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i \quad \text{in } \mathbb{R}^{N_h \times N}, \\
\boldsymbol{\Gamma}_i &= \mathbf{U}_i \tan^{-1}(\boldsymbol{\Sigma}_i) \mathbf{V}_i^T \quad \text{in } \mathbb{R}^{N_h \times N},
\end{aligned}
$$

   where the right-hand side of the first step is the thin singular value decomposition of the matrix in the left-hand side.
3. Given $\boldsymbol{\mu} \in \mathcal{P}$, interpolate each entry of the parameter-logarithm-matrix pairs $(\boldsymbol{\mu}^i, \boldsymbol{\Gamma}^i)_{i=1}^{N_Z}$ using a multivariate interpolation scheme for $\mathbb{R}^P$ to find $\boldsymbol{\Gamma} \in \mathbb{R}^{N_h \times N}$ associated with $\boldsymbol{\mu} \in \mathcal{P}$.
4. Compute the exponential map of the logarithm $\boldsymbol{\Gamma} \in \mathbb{R}^{N_h \times N}$ given by

$$
\begin{aligned}
\boldsymbol{\Gamma} &= \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T, \\
\mathbf{Z}_N &= \mathbf{Z}_N^0 \mathbf{V} \cos(\boldsymbol{\Sigma}) + \mathbf{U} \sin(\boldsymbol{\Sigma}).
\end{aligned}
$$

This interpolation method on the Grassmann manifold can be thought of as a generalization of the subspace-angle interpolation method [48, 47, 46]; the two methods are

equivalent when $N_Z = 2$ reduced bases are used for interpolation, but the former generalizes to an arbitrary number of reduced bases [4]. For problems with a large parameter extent, the reduced space interpolation methods can also be combined with the parameter-domain decomposition method discussed in Section 6.3.3.3; in this case, the interpolation is performed on a subset of all available reduced bases [2]. The reduced basis interpolation methods have been demonstrated for parameterized aeroelastic analysis of full aircraft configurations [4, 2] as discussed further in Section 6.3.6.

6.3.4 Offline-online computational decomposition

As briefly discussed in Section 6.3.1, model reduction achieves computational speedup through offline-online computational decomposition. The offline stage is expensive but is performed only once. The online stage is cheap, and it is invoked in real-time for many different inputs and/or parameters. To describe offline-online computational decomposition for linearized aerodynamics problems, we break down the model reduction procedure into three steps:

1. Collect the FOM snapshots and construct a reduced basis $\mathbf{Z}_N$ (or reduced bases $\{\mathbf{Z}_N^n\}$) using a method described in Section 6.3.2 or 6.3.3.
2. Construct the ROM operators by projecting the FOM operators onto the reduced basis $\mathbf{Z}_N$ according to (6.10).
3. Given the input $\delta v(t) \in \mathcal{Q}$, $t \in \mathcal{I}$, solve the ROM problem (6.11).

In general, Step 1 is the most expensive stage, as it requires time- or frequency-domain solutions of the FOM for a number of different control inputs and/or parameters. Step 2 also requires access to the FOM, and hence does require $\mathcal{O}(N_h)$ operations; however, this step is much cheaper than Step 1, as performing the projection (6.10) is much cheaper than solving the FOM (6.11). Step 3, which works exclusively with the ROM, requires $\mathcal{O}(N^\star)$ operations; since $N_h = \mathcal{O}(10^5 - 10^7)$ and $N = \mathcal{O}(10 - 100)$ for a typical aerodynamics problem, the ROM achieves significant computational reduction relative to the FOM.

The offline-online computational decomposition takes on different forms depending on whether the problem is parameterized. For nonparameterized problems, the offline stage comprises Steps 1 and 2; first a reduced basis is identified using a method in Section 6.3.2, and then the ROM is constructed in terms of the reduced operators (6.10). In the online stage, given an input $\delta v(t) \in \mathcal{Q}$, $t \in \mathcal{I}$, we invoke the ROM (6.11); note that the online stage requires only $\mathcal{O}(N^\star)$ operations.

For parameterized problems, the offline stage comprises only Step 1; either a global reduced basis or a set of reduced bases is constructed using a method in Section 6.3.3. In the online stage, given $\boldsymbol{\mu} \in \mathcal{P}$, we first identify an appropriate reduced basis: For the parameter-domain decomposition method discussed in Section 6.3.3.3, this step requires the identification of the subdomain $\mathcal{P}^n$ to which $\boldsymbol{\mu}$ belongs; for the

reduced space interpolation method discussed in Section 6.3.3.4, this step involves the interpolation of the reduced bases. We then perform Step 2; project the FOM operators onto $\mathbf{Z}_N$ to identify the ROM operators (6.10). We finally invoke the ROM to approximate the linearized aerodynamics problem for the given $\boldsymbol{\mu} \in \mathcal{P}$ and $\delta v(t) \in \mathcal{Q}$, $t \in \mathcal{I}$. Unlike the online stage for nonparameterized problems, the online stage for parameterized problems requires the access to the FOM in Step 2 and hence requires $\mathcal{O}(N_h)$ operations. Nevertheless, significant speedup can be achieved relative to the FOM as Step 2 is still much cheaper than the unsteady solution of the FOM.

We note that *if* the parameterized FOM operators admit a decomposition that is affine in functions of parameters, then the associated reduced operators can be precomputed in the offline stage and hence the online cost would be $\mathcal{O}(N^\star)$; however, most of the relevant problems in aerodynamics do not admit this so-called affine parameter decomposition. We refer to a review paper [57] for offline-online computational decomposition in the presence of affine parameter decomposition. We also note that it may be appropriate to invoke the empirical interpolation method [10, 31] or its variant to identify an approximate affine decomposition; see Chapter 5 of Volume 2.

6.3.5 Stability of the Galerkin-ROM

As discussed in Section 6.1, the focus of this handbook is on formulation and not theory. However, as time stability of ROMs (6.7) is one of the key issues in model reduction of linearized aerodynamics problems, we briefly mention relevant literature; here, time stability refers to the ability to bound some norm of the solution $\|u(t)\|_\star$ by the initial state and boundary conditions.

Barone et al. [9] and Kalashnikova et al. [39] analyze the time stability of the Galerkin-ROM (6.9). The works show that the ROM is stable if the symmetrized form of the hyperbolic system is used with appropriate boundary conditions. We note that for compressible Euler and Navier–Stokes equations (i) the symmetrized system is described in the entropy variables [35, 11]; (ii) the associated energy norm is given by $(w, v)_{A_0} = \int_\Omega v^T A_0 w dx$, where A_0 is the Jacobian of the conservative variables with respect to the entropy variables; and (iii) the mass matrix in (6.7) is also modified accordingly. Kalashnikova et al. [40] further extend the stability analysis to aeroelasticity problems where the structure is modeled by a linearized von Kármán plate equation.

In addition to analysis, we note there are ROM formulations that are designed to achieve guaranteed stability; we again restrict ourselves to works that have been demonstrated for aerodynamics problems. The Fourier-based formulation of Willcox and Megretski [72] discussed in Section 6.3.2.5, for instance, is guaranteed to preserve stability of the underlying FOM; the method has been applied to model reduction [72] and MPC [36] for which POD yields unstable ROMs. Amsallem and Farhat [5] also pro-

pose an online-efficient stabilization based on Petrov–Galerkin projection and apply it to aeroelastic analysis of a wing-store configuration.

6.3.6 Large-scale applications

We conclude this section on model reduction for linearized aerodynamics problems with a few applications to large-, industry-scale problems.

- *Aeroelastic analysis of the AGARD model 445.6 wing* [64]. In this work Thomas et al. consider flutter prediction of a weakened AGARD model 445.6 wing. The FOMs consist of $N_h \approx 2.6 \times 10^5$ to 7.8×10^5 aerodynamic degrees of freedom. The flutter boundaries for six different values of the base flow Mach number are analyzed. For each flight Mach number, the snapshots are computed for the first five structural resonance modes ($\{\delta\hat{v}^l\}$) and six frequencies ($\{\omega^k\}$); POD is applied to identify a ROM with $N = 55$ modes. The ROM is then used to construct the root loci with respect to the reduced velocity and to provide accurate predictions of flutter velocities.
- *Aeroelastic analysis of a full F-16 aeroelastic configuration* [2]. In this work Amsallem et al. consider model reduction of a full aeroelastic F-16 configuration. The FOM consists of $N_h \approx 2 \times 10^6 + 1.7 \times 10^5$ aerodynamic and structural degrees of freedom, respectively. The parameters are the base flow Mach number and angle of attack. In the offline stage, a set of reduced bases for 83 different flight configurations are prepared using the frequency-domain POD approach; each basis comprises $N = 90$ modes. In the online stage, the reduced bases are interpolated on a manifold as discussed in Section 6.3.3.4. For the five predictive test configurations considered, the error in the $L^2(\mathcal{I})$-norm of the unsteady lift varies from 0.4 % to 7 %. The time to solve the linearized system is reduced by a factor of 90 in the online stage. (However, the online stage also requires the computational of the steady-state equilibrium solution; when this step is taken into account, the overall speedup factor is approximately 7.) The aeroelasticity problem is also considered in [47, 46, 4].
- *Probabilistic analysis of unsteady turbine blades* [17]. In this work Bui-Thanh et al. consider model reduction of a two-blade turbine system to analyze the effect of geometric uncertainties on unsteady lift forces. The FOM consists $N_h \approx 1 \times 10^5$ degrees of freedom. The geometric modes are identified using principal component analysis on data from 145 real blades; geometric perturbations are parameterized using $P = 10$ parameters. The reduced basis are identified using a greedy algorithm modified for the high-dimensional parameter space; the resulting ROM consists of $N = 290$ modes. The reduced model is then invoked for 10,000 different geometries to estimate the distribution of the work per cycle (WPC). Relative to the FOM, the ROM achieves less than 0.5 % error in the mean and 2 % error in

the variance. The time to complete the 10,000 analyses is reduced from 516 hours for the FOM to 1.1 hours for the ROM, a computational reduction by a factor of 468.

6.4 Model reduction for nonlinear aerodynamics

In this section we discuss model reduction of nonlinear aerodynamics problems. While linearized analysis suffices for some aerodynamics scenarios, applications such as shape optimization and flight-parameter sweep require full nonlinear analysis. As some of the model reduction ingredients are the same as those discussed for linearized problems in Section 6.3, we focus on techniques and challenges that are unique to full nonlinear analysis.

6.4.1 Projection methods

While the Galerkin method is by far the most common approach for model reduction of linearized aerodynamics problems, there are a few different projection methods that are commonly used for nonlinear aerodynamics problems. We here review the two most popular methods, the Galerkin and minimum-residual methods, and provide a short discussion of other approaches.

6.4.1.1 Galerkin method

We first introduce the Galerkin approximation of the nonlinear aerodynamics problem (6.2). As in Section 6.3.1, we assume that a sequence of reduced basis spaces $V_{N=1} \subset \cdots \subset V_{N=N_{\max}}$ and the associated hierarchical reduced basis $\{\zeta^j\}_{j=1}^N$ is given; we discuss the procedures to generate the reduced basis in Section 6.4.3. The semi-discrete form of the Galerkin-ROM problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_N(t;\boldsymbol{\mu}) \in V_N$, $t \in \mathcal{I}$, such that

$$\frac{\partial u_N(t;\boldsymbol{\mu})}{\partial t} + r_h(u_N(t;\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } V_N', \tag{6.14}$$

and $u_N(t=0;\boldsymbol{\mu}) = \Pi_N u^0(\boldsymbol{\mu})$. Again, for $g \in V_h'$, the statement $g = 0$ in V_N' should be interpreted as $\langle g, v\rangle = 0\ \forall v \in V_N$. We then evaluate the output $s_N(t;\mu) = q_h(u_N(t;\mu);\mu)$.

We may also consider the algebraic form of the problem. We recall from Section 6.3.1 that we associate any function $v_N \in V_N$ with a generalized coordinate $\mathbf{v}_N \in \mathbb{R}^N$ by $v_N = \mathbf{v}_N^j \zeta^j$; we may also express the FOM generalized coordinate of $v_N \in V_N$ as $\mathbf{v}_h = \mathbf{v}_N^j \boldsymbol{\zeta}^j = \mathbf{Z}_N \mathbf{v}_N \in \mathbb{R}^{N_h}$. Given the basis, we define the ROM residual $\mathbf{r}_N : \mathbb{R}^N \times \mathcal{P} \to \mathbb{R}^N$, output functional $\mathbf{q}_N : \mathbb{R}^N \times \mathcal{P} \to \mathbb{R}$, and mass matrix $\mathbf{M}_N \in \mathbb{R}^{N\times N}$

such that

$$\mathbf{r}_N(\mathbf{w}_N;\boldsymbol{\mu}) \equiv \mathbf{Z}_N^T \mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) = \left(\langle r_h(\mathbf{w}_N^j\boldsymbol{\zeta}^j;\boldsymbol{\mu}),\boldsymbol{\zeta}^i\rangle\right)_{i=1}^N,$$
$$\mathbf{q}_N(\mathbf{w}_N;\boldsymbol{\mu}) \equiv \mathbf{q}_h(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) = q_h(\mathbf{w}_N^j\boldsymbol{\zeta}^j;\boldsymbol{\mu}),$$
$$\mathbf{M}_N \equiv \mathbf{Z}_N^T\mathbf{M}_h\mathbf{Z}_N = \left((\boldsymbol{\zeta}^j,\boldsymbol{\zeta}^i)_{L^2(\Omega)}\right)_{i,j=1}^N.$$

The algebraic form of the Galerkin-ROM problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_N(t;\mu) \in \mathbb{R}^N$, $t \in \mathcal{I}$, such that

$$\mathbf{M}_N \frac{d\mathbf{u}_N(t;\boldsymbol{\mu})}{dt} + \mathbf{r}_N(\mathbf{u}_N(t;\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } \mathbb{R}^N, \tag{6.15}$$

and $\mathbf{u}_N(t=0;\boldsymbol{\mu}) = \mathbf{u}_N^0(\boldsymbol{\mu})$, where $\mathbf{u}_N^0(\boldsymbol{\mu}) \in \mathbb{R}^N$ is the generalized coordinate for $\Pi_N u^0(\boldsymbol{\mu})$. We then evaluate the output $s_N(t;\boldsymbol{\mu}) = \mathbf{q}_N(\mathbf{u}_N(t;\boldsymbol{\mu});\boldsymbol{\mu})$. The operator form (6.14) and the algebraic form (6.15) are equivalent in the sense that $u_N(t;\boldsymbol{\mu}) = \sum_{j=1}^N \mathbf{u}_N^j(t;\boldsymbol{\mu})\boldsymbol{\zeta}^j$.

Most aerodynamics shape optimization and flight-parameter sweep scenarios consider steady-state solutions. The steady-state problem seeks $\mathbf{u}_N(\boldsymbol{\mu}) \in V_N$ such that

$$\mathbf{r}_N(\mathbf{u}_N(\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } \mathbb{R}^N, \tag{6.16}$$

and then evaluates $s_N(\boldsymbol{\mu}) \equiv \mathbf{q}_N(\mathbf{u}_N(\boldsymbol{\mu});\boldsymbol{\mu})$.

We make a few observations. First, the reduced-order Galerkin problem (6.14) (or (6.15)) is in semi-discrete form; as described for FOMs in Section 6.2.3, we apply a suitable time marching scheme to obtain a full-discrete form of the Galerkin-ROM problem. Second, the steady-state problem (6.16) is solved using a pseudo-time continuation method as discussed for FOMs in Section 6.2.3, and hence the unsteady equations are relevant also for steady-state problems. Third, although the approximation space V_N is of dimension N, the computation of the reduced residual $\mathbf{r}_N(\mathbf{w}_N;\boldsymbol{\mu}) = \mathbf{Z}_N^T\mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu})$ requires $\mathcal{O}(N_h) \gg \mathcal{O}(N)$ operations, because the FOM residual $\mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ must be projected onto the reduced basis $\mathbf{Z}_N \in \mathbb{R}^{N_h\times N}$. Hyperreduction, which enables $\mathcal{O}(N)$ evaluation of the residual, is discussed in Section 6.4.2.

6.4.1.2 Minimum-residual method

We now discuss an alternative projection method: the minimum-residual method. As the name suggests, we choose the element of V_N that minimizes the (dual) norm of the residual as our ROM solution. For steady problems, the minimum-residual problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_N(\boldsymbol{\mu}) \in V_N$ such that

$$u_N(\boldsymbol{\mu}) = \underset{w_N\in V_N}{\arg\inf} \left\| r_h(u_N(\boldsymbol{\mu});\boldsymbol{\mu})\right\|_{V_h'} \equiv \underset{w_N\in V_N}{\arg\inf} \sup_{v_h\in V_h} \frac{\langle r_h(u_N(\boldsymbol{\mu});\boldsymbol{\mu}), v_h\rangle}{\|v_h\|_{V_h}}. \tag{6.17}$$

An algebraic form of the problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\begin{aligned} \mathbf{u}_N(\mu) &= \underset{\mathbf{w}_N \in \mathbb{R}^N}{\arg\inf} \|\mathbf{r}_h(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu})\|^2_{\mathbf{W}_h} \\ &= \underset{\mathbf{w}_N \in \mathbb{R}^N}{\arg\inf} \mathbf{r}_h(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu})^T \mathbf{W}_h \mathbf{r}_h(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu}), \end{aligned} \tag{6.18}$$

where $\mathbf{W}_h \in \mathbb{R}^{N_h \times N_h}$ is the inner product matrix; the choice $\mathbf{W}_h = \mathbf{V}_h^{-1}$ for $\mathbf{V}_{h,ij} = (\varphi^j, \varphi^i)_{V_h}$, $i, j = 1, \ldots, N_h$, results in (6.18) to be equivalent to (6.17).

The minimum-residual formulation can also be extended to unsteady problems as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_N(t; \boldsymbol{\mu}) \in \mathbb{R}^N$, $t \in I$, such that

$$\mathbf{u}_N^k(\mu) = \underset{\mathbf{w}_N \in \mathbb{R}^N}{\arg\inf} \|\mathbf{r}_{h,\Delta t}(\mathbf{Z}_N \mathbf{w}_N; \{\mathbf{Z}_N \mathbf{u}_N^l(\boldsymbol{\mu})\}_{l=1}^{k-1}; \boldsymbol{\mu})\|_{\mathbf{W}_h}, \quad k = 1, \ldots, K.$$

The formulation minimizes the residual associated with each time step.

We make a few observations. First, the minimum-residual method can be cast as a Petrov–Galerkin method [50]; as a result, the method is also referred to as a least-squares Petrov–Galerkin method [21]. Second, the minimum-residual method is a very common approach for model reduction of steady nonlinear aerodynamics problems and has been used in works including [43, 44, 45, 69, 80, 79]. Third, similarly to the Galerkin method, the evaluation of the FOM residual in (6.18) requires $\mathcal{O}(N^h) \gg \mathcal{O}(N)$ operations. Hyperreduction, which enables $\mathcal{O}(N)$ evaluation of the residual, is discussed in Section 6.4.2.

6.4.1.3 Other approaches: interpolation- and L^1-based ROMs

While the Galerkin and minimum-residual methods are most commonly used methods for model reduction of nonlinear aerodynamics problems, some works have used interpolation-based ROMs, which deduce the reduced basis coefficients $\mathbf{u}_N \in \mathbb{R}^N$ through interpolation. In the context of aerodynamics, the approach has been applied to flight-parameter sweep scenarios: Bui-Thanh et al. [16] deduce the reduced basis coefficients using cubic splines for two-dimensional Euler flow over an airfoil; Franz et al. [30] deduce the reduced basis coefficients using a manifold learning technique for three-dimensional Euler flow over a wing.

We can also consider minimization of different norms of the residual to deduce $\mathbf{u}_N \in \mathbb{R}^N$. Of particular interest is the L^1-norm, which is a more natural norm for hyperbolic equations. Based on this observation, Abgrall and Crisovan [1] propose an ROM which identifies the solution through L^1-minimization and apply it to parameterized transonic Euler flow over an airfoil.

6.4.2 Hyperreduction

As discussed in Section 6.4.1, seeking the solution in a reduced space $V_N \subset V_h$ is insufficient to achieve $\mathcal{O}(N^\bullet)$ online cost for nonlinear problems. We need a means to approximate the projection of the FOM residual $\mathbf{r}_h(\mathbf{w}_N;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ onto the reduced basis $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N}$ in $\mathcal{O}(N)$ operations for the Galerkin method, and there is an analogous requirement for the minimum-residual method. This is the goal of hyperreduction, a term coined by Ryckelynck [58]. We here present hyperreduction approaches that have been used for aerodynamics problems; we refer to Chapter 5 of Volume 2 for a more general coverage. We follow the convention used in much of the hyperreduction literature and present formulations in algebraic form.

6.4.2.1 Minimum-residual collocation methods

We first consider arguably the simplest hyperreduction method: the minimum-residual method with a collocation-based approximation of the residual norm. To begin, we assume that the FOM residual can be decomposed into elemental contributions; the assumption holds for finite volume and finite element methods – the two most commonly used discretizations in aerodynamics – as the FOM residual is assembled element by element. We express this elemental decomposition of the residual as

$$\mathbf{r}_h(\mathbf{w}_h;\boldsymbol{\mu}) = \sum_{\kappa=1}^{N_e} \mathbf{r}_{h,\kappa}(\mathbf{w}_h;\boldsymbol{\mu}) \quad \text{in } \mathbb{R}^{N_h},$$

where $N_e \equiv |\mathcal{T}_h|$ is the number of elements and $\mathbf{r}_{h,\kappa} : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{N_h}$ is the FOM residual operator for the κ-th element. Note that $\mathbf{r}_{h,\kappa}(\mathbf{w}_h;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is mostly sparse, because a given element contributes to a small number of residual degrees of freedom.

We now proceed with hyperreduction. We first choose a small subset of $\tilde{N}_e$ sample elements $\tilde{\mathcal{T}}_h \subset \mathcal{T}_h$ so that $N \leq \tilde{N}_e \ll N_e$; we denote the associated sample element indices by $\tilde{T}$. (Quantities associated with hyperreduction bear $\tilde{\cdot}$ throughout this section.) We then consider the following hyperreduced approximation of the minimum-residual problem (6.18): Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\tilde{\mathbf{u}}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\tilde{\mathbf{u}}_N(\boldsymbol{\mu}) = \arg\min_{\mathbf{w}_N \in \mathbb{R}^N} \left\| \sum_{\kappa \in \tilde{\mathcal{T}}_h} \mathbf{r}_{h,\kappa}(\mathbf{w}_h;\boldsymbol{\mu}) \right\|_2. \tag{6.19}$$

We observe that if $\tilde{N}_e = \mathcal{O}(N) \ll N_h$, then we can solve this hyperreduced minimum-residual problem in $\mathcal{O}(N^\bullet)$ operations.

We can also describe the hyperreduction procedure algebraically. To this end, we first identify the set of $\tilde{N}_{\tilde{\mathcal{I}}}$ residual sample indices $\tilde{\mathcal{I}} \equiv \{\tilde{i}_1, \ldots, \tilde{i}_{\tilde{N}_{\tilde{\mathcal{I}}}}\}$ associated with the sample elements $\tilde{\mathcal{T}}_h$. For finite volume methods, $\tilde{N}_{\tilde{\mathcal{I}}} = N_c \tilde{N}_e$ as the number

of residual degrees of freedom associated with each element is equal to the number of components N_c in the PDE. We then introduce the associated sample matrix $\mathbf{P} = (e^{i_1}, \ldots, e^{i_{\tilde{N}_{\tilde{\mathcal{I}}}}}) \in \mathbb{R}^{N_h \times \tilde{N}_{\tilde{\mathcal{I}}}}$ whose j-th column is the canonical unit vector $e^{i_j} \in \mathbb{R}^{N_h}$. The minimum-residual collocation problem (6.19) is equivalent to

$$\tilde{\mathbf{u}}_N(\boldsymbol{\mu}) = \arg\min_{\mathbf{w}_N \in \mathbb{R}^N} \left\| \mathbf{P}^T \mathbf{r}_h(\mathbf{w}_h; \boldsymbol{\mu}) \right\|_2.$$

Here, to achieve hyperreduction, we evaluate the operator $(\mathbf{P}^T \mathbf{r}_h) : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{\tilde{N}_{\tilde{\mathcal{I}}}}$ by first checking which indices are requested by $\mathbf{P}$ and then computing the residual for only those indices.

The key to a successful hyperreduction by the minimum-residual collocation formulation lies in the selection of the sample elements $\tilde{\mathcal{T}}_h$, which is performed in the offline stage. We here review a few approaches that have been applied to aerodynamics problems.

Physics-informed selection. To our knowledge, LeGresley and Alonso [44] were the first to consider hyperreduction for aerodynamics problems. In the work, hyperreduction is achieved by including only 20 %–30 % of the elements near the airfoil in $\tilde{\mathcal{T}}_h$. This strategy was specialized for aerodynamic shape optimization, in which most of the solution variations are in the vicinity of the airfoil. Vendl et al. [66] also consider a physics-informed hyperreduction in the context of flight-parameter sweep; however, as the parameter affects the boundary conditions, they also included elements on the far-field boundary in $\tilde{\mathcal{T}}_h$.

Gappy POD on the state snapshots. To devise a more systematic approach to identify sample elements, Washabaugh et al. [69, 70] invoke gappy POD [29] on the solution snapshots $\mathbf{S} \equiv (\mathbf{u}_h(\boldsymbol{\mu}^1), \ldots, \mathbf{u}_h(\boldsymbol{\mu}^{N_s})) \in \mathbb{R}^{N_h \times N_s}$ and set the sample indices $\tilde{\mathcal{I}}$ for the minimum-residual collocation method equal to the gappy POD sample indices. Specifically, the method successively processes sets of snapshots $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$ in smaller batches $\mathbf{S}_k = (\mathbf{u}_h(\boldsymbol{\mu}^1), \ldots, \mathbf{u}_h(\boldsymbol{\mu}^k)) \in \mathbb{R}^{N_h \times k}$, $k = 1, \ldots, N_s$; assuming the sample indices $\tilde{\mathcal{I}}$ have been constructed for $\tilde{\mathbf{S}}_{k-1}$, the sample indices are updated for the batch $\tilde{\mathbf{S}}_k$ as follows:

1. Compute the gappy POD reconstruction of the snapshots:
   $\tilde{\mathbf{S}}_k = \mathbf{Z}_N (\mathbf{P}^T \mathbf{Z}_N)^\dagger \mathbf{P}^T \mathbf{S} \in \mathbb{R}^{N_h \times N_s}$. Here, $(\cdot)^\dagger$ denotes the pseudo-inverse.
2. Set $i^\star = \arg\max_{i \in [1, N_h]} \max_{j \in [1, N_s]} |\mathbf{S}_k - \tilde{\mathbf{S}}_k|_{ij}$.
3. Add the sample index: $\tilde{\mathcal{I}} = \tilde{\mathcal{I}} \cup i^\star$; update the sample matrix $\mathbf{P}$ accordingly.

This approach assumes that the sample indices with which the state can be approximated work well also for the residual; this assumption allows the method to work with the state and not the residual, which significantly reduces the offline cost relative to Gauss–Newton approximate tensor (GNAT) and empirical quadrature procedure (EQP) methods discussed in Sections 6.4.2.2 and 6.4.2.3, respectively. The method has been applied to full aircraft configuration under shape deformations [69] as discussed further in Section 6.4.5.

6.4.2.2 Gauss–Newton approximate tensor method

The GNAT method [21, 22] approximates the minimum-residual problem (6.18) for $\mathbf{W}_h = \mathbf{I}$, $\mathbf{u}_N(\boldsymbol{\mu}) = \arg\inf_{\mathbf{w}_N \in \mathbb{R}^N} \|\mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu})\|_2$, using a gappy POD approximation [29] of the residual and Jacobian and then solves the problem using the Gauss–Newton method. (Although the original work [21, 22] considers unsteady problems, for notational simplicity we here consider a steady problem.) The solution of (6.18) by the Gauss–Newton method requires successive solution of the linear least-squares problem: Find the update $\delta\mathbf{w}_N \in \mathbb{R}^N$ such that

$$\delta\mathbf{w}_N = \arg\min_{\mathbf{v}_n \in \mathbb{R}^N} \left\| \mathbf{J}_h(\mathbf{Z}_N\mathbf{v}_N)\mathbf{Z}_N\mathbf{v}_N + \mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N) \right\|_2. \tag{6.20}$$

The solution is then updated according to $\mathbf{w}_N \leftarrow \mathbf{w}_N + \alpha\delta\mathbf{w}_N$, where the step length $\alpha \in (0,1]$ is deduced by line search. The cost to solve this least-squares problem is $\mathcal{O}(N_h)$ as it requires the FOM residual and Jacobian.

To approximately solve (6.20) in $\mathcal{O}(N)$ operations, the GNAT method prepares three ingredients for a gappy POD approximation of the residual $\mathbf{r}_h : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{N_h}$: (i) a reduced basis for the residual $\mathbf{Z}^r \in \mathbb{R}^{N_h \times N_r}$, (ii) a set of sample indices $\mathcal{I} = \{i_1, \ldots, i_{\tilde{N}_\mathcal{I}}\}$ for $\tilde{N}_\mathcal{I} \geq N_r$, and (iii) the associated sample matrix $\mathbf{P} = (e^{i_1}, \ldots, e^{i_{\tilde{N}_\mathcal{I}}}) \in \mathbb{R}^{N_h \times \tilde{N}_\mathcal{I}}$ whose j-column is the canonical unit vector $e^{i_j} \in \mathbb{R}^{N_h}$. The residual is then approximated by regression: $\tilde{\mathbf{r}}_h(\mathbf{Z}_N\mathbf{w}_N) = \arg\min_{\mathbf{v} \in \mathbf{v}_r} \|\mathbf{P}^T(\mathbf{r}_h(\mathbf{Z}_N\mathbf{w}_N) - \mathbf{Z}^r\mathbf{v})\|_2$. The Jacobian is similarly approximated using a reduced basis for the Jacobian $\mathbf{Z}^J \in \mathbb{R}^{N_h \times N_J}$ and the same sample matrix $\mathbf{P}^T$ by regression: $\tilde{\mathbf{J}}_h(\mathbf{Z}_N\mathbf{w}_N)\mathbf{Z}_{N,j} = \arg\min_{\mathbf{v} \in \mathbf{v}_r} \|\mathbf{P}^T(\mathbf{J}_h(\mathbf{Z}_N\mathbf{w}_N)\mathbf{Z}_{N,j} - \mathbf{Z}^J\mathbf{v})\|_2$, $j = 1, \ldots, N$. The GNAT method solves this gappy POD-approximated minimum-residual problem using a gappy POD-approximated Gauss–Newton method.

Carlberg et al. [21, 22] introduce four variants of the GNAT method, named procedure 0–3. We here consider only procedure 1, which has been shown to exhibit good accuracy and robustness for unsteady aerodynamics problems. We outline the offline and online stages of the GNAT method.

Offline stage. In the offline stage, we construct all ingredients of GNAT: $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N_r}$, $\mathbf{Z}^r = \mathbf{Z}^J \in \mathbb{R}^{N_h \times N_r}$, and $\mathbf{P} \in \mathbb{R}^{N_h \times \tilde{N}_\mathcal{I}}$.

1. Choose a snapshot parameter set $\Xi_t = \{\boldsymbol{\mu}^i\}_{i=1}^{N_t} \subset \mathcal{P}$.
2. Solve the FOM (6.4) for each $\boldsymbol{\mu} \in \Xi_t$ to obtain $\{\mathbf{u}_h(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \Xi_t}$. Apply POD to the snapshots to obtain a state reduced basis $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N}$.
3. Solve the nonhyperreduced ROM (6.17) for each $\boldsymbol{\mu} \in \Xi_t$. Collect residual snapshots $\{\mathbf{r}_h(\mathbf{Z}_N\mathbf{u}_N(\boldsymbol{\mu});\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \Xi_s}$. Apply POD to the set to obtain a residual reduced basis $\mathbf{Z}^r \in \mathbb{R}^{N_h \times N_r}$ for $N_r \geq N$. Set $\mathbf{Z}^J = \mathbf{Z}^r$.
4. Apply the gappy POD procedure described in Section 6.4.2.1 (for the state snapshots) to the residual snapshots to determine the sample index $\tilde{\mathcal{I}}$ with $\tilde{N}_\mathcal{I} \geq N_r$ and the associated sample matrix $\mathbf{P} \in \mathbb{R}^{N_h \times \tilde{N}_\mathcal{I}}$.
5. Precompute $\mathbf{A} \equiv (\mathbf{P}^T\mathbf{Z}^J)^\dagger \in \mathbb{R}^{N_J \times N_\mathcal{I}}$ and $\mathbf{B} \equiv (\mathbf{Z}^J)^T\mathbf{Z}^r(\mathbf{P}^T\mathbf{Z}^r)^\dagger \in \mathbb{R}^{N_J \times N_\mathcal{I}}$.

Online stage. In the online stage, given $\boldsymbol{\mu} \in \mathcal{P}$, we seek $\mathbf{u}_N(\boldsymbol{\mu})$ such that

$$\mathbf{u}_N(\boldsymbol{\mu}) = \arg\min_{\mathbf{w}_N \in \mathbb{R}^N} \left\| \mathbf{Z}^r (\mathbf{P}^T \mathbf{Z}^r)^\dagger \mathbf{P}^T \mathbf{r}_h(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu}) \right\|_2.$$

This problem is solved using the Gauss–Newton method as follows:

1. Form $\mathbf{C}(\mathbf{w}_N) = \mathbf{P}^T \mathbf{J}_h(\mathbf{Z}_N \mathbf{w}_N) \mathbf{Z}_N$ and $\mathbf{D}(\mathbf{w}_N) = \mathbf{P}^T \mathbf{r}_h(\mathbf{Z}_N \mathbf{w})$.
2. Solve the linear least-squares problem: Find $\delta \mathbf{w}_N \in \mathbb{R}^N$ such that

$$\delta \mathbf{w}_N = \arg\min_{\mathbf{v} \in \mathbb{R}^N} \left\| \mathbf{AC}(\mathbf{w}_N) \mathbf{v} + \mathbf{BD}(\mathbf{w}_N) \right\|_2.$$

3. Update $\mathbf{w}_N \leftarrow \mathbf{w}_N + \alpha \delta \mathbf{w}_N$, where α is determined from line search.
4. If converged, terminate; otherwise return to 1.

The online computational cost is $\mathcal{O}(N^\bullet)$ and is independent of the FOM. To evaluate the output, the GNAT method does not explicitly hyperreduce the output functional $\mathbf{q}_h : \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}$, but simply leverages the fact that output functionals for most aerodynamics problems require evaluation on a small subset of elements, e. g., elements on aerodynamics surfaces. Hence, output evaluation constitutes a small fraction of the overall cost.

The GNAT method has been applied to large-scale simulation of (nonparameterized) unsteady turbulent flow over the Amhed body [21, 22] as discussed further in Section 6.4.5. We also refer to [20] for a detailed analysis of the method.

6.4.2.3 Galerkin method with empirical quadrature procedure

One of the limitations of the hyperreduction methods discussed in the previous two sections is that they do not provide a quantitative control of the solution and/or output error due to hyperreduction. One approach which provides such quantitative error control is the EQP [77, 75, 76]. To describe the method, we first introduce the hyperreduced residual $\tilde{\mathbf{r}}_N : \mathbb{R}^N \times \mathcal{P} \to \mathbb{R}^N$ and output functional $\tilde{\mathbf{q}}_N : \mathbb{R}^N \times \mathcal{P} \to \mathbb{R}$ of the form

$$\tilde{\mathbf{r}}_N(\mathbf{w}_N; \boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^r \mathbf{r}_{N,\kappa}(\mathbf{w}_N; \boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^r \mathbf{Z}_N^T \mathbf{r}_{h,\kappa}(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu}), \tag{6.21}$$

$$\tilde{\mathbf{q}}_N(\mathbf{w}_N; \boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^q \mathbf{q}_{N,\kappa}(\mathbf{w}_N; \boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^q \mathbf{q}_{h,\kappa}(\mathbf{Z}_N \mathbf{w}_N; \boldsymbol{\mu}); \tag{6.22}$$

here $\rho^r \in \mathbb{R}^{N_e}$ and $\rho^q \in \mathbb{R}^{N_e}$ are the EQP weights that are sparse (i. e., most entries are zero) so that the summands need to be evaluated for a small subset of elements. The

associated hyperreduced problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\tilde{\mathbf{u}}_N(t;\boldsymbol{\mu}) \in \mathbb{R}^N$, $t \in \mathcal{I}$, such that

$$\mathbf{M}_N \frac{d\tilde{\mathbf{u}}_N(t;\boldsymbol{\mu})}{dt} + \check{\mathbf{r}}_N(\tilde{\mathbf{u}}_{N,M}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \quad \text{in } \mathbb{R}^N,$$

for $\tilde{\mathbf{u}}_N(t=0;\boldsymbol{\mu}) = \mathbf{u}_N^0(\boldsymbol{\mu})$, and evaluate the output $\tilde{s}_N(t;\boldsymbol{\mu}) = \tilde{\mathbf{q}}_N(\tilde{\mathbf{u}}_N(t;\boldsymbol{\mu});\boldsymbol{\mu})$. We wish to find EQP weights $\rho^r \in \mathbb{R}^{N_e}$ and $\rho^q \in \mathbb{R}^{N_e}$ so that (i) $|s_N(t;\boldsymbol{\mu}) - \tilde{s}_N(t;\boldsymbol{\mu})| \leq \delta$ for a user-prescribed tolerance $\delta \in \mathbb{R}_{>0}$ and (ii) $\mathrm{nnz}(\rho^r) = \mathcal{O}(N)$ and $\mathrm{nnz}(\rho^q) = \mathcal{O}(N)$. The two conditions ensure the accuracy and online efficiency, respectively, of the hyperreduced ROM.

The EQP weights are computed in the offline stage by solving linear programs (LPs). We first introduce a parameter training set $\Xi_\mathrm{t} \equiv \{\hat{\boldsymbol{\mu}}^j\}_{j=1}^{N_t}$ and the associated training states $U_\mathrm{t} = \{\hat{\mathbf{u}}_N(\boldsymbol{\mu})\}_{\boldsymbol{\mu}\in\Xi_\mathrm{t}}$. The training states can be the nonhyperreduced ROM solution as it is done for GNAT; however, when used in conjunction with the greedy algorithm, U_t can be the hyperreduced ROM solution in a given iteration [75]. The general form of the linear program, denoted $\mathsf{LP}^\bullet$, where $^\bullet$ is the placeholder for the residual "r" or output function "q," is as follows: Find the basic feasible solution $\rho^{\bullet,\star} \in \mathbb{R}^{N_e}$ such that

$$\rho^{\bullet,\star} = \underset{\rho^\bullet\in\mathbb{R}^{N_e}}{\arg\min} \sum_{\kappa=1}^{N_e} \rho_\kappa^\bullet,$$

subject to nonnegativity constraints

$$\rho_\kappa^\bullet \geq 0, \quad \kappa = 1,\dots,N_e,$$

and manifold-accuracy and constant-integration constraints

$$\left(\begin{array}{ccc} \mathbf{a}_1^\bullet(\boldsymbol{\mu}^1) & \cdots & \mathbf{a}_{N_e}^\bullet(\boldsymbol{\mu}^1) \\ \vdots & \ddots & \vdots \\ \mathbf{a}_1^\bullet(\boldsymbol{\mu}^{N_\mathrm{t}}) & \cdots & \mathbf{a}_{N_e}^\bullet(\boldsymbol{\mu}^{N_\mathrm{t}}) \\ \hline |\kappa_1| & \cdots & |\kappa_{N_e}| \end{array}\right) \left(\begin{array}{c} \rho_1^\bullet \\ \vdots \\ \rho_{N_e}^\bullet \end{array}\right) \lesseqgtr \left(\begin{array}{c} \mathbf{b}^\bullet(\boldsymbol{\mu}^1) \\ \vdots \\ \mathbf{b}^\bullet(\boldsymbol{\mu}^{N_\mathrm{t}}) \\ \hline |\Omega| \end{array}\right) \pm \left(\begin{array}{c} \boldsymbol{\delta}^\bullet \\ \vdots \\ \boldsymbol{\delta}^\bullet \\ \hline \delta_\Omega \end{array}\right), \tag{6.23}$$

where $\mathbf{a}_\kappa(\boldsymbol{\mu}) \in \mathbb{R}^{N_c^\bullet}$, $\kappa = 1,\dots,N_e$, is a set of vectors that depends on the specific manifold accuracy constraint to be described shortly, $N_c^\bullet$ is the number of constraints per training parameter, $b^\bullet(\boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \mathbf{a}_\kappa^\bullet(\boldsymbol{\mu}) \in \mathbb{R}^{N_c^\bullet}$, $\boldsymbol{\delta}^\bullet \in \mathbb{R}^{N_c^\bullet}$ is the manifold-accuracy tolerance, $|\kappa| \equiv \int_\kappa dx$, and $|\Omega| \equiv \int_\Omega dx$. The LP can be solved using a simplex method. We now introduce specific manifold accuracy constraints for the residual (6.21) and output functional (6.22).

Residual EQP. The residual EQP weights $\rho^r \in \mathbb{R}^{N_e}$ are found by solving $\mathsf{LP}^r(\Xi_t, U_t, \delta^r)$. As our goal is to control the output error, we introduce a reduced basis approximation of the dual problem: Given $\boldsymbol{\mu} \in \mathcal{P}$ and the linearization state $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$, find

the dual solution $\mathbf{z}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\mathbf{J}_N(\hat{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu})^T \mathbf{z}_N(\boldsymbol{\mu}) = \mathbf{g}_N(\hat{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu}) \quad \text{in } \mathbb{R}^N.$$

As discussed in the context of balanced POD in Section 6.3.2.4, the dual solution relates the residual to the output error. The manifold-accuracy constraint (6.23) for the residual imposes $N_c^r = N$ constraints per training parameter given by

$$\mathbf{a}_\kappa^r(\boldsymbol{\mu}) \equiv \left|\mathbf{z}_N(\boldsymbol{\mu})\right| \circ \left|\mathbf{r}_{N,\kappa}(\hat{\mathbf{u}}(\boldsymbol{\mu});\boldsymbol{\mu})\right| \quad \text{in } \mathbb{R}^N,$$

and $\boldsymbol{\delta}^r = \frac{\delta^r}{2}\mathbf{1}_N$, where $\mathbf{1}_N \in \mathbb{R}^N$ is the vector of all ones and $\circ$ is the Hadamard (i. e., entrywise) product. Overall, LP^r has N_e unknowns, N_e nonnegativity constraints, and $2(N_t N + 1)$ inequality constraints (where the leading factor of two accounts for the upper and lower bounds in (6.23)).

Output functional EQP. The output EQP weights $\rho^q \in \mathbb{R}^{N_e}$ are similarly found by solving $\mathsf{LP}^q(\Xi_t, U_t, \delta^q)$. The manifold-accuracy constraint (6.23) for the output functional imposes $N_c^q = 1$ constraint per training parameter given by

$$\mathbf{a}_\kappa^q(\boldsymbol{\mu}) \equiv \mathbf{q}_\kappa(\tilde{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu}) \quad \text{in } \mathbb{R}.$$

Overall, LP^q has N_e unknowns, N_e nonnegativity constraints, and $2(N_t + 1)$ inequality constraints; the LP for the output functional is much smaller than that for the residual.

Output a posteriori error estimate. The EQP method also provides an a posteriori error estimate for the output error. The error estimate is based on the dual-weighted residual method [14]. To this end, we first introduce a separate reduced basis for the dual problem $\mathbf{Z}_N^{\mathrm{du}} \in \mathbb{R}^{N_h \times N}$, which is different from the primal reduced basis $\mathbf{Z}_N \in \mathbb{R}^{N_h \times N}$. We then introduce an EQP approximation of the residual, Jacobian, and output gradient evaluated with respect to the dual reduced basis $\mathbf{Z}_N^{\mathrm{du}}$: $\tilde{\mathbf{r}}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{P} \to \mathbb{R}^N$, $\tilde{\mathbf{J}}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N \times N}$ and $\tilde{\mathbf{g}}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$ such that

$$\begin{aligned}
\tilde{\mathbf{r}}_N^{\mathrm{du}}(\mathbf{w};\boldsymbol{\mu}) &\equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{r}_{N,\kappa}^{\mathrm{du}}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{Z}_N^{\mathrm{du}\,T} \mathbf{r}_{h,\kappa}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}), \\
\tilde{\mathbf{J}}_N^{\mathrm{du}}(\mathbf{w};\boldsymbol{\mu}) &\equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{J}_{N,\kappa}^{\mathrm{du}}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{Z}_N^{\mathrm{du}\,T} \mathbf{J}_{h,\kappa}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu})\mathbf{Z}_N^{\mathrm{du}}, \\
\tilde{\mathbf{g}}_N^{\mathrm{du}}(\mathbf{w};\boldsymbol{\mu}) &\equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{g}_{N,\kappa}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}) \equiv \sum_{\kappa=1}^{N_e} \rho_\kappa^\eta \mathbf{Z}_N^{\mathrm{du}\,T} \mathbf{g}_{h,\kappa}(\mathbf{Z}_N\mathbf{w}_N;\boldsymbol{\mu}),
\end{aligned}$$

for some EQP weights $\rho^\eta \in \mathbb{R}^{N_e}$ computed in the offline stage. The EQP dual problem is as follows: Given $\boldsymbol{\mu} \in \mathcal{D}$ and $\tilde{\mathbf{u}}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$, find $\tilde{\mathbf{z}}_N^{\mathrm{du}}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\tilde{\mathbf{J}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu})^T \tilde{\mathbf{z}}_N^{\mathrm{du}}(\boldsymbol{\mu}) = \tilde{\mathbf{g}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu}) \quad \text{in } \mathbb{R}^N.$$

The output error estimate is given by

$$\tilde{\eta}_N^{\mathrm{rb}}(\boldsymbol{\mu}) \equiv \left|\tilde{\mathbf{z}}_N^{\mathrm{du}}(\boldsymbol{\mu})^T \tilde{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu})\right|.$$

Assuming $\mathrm{nnz}(\rho^\eta) = \mathcal{O}(N)$, this error estimate is computable in $\mathcal{O}(N)$ operations.

The output error estimate EQP weights $\rho^\eta \in \mathbb{R}^{N_e}$ is given by a linear program $\mathsf{LP}^\eta(\Xi_{\mathrm{t}}, U_{\mathrm{t}}, \delta^\eta)$. The manifold-accuracy constraint (6.23) for the output error estimate imposes $N_c^\eta = 3N$ constraints per training parameter given by

$$\mathbf{a}_{N,\kappa}^\eta(\boldsymbol{\mu}) \equiv \begin{pmatrix} \max\{|\mathbf{z}_N^{\mathrm{du}}(\boldsymbol{\mu})|, \mathbf{z}_{\min}^{\mathrm{du}}\} \circ |\mathbf{r}_{N,\kappa}^{\mathrm{du}}(\boldsymbol{\mu})| \\ \max\{|\mathbf{r}_N^{\mathrm{du}}(\boldsymbol{\mu})|, \mathbf{r}_{\min}^{\mathrm{du}}\} \circ |\mathbf{J}_N^{\mathrm{du}}(\boldsymbol{\mu})^{-T} \mathbf{J}_{N,\kappa}^{\mathrm{du}}(\boldsymbol{\mu})^T \mathbf{z}_N^{\mathrm{du}}(\boldsymbol{\mu})| \\ \max\{|\mathbf{r}_N^{\mathrm{du}}(\boldsymbol{\mu})|, \mathbf{r}_{\min}^{\mathrm{du}}\} \circ |\mathbf{J}_N^{\mathrm{du}}(\boldsymbol{\mu})^{-T} \mathbf{g}_{N,\kappa}^{\mathrm{du}}(\boldsymbol{\mu})| \end{pmatrix} \quad \text{in } \mathbb{R}^{3N};$$

here $\mathbf{z}_{\min}^{\mathrm{du}} \equiv (\upsilon\boldsymbol{\delta}^\eta/N)^{1/2}/2$ and $\mathbf{r}_{\min} \equiv (\boldsymbol{\delta}^\eta/(\upsilon N))^{1/2}/4$ for $\upsilon \equiv \|\mathbf{z}_N^{\mathrm{du}}(\boldsymbol{\mu})\|_2/\|\mathbf{r}_N^{\mathrm{du}}(\boldsymbol{\mu})\|_2$, the maximum operator is taken entrywise, and all entities with the argument $\boldsymbol{\mu}$ are evaluated about the state $\tilde{\mathbf{u}}(\tilde{\boldsymbol{\mu}})$ and the parameter $\boldsymbol{\mu}$; e. g., $\mathbf{r}_{N,\kappa}^{\mathrm{du}}(\boldsymbol{\mu}) \equiv \mathbf{r}_{N,\kappa}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu})$. Overall, LP^η has N_e unknowns, N_e nonnegativity constraints, and $2(3N_tN + 1)$ inequality constraints.

The EQP method has been applied to two- and three-dimensional turbulent aerodynamic flows in the context of flight-parameter sweep [75, 76]. The rapidly computable output error estimate enables the construction of a reduced model that meets the user-prescribed error tolerance in an automated manner in the offline stage and provides reliable predictions in the online stage.

6.4.2.4 Choice of a hyperreduction procedure

We make a few remarks about the choice of a hyperreduction method for aerodynamics problems. One of the challenges in hyperreduction for aerodynamics is that the FOM is typically very large, with millions of degrees of freedom, and hence the offline training cost cannot be neglected in a practical engineering setting. This is unlike some classical model reduction scenarios, where the offline cost is often neglected. The other challenge is the stability; the hyperreduced system must provide time stability for unsteady simulations to produce meaningful results and for steady simulations to find solutions using the PTC procedure. There exist many examples in the literature where a hyperreduction method that works well for other nonlinear problems has been found to be insufficient for aerodynamics problems.

For instance, the missing point estimate [8] chooses the sample indices $\tilde{\mathcal{I}}$ such that the associated sample matrix $\mathbf{P}$ minimizes the condition number of $\mathbf{Z}_N^T\mathbf{P}\mathbf{P}^T\mathbf{Z}_N$; however, the method was deemed too expensive for steady aerodynamics problems in [66]. The empirical interpolation method [10, 31] and its discrete counterpart [23], which are arguably the most common hyperreduction methods, to our knowledge have

seen limited use in aerodynamics; in fact, Carlberg et al. [22, 20] report temporal instability for turbulent unsteady flows. Similarly, the GNAT method, which has been used successfully for nonparameterized unsteady problems, was deemed too expensive for parameterized steady aerodynamics problems in Washabaugh [70]; we also refer to the work for detailed discussion of the choice of a hyperreduction method.

6.4.3 Construction of reduced basis

Techniques to find an appropriate reduced basis for nonlinear aerodynamics problems are largely the same as those for linearized aerodynamics problems discussed in Sections 6.3.2 and 6.3.3. By far the most popular method to generate reduced bases for nonlinear aerodynamics problems is POD [43, 44, 45, 69, 80, 79, 66, 21, 22]. For unsteady problems, the snapshots are collected for K time steps to yield $\mathbf{S} = \{\mathbf{u}_h^k \approx \mathbf{u}_h(t^k)\}_{k=1}^K$; for parameterized problems, the snapshots are collected for N_t training parameters $\Xi_\mathrm{t} \equiv \{\boldsymbol{\mu}^i\}_{i=1}^{N_\mathrm{t}}$ to yield $\mathbf{S} = \{\mathbf{u}_h(\boldsymbol{\mu})\}_{\boldsymbol{\mu}\in\Xi_\mathrm{t}}$. Given the snapshot matrix $\mathbf{S}$, the POD procedure to identify $\mathbf{Z}_N \in \mathbb{R}^{N_h\times N}$ is described in the context of linearized problems in Section 6.3.2.2. For the EQP method which provides an online efficient a posteriori error estimate, it is also possible to identify the reduced basis using the weak greedy algorithm discussed in Section 6.3.3.2 [75, 76]. We note that while the "standard" POD readily extends to nonlinear problems, some of its variants which rely on the linearity of the PDE, such as frequency-domain POD or balanced POD, do not.

6.4.4 Treatment of moving discontinuities

One of the challenges in model reduction of transonic aerodynamics problems is the treatment of shocks. The fundamental challenge is that if $\mathbf{u}_h(t;\boldsymbol{\mu})$ contains a discontinuity whose location depends on $t \in \mathcal{I}$ or $\boldsymbol{\mu} \in \mathcal{P}$, then the Kolmogorov N-width of $\{\mathbf{u}_h(t;\boldsymbol{\mu})\}_{t\in\mathcal{I},\boldsymbol{\mu}\in\mathcal{P}}$ is large and the solution manifold is not amenable to a low-dimensional approximate of the form $u_N(\boldsymbol{\mu}) = \zeta^j \mathbf{u}_N^j(\boldsymbol{\mu})$. We provide a brief overview of methods developed to address the challenge. We restrict our coverage to methods tested for multidimensional aerodynamics problems, and refer to the references in [53] and a review paper [54] for a more general coverage.

Domain decomposition. One way to address the problem is to forgo the reduction of the state over the entire domain and to only reduce solution over a portion of the domain, as proposed for transonic Euler flows by LeGresley and Alonso [45]. Namely, we first decompose the domain into two regions: (i) region $\Omega_\mathrm{rom} \subset \Omega$ over which the solution varies smoothly and hence $\{u_h(\boldsymbol{\mu})\,|_{\Omega_\mathrm{rom}}\}_{\boldsymbol{\mu}\in\mathcal{P}}$ is amenable to model reduction, and (ii) region $\Omega_\mathrm{fom} \equiv \Omega \setminus \Omega_\mathrm{rom}$ which contains moving discontinuities and hence is not amenable to model reduction. We then approximate the solution $u_h(\boldsymbol{\mu})|_{\Omega_\mathrm{rom}}$ using a reduced basis $\{\zeta^j\,|_{\Omega_\mathrm{rom}}\}_{j=1}^N$ and $u_h(\boldsymbol{\mu})|_{\Omega_\mathrm{fom}}$ using the native basis of the FOM.

Nonlinear model reduction. Another approach to address moving discontinuities is to consider nonlinear model reduction. Here, nonlinear model reduction refers to approaches that approximate the solution in not a linear space V_N but in a nonlinear space. (Nonlinear model reduction should not be confused with linear model reduction of nonlinear PDEs, which has been considered so far in this section.) Nonlinear model reduction approaches considered by both Cagniart et al. [19] and Nair and Balajewicz [53] are based on the following observation: If the snapshots can be translated in space such that the shocks are aligned, then the snapshots can be effectively compressed using a linear model reduction technique (e. g., POD). Specifically, the approach approximates the solution $u_h(\cdot;\boldsymbol{\mu}) \in V_h$ by

$$u_N(x;\boldsymbol{\mu}) = \zeta^j(x;\boldsymbol{\mu})\mathbf{u}_N^j(\boldsymbol{\mu})$$

for some $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ and a parameter-dependent basis

$$\zeta^j(x;\boldsymbol{\mu}) = u_h(y_j(x,\boldsymbol{\mu});\boldsymbol{\mu}), \quad j = 1,\dots,N,$$

where $y_j : \Omega \times \mathcal{P} \to \mathbb{R}^d$, $j = 1,\dots,N$, are parameter-dependent translation functions. The translation functions $\{y_j\}$ are trained in the offline stage such that the shock locations for the translated basis $\zeta^j(\cdot;\boldsymbol{\mu}) = u_h(y_j(\cdot,\boldsymbol{\mu}))$ are (approximately) aligned with the shock in $u_h(\cdot;\boldsymbol{\mu})$. Nonlinear approximation of shocks is a relatively new development in the field of model reduction, and hence we refer to [19, 53, 54] and references therein for specific implementations. The nonlinear model reduction approach has been applied to transonic Euler over an airfoil [19] and supersonic forward step [53].

6.4.5 Large-scale applications

We conclude this section with a few examples of model reduction applied to large-, industry-scale nonlinear aerodynamics problems.

- *Unsteady turbulent flow past Amhed body* [22]. In this work Carlberg et al. consider model reduction of nonparameterized turbulent flow over the Ahmed body modeled by detached eddy simulation. The FOM consists of $N_h \approx 1.7 \times 10^7$ spatial degrees of freedom. The FOM is hyperreduced using the GNAT method; the resulting ROM uses a reduced basis of the size $N = 283$ for the state, a reduced basis of the size $N_R = N_J = 1{,}514$ for the residual and Jacobian, and $\tilde{N}_e = 378$ sample nodes. The ROM reproduces the unsteady drag time history with less than 1 % discrepancy. The FOM requires 13 hours using 512 cores, whereas the ROM requires 3.9 hours using 4 cores; the ROM reduces the computational cost by a factor of 438.
- *Parametric shape deformation of the NASA Common Research Model* [69]. In this work Washabaugh et al. consider model reduction of steady RANS-SA flow over the NASA Common Research Model under parametric shape deformation. The FOM consists of $N_h \approx 6.8 \times 10^7$ degrees of freedom and is parameterized by four

shape parameters: wingspan, washout, streamwise wingtip rake, and vertical wingtip rake. The ROM based on the minimum-residual formulation with the gappy POD collocation hyperreduction uses $N = 23$ modes and $\tilde{N}_e = 5000$ sample nodes. The ROM achieves less than 0.3 % error in drag for test parameters considered. A single simulation of the FOM requires 2 hours using 1024 cores, whereas the ROM requires 2.8 minutes on a laptop.

6.5 Summary and conclusions

In this chapter, we surveyed model reduction techniques for linearized and nonlinear aerodynamics problems that have been developed in the past two decades. We discussed essential ingredients of model reduction, with an emphasis on techniques that are designed to address challenges in aerodynamics, including convection dominance, nonlinearity, limited stability, limited regularity, and a wide range of scales. We also reviewed successful applications of model reduction to large-scale industry-relevant aerodynamics problems to date. There still exist many open challenges to model reduction of complex aerodynamics problems. Their industrial relevance and challenging nature make them arguably an ideal testbed to develop and assess the next generation of model reduction algorithms.

Bibliography

[1] R. Abgrall and R. Crisovan, Model reduction using L1-norm minimization as an application to nonlinear hyperbolic problems, *Int. J. Numer. Methods Fluids*, **87** (12) (2018), 628–651.

[2] D. Amsallem, J. Cortial, and C. Farhat, Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information, *AIAA J.*, **48**(9) (2010), 2029–2037.

[3] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat, Model predictive control under coupled fluid-structure constraints using a database of reduced-order models on a tablet. *AIAA* 2013-2588, AIAA, 2013.

[4] D. Amsallem and C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, *AIAA J.*, **46** (7) (2008), 1803–1813.

[5] D. Amsallem and C. Farhat, Stabilization of projection-based reduced-order models, *Int. J. Numer. Methods Eng.*, **91** (4) (2012), 358–377.

[6] J. S. R. Anttonen, P. I. King, and P. S. Beran, POD-based reduced-order models with deforming grids, *Math. Comput. Model.*, **38** (1–2) (2003), 41–62.

[7] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, Unified analysis of discontinuous Galerkin methods for elliptical problems, *SIAM J. Numer. Anal.*, **39** (5) (2002), 1749–1779.

[8] P. Astrid, S. Weiland, K. Willcox, and T. Backx, Missing point estimation in models described by proper orthogonal decomposition, *IEEE Trans. Autom. Control*, **53** (10) (2008), 2237–2251.

[9] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist, Stable Galerkin reduced order models for linearized compressible flow, *J. Comput. Phys.*, **228** (6) (2009), 1932–1946.

[10] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Acad. Sci. Paris, Ser. I*, **339** (2004), 667–672.

[11] T. Barth and P. Charrier, *Energy Stable Flux Formulas for the Discontinuous Galerkin Discretization of First-Order Nonlinear Conservation Laws*, NASA Technical Report NAS-01-001, NAS, 2001.

[12] T. J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in D. Kröner, M. Ohlberger, and C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, pp. 195–282, Springer-Verlag, 1999.

[13] T. J. Barth and M. G. Larson, *A Posteriori Error Estimates for Higher Order Godunov Finite Volume Methods on Unstructured Meshes*, Technical report, Complex Applications III, R. Herbin and D. Kroner (eds.), HERMES Science Publishing Ltd, 2002.

[14] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numer.*, **10** (2001), 1–102.

[15] A. N. Brooks and T. J. R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.*, **32** (1–3) (1982), 199–259.

[16] T. Bui-Thanh, M. Damodaran, and K. Willcox, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. *AIAA* 2003-4213, AIAA, 2003.

[17] T. Bui-Thanh, K. Willcox, and O. Ghattas, Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications, *AIAA J.*, **46** (10) (2008), 2520–2529.

[18] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, *J. Comput. Phys.*, **224** (2) (Jun 2007), 880–896.

[19] N. Cagniart, R. Crisovan, Y. Maday, and R. Abgrall, Model order reduction for hyperbolic problems: a new framework, *<hal-01583224>*, 2017.

[20] K. Carlberg, M. Barone, and H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, *J. Comput. Phys.*, **330** (2017), 693–734.

[21] K. Carlberg, C. Bou-Mosleh, and C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *Int. J. Numer. Methods Eng.*, **86** (2) (2011), 155–181.

[22] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, *J. Comput. Phys.*, **242** (2013), 623–647.

[23] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via Discrete Empirical Interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[24] B. Cockburn, Discontinuous Galerkin methods, *Z. Angew. Math. Mech.*, **83** (11) (2003), 731–754.

[25] M. J. de C. Henshaw, K. J. Badcock, G. A. Vio, C. B. Allen, J. Chamberlain, I. Kaynes, G. Dimitriadis, J. E. Cooper, M. A. Woodgate, A. M. Rampurawala, D. Jones, C. Fenwick, A. L. Gaitonde, N. V. Taylor, D. S. Amor, T. A. Eccles, and C. J. Denley, Non-linear aeroelastic prediction for aircraft applications, *Prog. Aerosp. Sci.*, **43** (4–6) (2007), 65–137.

[26] E. H. Dowell and K. C. Hall, Modeling of fluid-structure interaction, *Annu. Rev. Fluid Mech.*, **33** (1) (2001), 445–490.

[27] J. L. Eftang, A. T. Patera, and E. M. Rønquist, An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations, *SIAM J. Sci. Comput.*, **32** (6) (2010), 3170–3200.

[28] B. I. Epureanu, A parametric analysis of reduced order models of viscous flows in turbomachinery, *J. Fluids Struct.*, **17** (7) (2003), 971–982.

[29] R. Everson and L. Sirovich, Karhunen–Loève procedure for gappy data, *J. Opt. Soc. Am. A, Opt. Image Sci.*, **12** (8) (1995), 1657–1664.
[30] T. Franz, R. Zimmermann, S. Görtz, and N. Karcher, Interpolation-based reduced-order modelling for steady transonic flows via manifold learning, *Int. J. Comput. Fluid Dyn.*, **28** (3–4) (2014), 106–121.
[31] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *ESAIM: M2AN*, **41** (3) (2007), 575–605.
[32] B. Haasdonk and M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, *Math. Model. Numer. Anal.*, **42** (2) (2008), 277–302.
[33] K. C. Hall, Eigenanalysis of unsteady flows about airfoils, cascades, and wings, *AIAA J.*, **32** (12) (1994), 2426–2432.
[34] K. C. Hall, J. P. Thomas, and E. H. Dowell, Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows, *AIAA J.*, **38** (10) (2000), 1853–1862.
[35] A. Harten, On the symmetric form of systems of conservation laws with entropy, *J. Comput. Phys.*, **49** (1) (1983), 151–164.
[36] S. Hovland, J. T. Gravdahl, and K. E. Willcox, Explicit model predictive control for large-scale systems via model reduction, *J. Guid. Control Dyn.*, **31** (4) (2008), 918–926.
[37] T. J. Hughes, L. P. Franca, and G. M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations, *Comput. Methods Appl. Mech. Eng.*, **73** (2) (1989), 173–189.
[38] F. T. Johnson, E. N. Tinoco, and N. J. Yu, Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle, *Comput. Fluids*, **34** (10) (2005), 1115–1151.
[39] I. Kalashnikova and M. F. Barone, On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment, *Int. J. Numer. Methods Eng.*, **83** (10) (2010), 1345–1375.
[40] I. Kalashnikova, M. F. Barone, and M. R. Brake, A stable Galerkin reduced order model for coupled fluid-structure interaction problems, *Int. J. Numer. Methods Eng.*, **95** (2) (2013), 121–144.
[41] C. T. Kelley and D. E. Keyes, Convergence analysis of pseudo-transient continuation, *SIAM J. Numer. Anal.*, **35** (2) (1998), 508–523.
[42] T. Kim, Frequency-domain Karhunen–Loeve method and its application to linear dynamic systems, *AIAA J.*, **36** (11) (1998), 2117–2123.
[43] P. LeGresley and J. Alonso, Airfoil design optimization using reduced order models based on proper orthogonal decomposition, *AIAA* 2000-2545, AIAA, 2000.
[44] P. A. LeGresley and J. J. Alonso, Investigation of non-linear projection for POD based reduced order models for aerodynamics, *AIAA* 2001–0926, AIAA, 2001.
[45] P. A. LeGresley and J. J. Alonso, Dynamic domain decomposition and error correction for reduced order models, *AIAA* 2003-250, AIAA, 2003.
[46] T. Lieu and C. Farhat, Adaptation of aeroelastic reduced-order models and application to an F-16 configuration, *AIAA J.*, **45** (6) (Jun 2007), 1244–1257.
[47] T. Lieu, C. Farhat, and M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, *Comput. Methods Appl. Mech. Eng.*, **195** (41–43) (2006), 5730–5742.
[48] T. Lieu and M. Lesoinne, Parameter adaptation of reduced order models for three-dimensional flutter analysis, *AIAA* 2004-888, AIAA, 2004.
[49] D. J. Lucia, P. S. Beran, and W. A. Silva, Reduced-order modeling: new approaches for computational physics, *Prog. Aerosp. Sci.*, **40** (1–2) (2004), 51–117.
[50] Y. Maday, A. T. Patera, and D. V. Rovas, A blackbox reduced-basis output bound method for noncoercive linear problems, in *Nonlinear Partial Differential Equations and their Applications – Collège de France Seminar Volume XIV*, pp. 533–569, Elsevier, 2002.

[51] D. Mavriplis, D. Darmofal, D. Keyes, and M. Turner, Petaflops opportunities for the NASA fundamental aeronautics program, *AIAA* 2007-4084, AIAA, 2007.
[52] B. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, *IEEE Trans. Autom. Control*, **26** (1) (1981), 17–32.
[53] N. J. Nair and M. Balajewicz, Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks, *Int. J. Numer. Methods Eng.*, **117** (12) (2019), 1234–1262.
[54] M. Ohlberger and S. Rave, Reduced basis methods: success, limitations and future challenges, in *Proceedings of the Conference Algoritmy*, pp. 1–12, 2016.
[55] M. Romanowski, Reduced order unsteady aerodynamic and aeroelastic models using Karhunen–Loeve eigenmodes, *AIAA* 1996-3981, AIAA, 1996.
[56] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *Int. J. Bifurc. Chaos*, **15** (03) (2005), 997–1013.
[57] G. Rozza, D. B. P. Huynh, and A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations – Application to transport and continuum mechanics, *Arch. Comput. Methods Eng.*, **15** (3) (2008), 229–275.
[58] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, *J. Comput. Phys.*, **202** (1) (2005), 346–366.
[59] R. Schmit and M. Glauser, Improvements in low dimensional tools for flow-structure interaction problems: using global POD, *AIAA* 2004-889, AIAA, 2004.
[60] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, *Q. Appl. Math.*, **45** (3) (1987), 561–571.
[61] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*, Nasa/cr-2014-218178, NASA, 2014.
[62] P. R. Spalart and S. R. Allmaras, A one-equation turbulence model for aerodynamics flows, *Rech. Aérosp.*, **1** (1994), 5–21.
[63] J. A. Taylor and M. N. Glauser, Towards practical flow sensing and control via POD and LSE based low-dimensional tools, *J. Fluids Eng.*, **126** (3) (2004), 337.
[64] J. P. Thomas, E. H. Dowell, and K. C. Hall, Three-dimensional transonic aeroelasticity using proper orthogonal decomposition-based reduced-order models, *J. Aircr.*, **40** (3) (2003), 544–551.
[65] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, 2009.
[66] A. Vendl, H. Faßbender, S. Görtz, R. Zimmermann, and M. Mifsud, Model order reduction for steady aerodynamics of high-lift configurations, *CEAS Aeronaut. J.*, **5** (4) (2014), 487–500.
[67] K. Veroy, C. Prud'homme, D. Rovas, and A. Patera, A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, *AIAA* 2003-3847, AIAA, 2003.
[68] K. Washabaugh, D. Amsallem, M. Zahr, and C. Farhat, Nonlinear model reduction for CFD problems using local reduced-order bases, *AIAA* 2012-2686, AIAA, (2012).
[69] K. Washabaugh, M. J. Zahr, and C. Farhat, On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries, *AIAA* 2016-1814, AIAA, 2016.
[70] K. M. Washabaugh, *A Scalable Model Order Reduction Framework for Steady Aerodynamic Design Applications*. PhD thesis, Stanford University, 2016.
[71] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, *Comput. Fluids*, **35** (2) (2006), 208–226.

[72] K. Willcox and A. Megretski, Fourier series for accurate, stable, reduced-order models in large-scale linear applications, *SIAM J. Sci. Comput.*, **26** (3) (2005), 944–962.
[73] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA J.*, **40** (11) (2002), 2323–2330.
[74] K. Willcox, J. Peraire, and J. White, An Arnoldi approach for generation of reduced-order models for turbomachinery, *Comput. Fluids*, **31** (3) (2002), 369–389.
[75] M. Yano, Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws, *Adv. Comput. Math.*, **45** (5–6) (2019), 2287–2320.
[76] M. Yano, Goal-oriented model reduction of parametrized nonlinear PDEs; application to aerodynamics, *Int. J. Numer. Methods Eng.*, accepted, 2020.
[77] M. Yano and A. T. Patera, An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs, *Comput. Methods Appl. Mech. Eng.*, **344** (2019), 1104–1123.
[78] M. J. Zahr and C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1111–1135.
[79] R. Zimmermann and S. Görtz, Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model, *Aeronaut. J.*, **116** (1184) (2012), 1079–1100.
[80] R. Zimmermann and S. Görtz, Non-linear reduced order models for steady aerodynamics, *Proc. Comput. Sci.*, **1** (1) (2012), 165–174, ICCS 2010.

Bülent Karasözen

7 Model order reduction in neuroscience

Abstract: The human brain contains approximately 10^9 neurons, each with approximately 10^3 connections, synapses, with other neurons. Most sensory, cognitive, and motor functions of our brains depend on the interaction of a large population of neurons. In recent years, many technologies have been developed for recording large numbers of neurons either sequentially or simultaneously. Increases in computational power and algorithmic developments have enabled advanced analyses of the neuronal population parallel to the rapid growth of quantity and complexity of the recorded neuronal activity. Recent studies made use of dimensionality and model order reduction techniques to extract coherent features which are not apparent at the level of individual neurons. It has been observed that the neuronal activity evolves on low-dimensional subspaces. The aim of model reduction of large-scale neuronal networks is the accurate and fast prediction of patterns and their propagation in different areas of the brain. Spatiotemporal features of brain activity are identified on low-dimensional subspaces with methods such as dynamic mode decomposition, proper orthogonal decomposition, the discrete empirical interpolation method, and combined parameter and state reduction. In this chapter, we give an overview of the currently used dimensionality reduction and model order reduction techniques in neuroscience.

Keywords: neuroscience, dimensionality reduction, proper orthogonal decomposition, discrete empirical interpolation, dynamic mode decomposition, state and parameter estimation

MSC 2010: 93A15, 92C55, 37M10, 37M99, 37N40, 65R32

7.1 Introduction

Due to the advances in recording and imaging technologies, the number of recorded signals from brain cells increased significantly in the last few years. The recorded spatio-temporal neural activity gives rise to networks with complex dynamics. In neuroscience, molecular and cellular level details are incorporated in large-scale models of the brain in order to reproduce phenomena such as learning and behavior. The rapid growth of simultaneous neuronal recordings in scale and resolution brings challenges with the analysis of the neuronal population activity. New computational approaches have to be developed to analyze, visualize, and understand large-scale recordings of neural activity. While algorithmic developments and the availability of significantly

Bülent Karasözen, Department of Mathematics & Institute of Applied Mathematics, Middle East Technical university, Ankara, Turkey

https://doi.org/10.1515/9783110499001-007

more computing power have enabled analysis of larger neuronal networks, these advances cannot keep pace with the increasing size and complexity of recorded activity. The activity of complex networks of neurons can often be described by relatively few distinct patterns. Model order reduction techniques enable us to identify the coherent spatio-temporal patterns.

The presence or absence of a neural mechanism can be analyzed for neuronal populations. Dimensionality reduction methods [6] are data-driven statistical techniques for forming and evaluating hypotheses about population activity structure, which are summarized in Section 7.2. One of the goals of neuroscience is the fast and accurate prediction of the potential propagation in neurons. The differential equations describing the propagation of potential in neurons have been developed and are described by Hodgkin and Huxley equations [12]. They consist of a coupled system of ordinary and partial differential equations (ODEs and PDEs). The dimension of the associated discretized systems is very large for accurately simulating neurons with realistic morphological structure and synaptic inputs. In Section 7.3 we present two model order reduction approaches based on proper orthogonal decomposition (POD) and the discrete empirical interpolation method (DEIM) [5], which can predict accurately the potential propagation in large-scale neuronal networks leading to important speedups [17, 16, 2]. Using the functional neuroimaging data from electroencephalography (EEG) or functional magnetic resonance imaging (fMRI), different regions of the brain can be inferred by dynamic causal modeling (DCM) [7]. Effective connectivity is parameterized in terms of coupling among unobserved brain states and neuronal activity in different regions. In Section 7.4 we describe a combined state and parameter reduction for parameter estimation and identification [10] to extract effective connectivity in neuronal networks from measured data, such as EEG or fMRI data. In Section 7.5 the data-driven, equation-free model order reduction method dynamic mode decomposition (DMD) is described for identifying sleep spindle networks [3]. Reduced-order models (ROMs) with POD and the DEIM and four variants of them are presented for neuronal synaptic plasticity and neuronal spiking networks in Section 7.6.

7.2 Dimensionality reduction methods

Coordination of responses across neurons exists only at the level of the population and not at the level of single neurons. The presence or absence of a neural mechanism can be analyzed for neuronal populations. Dimensionality reduction methods are data-driven statistical techniques for forming and evaluating hypotheses about population activity structure. They produce low-dimensional representations of high-dimensional data with the aim to extract coherent patterns which preserve or highlight some feature of interest in the data [6]. The recorded neurons of dimension D are likely

not independent of each other, because they belong to a common network of neuronal populations. From the high-dimensional data of neuronal recordings, a smaller number of explanatory variables K ($K < D$) are extracted with the help of dimensionality reduction methods. The explanatory variables are not directly observed; therefore they are referred to as latent variables. The latent variables define a K-dimensional space representing coherent patterns of the noisy neural activity of D neurons.

There exist several dimensionality reduction methods which differ in the statistical interpretation of the preserved and discarded features of the neuronal populations. We summarize the commonly used statistical methods of dimensionality reduction methods in [6], where further references about the methods can be found.

Principal component and factor analysis. The most widely known technique to extract coherent patterns from high-dimensional data is modal decomposition. A particularly popular modal decomposition technique is principal component analysis (PCA), which derives modes ordered by their ability to account for energy or variance in the data. In particular, PCA is a static technique and does not model temporal dynamics of time-series data explicitly, so it often performs poorly in reproducing dynamic data, such as recordings of neural activity. The low-dimensional space identified by PCA captures variance of all types, including firing rate variability and spiking variability, whereas factor analysis discards the independent variance for each neuron. and preserves variance that is shared across neurons.

Time series methods. The temporal dynamics of the population activity can be identified if the data come from a time series. The most commonly used time series methods for dimensionality reduction neural recordings are hidden Markov models (HMMs) [18], kernel smoothing followed by a static dimensionality reduction, Gaussian process factor analysis [35], latent linear dynamical systems [4], and latent nonlinear dynamical systems [26]. They produce latent neural trajectories that capture the shared variability across neurons. The HMM is applied when a jump between discrete states of neurons exists, other methods identify smooth changes in firing rates over time.

Methods with dependent variables. On many neuronal recordings the high-dimensional firing rate space is associated with labels of one or more dependent variables, like stimulus identity, decision identity, or a time index. The dimensionality reduction aims in this case to project the data such that differences in these dependent variables are preserved. Linear discriminant analysis can be used to find a low-dimensional projection in which the G groups to which the data points belong are well separated.

Nonlinear dimensionality reduction methods. All the previous methods assume a linear relationship between the latent and observed variables. When the data lie on a low-dimensional, nonlinear manifold in the high-dimensional space, a linear method may require more latent variables than the number of true dimensions of the data. The most frequently used methods to identify nonlinear manifolds are Isomap79 [31] and locally linear embedding [28]. Because the nonlinear methods use local neighborhoods to estimate the structure of the manifold, population responses may not evenly

explore the high-dimensional space. Therefore theses methods should be used with care.

7.3 Proper orthogonal decomposition and discrete empirical interpolation for the Hodgkin–Huxley model

One of the goals of neuroscience is the fast and accurate prediction of the potential propagation in neurons. The differential equations describing propagation of potential in neurons are described by Hodgkin–Huxley (HH) cable equations [12]. They consist of a coupled system of ODEs and PDEs. Accurate simulation of morphology, kinetics, and synaptic inputs of neurons requires solution of large systems of nonlinear ODEs. The complexity of the models is determined by the synapse density and the dentritic length. In simulations, for one synapse per micron on a cell with a dendrite of 5 mm, 5,000 compartments each with 10 variables are needed, which results in 50,000 coupled nonlinear system of ODEs [17, 16]. To recover complex dynamics, efficient reduced-order neuronal methods are developed using POD and the DEIM from the snapshots of the in space and time discretized coupled PDEs and ODEs [17, 16, 2]. In this section we describe two of them. They differ in the formulation of the HH cable equation and of the equations for the gating variables.

7.3.1 Morphologically accurate reduced-order modeling

The neuronal full-order models (FOMs) in [17, 16] consist of D branched dendritic neurons $B = \sum_{d=1}^{D} B_d$ meeting at the soma, where the d-th neuron has B_d branches. It is assumed that the branch b carries C distinct ionic currents with associated densities and $G_{bc}(x)$ and reversal potentials E_c, $c = 1, \ldots, C$. The kinetics of current c on branch b are governed by the F_c gating variables, w_{bcf}, $f = 1, \ldots, F_c$. When subjected to input at S_b synapses, the nonlinear HH cable equation for the transmembrane potential $v_b(x,t)$ with the equation for the gating variables w_{bcf} is given by (see [2], Figure 1, model network with three cables)

$$\begin{aligned} a_b C_m \frac{\partial v_b}{\partial t} = {} & \frac{1}{2R_i}\frac{\partial}{x}\left(a_b^2 \frac{\partial v_b}{\partial x}\right) \\ & - a_b \sum_{c=1}^{C} G_{bc}(x)(v_b - E_c) \prod_{f=1}^{F_c} w_{bcf}^{q_{cf}} \\ & \frac{1}{2\pi} \sum_{s=1}^{S_b} g_{bs}(t)\delta(x - x_{bs})(v_b - E_{bs}), \end{aligned} \tag{7.1}$$

$$\frac{\partial w_{bcf}}{\partial t} = \frac{w_{cf,\infty}(v_b) - w_{bcf}}{\tau_{cf}(v_b)}, \quad 0 < x < l_b,\ t > 0, \tag{7.2}$$

where $g_{bs}(nS)$ is the time course, x_{bs} is the spatial location, and E_{bs} is the reversal potential of the s-th synapse on branch b. The variables and parameters in (7.1) are described in [17, 16].

These branch potentials interact at J junction points, where junction J denotes the soma. The D dendrites join at the soma. Continuity of the potential at the soma leads to a common value at the current balance denoted by $v_\sigma(t)$. Then the networked form of (7.1) becomes

$$\begin{aligned} a_b C_m \frac{\partial v_\sigma}{\partial t} = {} & \frac{\pi}{A_\sigma R_i} \sum_{d=1}^{D} \frac{\partial}{\partial x}\left(a^2_{b_j^d}(l_{b_j^d}) \frac{\partial v_{b_{jd}}(l_{b_{jd}}, t)}{\partial x} \right) \\ & - a_b \sum_{c=1}^{C} G_{\sigma c}(x)(v_\sigma - E_c) \prod_{f=1}^{F_c} w_{\sigma cf}^{q_{cf}}(t) \\ & \frac{1}{A_\sigma} \sum_{s=1}^{S_b} g_{\sigma s}(t)(v_\sigma(t) - E_{\sigma s}), \end{aligned} \tag{7.3}$$

$$\frac{\partial w_{\sigma cf}(t)}{\partial t} = \frac{w_{cf,\infty}(v_\sigma(t)) - w_{\sigma cf}(t)}{\tau_{cf}(v_\sigma)(t)}, \quad 0 < x < l_b,\ t > 0. \tag{7.4}$$

When the cell is partitioned into N compartments, with C distinct ionic currents per compartment and with F gating variables per current, the following nonlinear ODEs are obtained:

$$\begin{aligned} v'(t) = {} & Hv(t) - \big(\Phi(w(t))e\big).v(t) + \Phi(w(t))E_i \\ & + G(t).\big(v(t) - E_s\big), \quad v(t) \in \mathbb{R}^N, \end{aligned} \tag{7.5}$$

$$w'(t) = \big(A(v(t)) - w(t)\big)./B(v(t)), \quad w(t) \in \mathbb{R}^{N \times C \times F}, \tag{7.6}$$

where $H \in \mathbb{R}^{N \times N}$ is the Hines matrix [11], $e = [1\ 1 \cdots 1]^T \in \mathbb{R}^C$ and the "dot" operator, $a.b$, denotes elementwise multiplication; E_i and E_s are the vector of channel reversal potentials and the vector of synaptic reversal potentials, respectively. Equation (7.5) is discretized in time by the second-order discretized Euler scheme [11].

In [16] using the snapshots of $v(t)$ and of the nonlinear term $N(v(t), w(t)) \equiv \big(\Phi(w(t))e\big).v(t) - \Phi(w(t))E_i$ at times $t_1, t_2, \ldots, t_n$ the POD and DEIM modes are constructed.

The reduced membrane potentials v_r are constructed using the POD basis, and the reduced gating variables w_r are obtained after applying the DEIM to the nonlinear terms. The ROM in [16] preserves the spatial precision of synaptic input and captures accurately the subthreshold and spiking behaviors.

In [17] a linearized quasi-active reduced neuronal model is constructed using balanced truncation and $\mathcal{H}_2$-approximation of transfer functions in time. ROMs preserve the input-output relationship and reproduce only subthreshold dynamics.

7.3.2 Energy-stable neuronal reduced-order modeling

In [1, 2] a different form of the HH cable equation and ODEs for gating variables is considered. The intracellular potential $v(x,t)$ and three gating variables $m(x,t)$, $h(x,t)$, and $n(x,t)$ describe the activation and deactivation of the ion channels, of the sodium channels, and of the potassium channels, respectively. A single cable in the computational domain $(x,t) \in [0,L] \times (0,T]$ describing the distribution of the potential $u(x,t)$ is given by [1, 2]

$$\frac{\partial u}{\partial t} = \frac{\mu}{a(x)}(a(x)^2 u_x)_x - \frac{1}{C_m} g(m,h,n)u + \frac{1}{C_m} f(m,h,n,x,t), \tag{7.7}$$

where $\mu = \frac{1}{2C_m R_i} > 0$, $a(x)$ is the radius of the neurons, C_m is the specific membrane capacitance, and R_i is the axial resistivity. The conductance $g(x,t)$ is a polynomial of the gating variables

$$g(x,t) = g_1 m^3 h + g_2 n^4 + g_3 > 0, \tag{7.8}$$

with the source term

$$f(m,h,n,x,t) = g_1 E_1 m^2 h + g_2 E_2 n^4 + g_3 E_3 - i(x,t), \tag{7.9}$$

where E_l, $l = 1,2,3$, are equilibrium potentials and $i(x,t)$ is the input current at x,

$$i(x,t) = \sum_{s=1}^{N_s} i_s(x,t), \quad x \in [0,L]. \tag{7.10}$$

The nonlinear ODEs for the gating variables are given by

$$\begin{aligned} \frac{\partial m}{\partial t} &= \alpha_m(v(x,t)(1-m(x,t)) - \beta_m v(x,t))m(x,t), \\ \frac{\partial h}{\partial t} &= \alpha_h(v(x,t))(1-h(x,t)) - \beta_h v(x,t))h(x,t), \\ \frac{\partial n}{\partial t} &= \alpha_n(v(x,t))(1-n(x,t)) - \beta_n v(x,t))n(x,t). \end{aligned} \tag{7.11}$$

Expressions for α_m, α_h, α_n, β_m, β_h, β_n and boundary conditions can be found in [2].

In [1, 2], a model network with three cables connected to a soma is used. The equations governing the potential propagation in a network of N_c neuron cables-dendrites and/or axons with the superscript $^{(c)}$, $c = 1, \dots N_c$, are given as

$$\begin{aligned} \frac{\partial v^{(c)}}{\partial t} = &\frac{\mu}{a^{(c)}(x^{(c)})}((a^{(c)}(x^{(c)})^2)v_x^{(c)})_x - \frac{1}{C_m} g(m^{(c)}, h^{(c)}, n^{(c)})u^{(c)} \\ &+ \frac{1}{C_m} f(m^{(c)}, h^{(c)}, n^{(c)}, x^{(c)}, t), \end{aligned} \tag{7.12}$$

$$\begin{aligned}
\frac{\partial m^{(c)}}{\partial t} &= \alpha_m(v^{(c)}(1-m^{(c)}) - \beta_m v^{(c)})m^{(c)},\\
\frac{\partial h^{(c)}}{\partial t} &= \alpha_h(v^{(c)})(1-h^{(c)}) - \beta_h v(c))h^{(c)},\\
\frac{\partial n^{(c)}}{\partial t} &= \alpha_n(v^{(c)}))(1-n^{(c)}) - \beta_n v^{(c)})n^{(c)},
\end{aligned} \tag{7.13}$$

for $x^{(c)} \in \Omega^{(c)} = [0, L^{(c)}]$ together with the boundary conditions.

The semi-discrete forms of these equations are approximated using energy-stable summation by parts [1, 2] for the model network. The reduced-order bases (ROBs) for multiple cables of identical lengths are assembled into a network in block form. The block structure of the ROB allows a flexible structure-preserving model reduction approach with an independent approximation in each cable and energy stability and accuracy properties follow from this block structure. Computation of the time-varying reduced variables in the gating equations at every time t is costly because they scale with the dimension of the FOM. A nonnegative variant of the DEIM, nonnegative DEIM (NNDEIM), is developed in [2] to preserve the structure and energy stability properties of the equations.

The capability of the greedy-based approach to generate accurate predictions in large-scale neuronal networks is demonstrated for systems with more than 15,000 degrees of freedom. The state variable ROB of dimension $l = 15$ with POD modes and the nonnegative ROBs of dimension $p = 60$ with NNDEIM modes are constructed using a greedy approach to predict the potential variation at the soma. The speedup factor of simulations is about 20, which is larger than that of Galerkin projection, which is only 1.3 without the NNDEIM.

7.4 Combined state and parameter reduction for dynamic causal modeling

In neuroscience, different regions of the brain are inferred using neuroimaging data from EEG or fMRI recordings using the method of DCM [7]. Effective connectivity is parameterized in terms of coupling among unobserved brain states and neuronal activity in different regions. In DCM the neuronal activity of the observed brain region is represented as a single-input single-output (SISO) system

$$\dot{x} = A_{\text{dyn}}(\mu)x + B_{\text{dyn}}u, \tag{7.14}$$

with the parameterized connectivity $A_{\text{dyn}}(\mu)$ and external input matrices B_{dyn}.

Linearization of the nonlinear DCM hemodynamic forward submodel (balloon model) [7] transforms the neuronal activity to the measured blood oxygen level-dependent (BOLD) response. Linearization around the equilibrium results in the

following SISO system:

$$B_{\text{obs}} := (1\ 0\ 0\ 0)^T, \quad C_{\text{obs}} = (0\ 0\ V_0k_1\ V_0k_2), \tag{7.15}$$

$$\dot{z}_i = A_{\text{obs}}z_i + B_{\text{obs}}x_i, \tag{7.16}$$

$$y_i = C_{\text{obs}}z_i, \tag{7.17}$$

$$z_0 = (0\ 0\ 0\ 0)^T, \tag{7.18}$$

$$A_{\text{obs}} := \begin{pmatrix} \frac{1}{\tau_s} & \frac{1}{\tau_f} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\tau_0 E_0}(1-(1-E_0)(1-\ln(1-E_0))) & \frac{1}{\tau_0} & \frac{1-\alpha}{\tau_0\alpha} \\ 0 & \frac{1}{\tau_0} & 0 & \frac{1}{\tau_0\alpha} \end{pmatrix}. \tag{7.19}$$

The fMRI measurements at the i-th brain region are reflected by the output variables y_i. For the meaning of the variables and parameters in (7.15) and (7.19) we refer to [10, 9]. The linearized forward submodels are embedded into the fMRI connectivity model

$$\begin{pmatrix} \dot{x} \\ \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ z_{N_{\text{dyn}}} \end{pmatrix} = \begin{pmatrix} A_{\text{dyn}}(\mu) & 0 & 0 & \cdots & 0 \\ \delta_{1,1} & A_{\text{obs}} & 0 & & \\ \delta_{2,1} & 0 & A_{\text{obs}} & & \\ \vdots & & \ddots & & \\ \delta_{1,N_{\text{dyn}}} & & & A_{\text{obs}} & \end{pmatrix} \begin{pmatrix} x \\ z_1 \\ z_2 \\ \vdots \\ z_{N_{\text{dyn}}} \end{pmatrix} + \begin{pmatrix} B_{\text{dyn}} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} v, \tag{7.20}$$

$$y = \begin{pmatrix} 0 \begin{pmatrix} C_{\text{obs}} & & \\ & \ddots & \\ & & C_{\text{obs}} \end{pmatrix} \end{pmatrix} \begin{pmatrix} x \\ z_1 \\ z_2 \\ \vdots \\ z_{N_{\text{dyn}}} \end{pmatrix}, \tag{7.21}$$

where $\delta_{ij} \in \mathbb{R}^{4\times N_{\text{dyn}}}$ denotes the Kronecker matrix with nonzero elements located at the (i,j)-th component.

The linearized state-space forward model (7.20) and (7.21) corresponds to a multiple-input multiple-output (MIMO) system,

$$\dot{x}(t) = A(\mu)x(t) + Bu(t), \quad y(t) = Cx(t), \tag{7.22}$$

where $x \in \mathbb{R}^N$ is the internal state, $u \in \mathbb{R}^J$ is the external input, $y \in \mathbb{R}^O$ is the observed output, and μ are the parameters describing different conditions.

For large numbers of $M := N^2$ parameters, the computational cost for inferring the parameters and states is very high. In [10, 8] a combined state and parameter model order reduction is developed for parameter estimation and identification to extract effective connectivity. The inversion procedure consists of two phases, the offline and online phases. In the offline phase, the underlying parameterized model is reduced jointly in states and parameters. In the online phase, the ROM's parameters are estimated to fit the observed experimental data. Using state and parameter reduction, the computational cost is reduced in the offline phase. The simultaneous reduction of state and parameter space is based on Galerkin projections with the orthogonal matrices for the state $V \in \mathbb{R}^{N\times n}$ and for the parameters $P \in \mathbb{R}^{M\times m}$. The reduced model is of lower order $n \ll N$, $m \ll M$ than the original FOM. The reduced states $x_r(t) \in \mathbb{R}^n$ and the reduced parameters $\mu \in \mathbb{R}^m$ are computed as

$$\dot{x}_r(t) = A_r(\mu_r)x_r(t) + B_r u(t), \quad y_r(t) = C_r x(t), \tag{7.23}$$

with a reduced initial condition $x_{r,0} = V^T x_0$ and the reduced components

$$\begin{aligned} \mu_r &= P^T \mu \in \mathbb{R}^m, \\ A_r(\mu_r) &= V^T A(P\mu_r)V \in \mathbb{R}^{n\times n}, \\ B_r &= V^T B \in \mathbb{R}^{n\times J}, \\ C_r &= CV \in \mathbb{R}^{O\times m}. \end{aligned}$$

In the online phase, the optimization-based inverse problem is combined with the reduction of state and parameter space. The inversion is based on a generalized data-driven optimization approach to construct the ROMs in [23] and enhanced with the Monte Carlo method to speed up the computations. The state projection $V \in \mathbb{R}^{N\times n}$ and parameter projection $P \in \mathbb{R}^{m\times m}$ are determined iteratively based on a greedy algorithm by maximizing the error between the high-fidelity original and the low-dimensional reduced model in the Bayesian setting.

Numerical experiments with the DCM model in [23] show the highly dimensional neuronal network system is efficiently inverted due to the short offline durations. In the offline phase, Monte Carlo-enhanced methods require more than an order of magnitude less offline time compared to the original and data misfit-enhanced methods. In the online phase the ROM has a speedup factor of about an order of magnitude more compared to the full-order inversion. The output error of the data misfit-enhanced method is close to that of the full-order method. The output errors of Monte Carlo decrease with increasing numbers of simulation but do not reach the output error of the full-order system. The source code is available in MATLAB [8].

7.5 Dynamic mode decomposition

DMD is a data-driven, equation-free ROM technique [20]. It was initially developed to reduce the high-dimensional dynamic data obtained from experiments and simulations in fluid mechanics into a small number of coupled spatio-temporal modes [29, 30]. DMD was applied to explore spatio-temporal patterns in large-scale neuronal recordings in [3]. DMD can be interpreted as combination of discrete Fourier transform (DFT) in time and PCA [14] in space. Both PCA and independent component analysis [13] are static techniques, which perform poorly in reproducing dynamic data, such as recordings of neural activity.

The data are taken from electrocorticography (ECoG) recordings. Voltages from n channels of an electrode array were sampled every Δt. These measurements are arranged at snapshot k to the column vectors $\mathbf{x}_k$. The m snapshots in time construct data matrices shifted in time with Δt,

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \tag{7.24}$$

These matrices are assumed to be related linearly in time,

$$\mathbf{X}' = \mathbf{A}\mathbf{X}. \tag{7.25}$$

The DMD of the data matrix pair $\mathbf{X}$ and $\mathbf{X}'$ is given by the eigendecomposition of $\mathbf{A}$ using the singular value decomposition of the data matrix $\mathbf{X} = U\Sigma V^*$ by computing the pseudo-inverse $\mathbf{A} \approx \mathbf{X}'\mathbf{X}^\dagger \equiv \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*$. The spatio-temporal modes are computed by the exact DMD algorithm [32].

Because DMD does not contain explicit spatial relationships between neighboring measurements, traveling waves occurring in neuronal networks cannot be captured well with a few coherent modes. DMD was also used as a windowed technique with a temporal window size constrained by lower bound as for DFT. In contrast to the fluid dynamics where $n \gg m$, in neuroscience the electrode arrays have tens of channels n in the recordings with m number of snapshots in the windows data per second, so that $n < m$. The number of singular values v of $\mathbf{X}$ are less than n and $m - 1$, which restricts the maximum number of DMD modes and eigenvalues to n. Because of this the dynamics can be captured over m snapshots. The rank mismatch is resolved by appending to the snapshot measurements with $h - 1$ time-shifted versions of the data matrices. The augmented data matrix $\mathbf{X}_{\text{aug}}$ is given as

$$\mathbf{X}_{\text{aug}} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-h} \\ | & | & & | \\ | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{m-h-1} \\ | & | & & | \\ & & \cdots & \\ | & | & & | \\ \mathbf{x}_h & \mathbf{x}_{h+1} & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}. \tag{7.26}$$

The augmented matrices $\mathbf{X}_{\text{aug}}$ and $\mathbf{X}'_{\text{aug}}$ are Hankel matrices, which are constant along the skew diagonal, as in the eigenvalue realization algorithm [15]. The number of stacks h is chosen such that $hn > 2m$. A measure to determined the optimal number of stacks h is the approximation error

$$E = \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F},$$

where $\|\cdot\|_F$ is the Frobenius norm. The approximation error E is decreasing with increasing number of stacks h and reaches a plateau, so that the DMD accuracy does not significantly increase.

DMD is applied in [3] as an automated approach to detect and analyze reliably spatial localization and frequencies of sleep spindle networks from human ECoG recordings. A MATLAB implementation is available at github.com/bwbrunton/dmd-neuro/.

7.6 Reduced-order modeling of biophysical neuronal networks

Recently ROMs for ODEs

$$\dot{x}(t) = A(t)x(t) + f\big(x(t)\big) + Bu(t) \tag{7.27}$$

were constructed using POD and the DEIM to investigate input-output behavior of the neuronal networks in the brain [22, 21], where $x(t)$ are state and $u(t)$ are input variables.

The model in [22] is based on the chemical reactions of molecules in synapses, that are the intercellular information transfer points of neurons. The signaling pathways in striatal synaptic plasticity are modeled in [19]. This model describes how certain molecules, which are a prerequisite for learning in the brain, act in synapses. The stoichiometric equations obey the law of mass action, which leads to a deterministic

system of 44 ODEs of the form (7.27). The state $x(t)$ of the control system (7.27) is a collection of ions, molecules, and proteins that act in neuronal synapses. The linear part of (7.27) is sparse, the nonlinearities are quadratic. The time-dependent stimulus $u(t)$ consists of molecules that are important for neuronal excitability and plasticity, calcium and glutamate.

In [21], a nonlinear biophysical network model is considered, describing synchronized population bursting behavior of heterogeneous pyramidal neurons in the brain [27]. Neurons communicate by changing their membrane voltage to create action potentials (spikes), propagating from cell to cell. Spiking is the fundamental method of sensory information processing in the brain, and synchronized spiking is an emergent property of biological neuronal networks. The ODE system (7.27) in [21] consists of the states $x(t)$ as a collection of 50 neurons, each modeled with 10 ODEs, leading totally to a system of ODEs of dimension 500. Each cell is modeled with HH equations, where each cell has only two compartments (dendrites and soma) and these compartments have different ion channels. The state variables $x(t)$ include the voltages of somatic and dendritic compartments, the dendritic calcium concentration, and synaptic and ion channel gating variables, and the input $u(t)$ is an injected current. Additionally, the soma compartment voltages are coupled to dentritic compartments of randomly chosen cells. This networking of the output of cells as input to other cells is key for producing synchronized population behavior. The nonlinearities are HH type, i. e., exponential functions as well as cubic and quartic polynomials.

In [22], POD-DEIM was applied to a data-driven biological model of plasticity in the brain (7.27). The ROMs with POD-DEIM reduce significantly the simulation time and error between the original model and reduced-order solutions can be tuned by adjusting the number of POD and DEIM bases independently. When the ROMs are trained in a matching time interval of 10,000 seconds, accurate results are obtained. However, generalizing the reduced model to longer time intervals is challenging, which is characteristic for all nonlinear models.

In [21], the network model (7.27) is reduced with localized DEIM [24], discrete adaptive POD (DAPOD) [33, 34], and adaptive DEIM (ADEIM) [25]. DEIM and the variations are used here in combination with POD. ROMs require large numbers of POD and DEIM bases, to accurately simulate the input-output behavior in the ROMs. In this model, every cell is heterogeneous in parameters and there are also jump/reset conditions, which are factors that pose additional challenges to the ROMs. However, the ROMs in [21] were able to replicate the emergent synchronized population activity in the original network model. DAPOD and ADEIM perform best in preserving the spiking activity of the original network model. ADEIM is too slow and does not allow low enough dimensions to offset the computational costs of online adaptivity. DAPOD is able to find a lower-dimensional POD basis online than the other methods find offline, but the runtime is close to that of the original model.

Bibliography

[1] D. Amsallem and J. Nordström, High-order accurate difference schemes for the Hodgkin–Huxley equations, *J. Comput. Phys.*, **252** (2013), 573–590.

[2] D. Amsallem and J. Nordström, Energy stable model reduction of neurons by nonnegative discrete empirical interpolation, *SIAM J. Sci. Comput.*, **38** (2) (2016), B297–B326.

[3] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition, *J. Neurosci. Methods*, **258** (2016), 1–15.

[4] L. Buesing, J. H. Macke, and M. Sahani, Spectral learning of linear dynamics from generalised-linear observations with application to neural population data, in F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1682–1690, Curran Associates, Inc., 2012.

[5] S. Chaturantabut and D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[6] J. P. Cunningham and Byron M. Yu, Dimensionality reduction for large-scale neural recordings, *Nat. Neurosci.*, **17** (11) (2014), 1500–1509.

[7] K. J. Friston, L. Harrison, and W. Penny, Dynamic causal modelling, *NeuroImage*, **19** (4) (2003), 1273–1302.

[8] C. Himpe, *OPTMOR – OPTimization-based Model Order Reduction (version 1.2)*, 2015.

[9] C. Himpe, *Combined State and Parameter Reduction: For Nonlinear Systems with an Application in Neuroscience*, Internationaler Fachverlag für Wissenschaft & Praxis, 2016.

[10] C. Himpe and M. Ohlberger, Data-driven combined state and parameter reduction for inverse problems, *Adv. Comput. Math.*, **41** (5) (2015), 1343–1364.

[11] M. Hines, Efficient computation of branched nerve equations, *Int. J. Biomed. Comput.*, **15** (1) (1984), 69–76.

[12] A. L. Hodgkin and A. F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *Bull. Math. Biol.*, **52** (1) (Jan 1990), 25–71.

[13] A. Hyvärinen and E. Oja, Independent component analysis: algorithms and applications, *Neural Netw.*, **13** (4) (2000), 411–430.

[14] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer-Verlag, New York, 2005.

[15] J. N. Juang and R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, *J. Guid. Control Dyn.*, **8** (5) (1985), 620–627.

[16] A. R. Kellems, S. Chaturantabut, D. C. Sorensen, and S. J. Cox, Morphologically accurate reduced order modeling of spiking neurons, *J. Comput. Neurosci.*, **28** (3) (2010), 477–494.

[17] A. R. Kellems, D. Roos, N. Xiao, and S. J. Cox, Low-dimensional, morphologically accurate models of subthreshold membrane potential, *J. Comput. Neurosci.*, **27** (2) (2009), 161.

[18] C. Kemere, G. Santhanam, B. M. Yu, A. Afshar, S. I. Ryu, T. H. Meng, and K. V. Shenoy, Detecting neural-state transitions using hidden Markov models for motor cortical prostheses, *J. Neurophysiol.*, **100** (4) (2008), 2441–2452.

[19] B. Kim, S. L. Hawes, F. Gillani, L. J. Wallace, and K. T. Blackwell, Signaling pathways involved in striatal synaptic plasticity are sensitive to temporal pattern and exhibit spatial specificity, *PLoS Comput. Biol.*, **9** (3) (2013), e1002953.

[20] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2016.

[21] M. Lehtimäki, L. Paunonen, and M.-L. Linne, Projection-based order reduction of a nonlinear biophysical neuronal network model, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1–6, 2019.
[22] M. Lehtimäki, L. Paunonen, S. Pohjolainen, and M.-L. Linne, Order reduction for a signaling pathway model of neuronal synaptic plasticity, *IFAC-PapersOnLine*, **50** (1) (2017), 7687–7692, 20th IFAC World Congress.
[23] C. Lieberman, K. Willcox, and O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2523–2542.
[24] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, Localized discrete empirical interpolation method, *SIAM J. Sci. Comput.*, **36** (1) (2014), A168–A192.
[25] B. Peherstorfer and K. Willcox, Online adaptive model reduction for nonlinear systems via low-rank updates, *SIAM J. Sci. Comput.*, **37** (4) (2015), A2123–A2150.
[26] B. Petreska, B. M. Yu, J. P. Cunningham, S. Gopal, S. I. Ryu, K. V. Shenoy, and M. Sahani, Dynamical segmentation of single trials from population neural data, in J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 24*, pp. 756–764, Curran Associates, Inc., 2011.
[27] P. F. Pinsky and J. Rinzel, Intrinsic and network rhythmogenesis in a reduced traub model for CA3 neurons, *J. Comput. Neurosci.*, **1** (1) (1994), 39–60.
[28] S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (5500) (2000), 2323–2326.
[29] C. W. Rowley, I. Mezić, S. Bahheri, P. Schlatter, and D. S. Hennigson, Spectral analysis of nonlinear flows, *J. Fluid Mech.*, **641** (2009), 115–127.
[30] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.*, **656** (2010), 5–28.
[31] J. B. Tenenbaum, V. de Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, **290** (5500) (2000), 2319–2323.
[32] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: Theory and applications, *J. Comput. Dyn.*, **1** (2) (2014), 391–421.
[33] M. Yang and A. Armaou, Dissipative distributed parameter systems on-line reduction and control using DEIM/APOD combination, in *2018 Annual American Control Conference (ACC)*, pp. 2557–2562, 2018.
[34] M. Yang and A. Armaou, Revisiting APOD accuracy for nonlinear control of transport reaction processes: A spatially discrete approach, *Chem. Eng. Sci.*, **181** (2018), 146–158.
[35] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity, *J. Neurophysiol.*, **102** (1) (2009), 614–635.

Niccolò Dal Santo, Andrea Manzoni, Stefano Pagani, and
Alfio Quarteroni

8 Reduced-order modeling for applications to the cardiovascular system

Abstract: The capability to provide fast and reliable numerical simulations is of paramount importance when dealing with complex applications arising from medicine. More than for other branches of engineering and applied sciences, performing accurate computations in a short amount of time – minutes, rather than hours, or even days – is crucial when dealing with problems arising from life sciences, like, e. g., in the simulation of the cardiovascular system. Moreover, many sources of variability carried by subject-specific features have to be incorporated into the mathematical models, to quantify their impact on the computed results. For these reasons, bringing computational results into clinical practice represents a great challenge. Reduced-order modeling techniques such as the reduced basis method represent a key tool towards the possibility to address these challenges, thus making the numerical modeling of the cardiovascular system a new, fascinating testbed for these methodologies.

Keywords: reduced basis method, proper orthogonal decomposition, hyperreduction techniques, hemodynamics, cardiac electrophysiology

MSC 2010: 65M60, 65N12, 76M10

8.1 Numerical simulations in clinical practice

The numerical modeling of the cardiovascular system is a research topic that has attracted remarkable interest from the scientific computing community because of the intrinsic mathematical and computational difficulty, and due to the increasing impact of cardiovascular diseases worldwide. A wealth of models are nowadays available to address both physiological and pathological instances, aiming at better understanding the quantitative processes governing the blood circulation and opening new fron-

Acknowledgement: We acknowledge the Swiss National Supercomputing Center for providing the CPU resources under project ID s796.

Niccolò Dal Santo, Institute of Mathematics, École Polytechnique Fédérale de Lausanne (EPFL), Station 8, 1015 Lausanne, Switzerland
Andrea Manzoni, Stefano Pagani, MOX – Department of Mathematics, Politecnico di Milano, P.za Leonardo da Vinci 32, 20133 Milano, Italy
Alfio Quarteroni, MOX – Department of Mathematics, Politecnico di Milano, P.za Leonardo da Vinci 32, 20133 Milano, Italy; and Institute of Mathematics, École Polytechnique Fédérale de Lausanne (EPFL), Station 8, 1015 Lausanne, Switzerland

https://doi.org/10.1515/9783110499001-008

tiers in therapeutic planning and design of implantable devices (e. g., medical stents and cardiac defibrillators). Numerical simulations provide useful insights into the behavior of quantities which cannot be measured directly, such as the wall shear stress (WSS) over a portion of the lumen boundary or the transmembrane potential in the myocardium. This is meant to enable quantitative analysis in several virtual scenarios to support clinicians' decisions and to enhance common diagnostic practices based on medical imaging.

However, a number of difficulties (including, e. g., the lack of several physical parameters or the uncertainty affecting their values, as well as the presence of a wide range of spatio-temporal scales) arises when solving problems occurring in the description of the cardiovascular system. Even more importantly, their numerical simulation by means of high-fidelity approximation techniques, also called full-order models (FOMs), such as the finite element (FE) method might be extremely demanding. This is the case, for instance, of numerical simulations of cardiac electromechanics, as well as the description of fluid dynamics of blood flowing through the heart chambers (two ventricles and two atria). The numerical solution of these problems – expressed in the form of systems of partial differential equations (PDEs) – easily involves up to $O(10^6)$ degrees of freedom and several hours (or even days) of computational time, also on powerful parallel architectures. Consequently, high-fidelity approximation techniques become prohibitive when we expect them to deal quickly, but accurately, with the repetitive solution of problems in view, e. g., of model personalization – that is, the adaptation of model inputs to subject-specific conditions. For instance, when simulating the electric activity of the heart (Section 8.2.2) the inputs to be personalized may include the shape of the domain (e. g., the geometry of atria and ventricles), physical parameters (e. g., electric conductivities or the orientation of tissue fibers), and initial and boundary conditions. In this context, physical indices and outputs of clinical interest can be directly approximated through the numerical solution of nonlinear parameterized coupled systems of PDEs, such as the bidomain or monodomain equations, equipped with a system of ordinary differential equations (ODEs) encoding suitable ionic models; see, e. g., [20, 51, 53]. Furthermore, very often input/output evaluations represent building blocks of more complex problems, as data assimilation, parameter estimation, and uncertainty quantification problems.

To reduce the computational complexity of the FOM in all these contexts, reduced-order models (ROMs) such as the reduced basis (RB) method for parameterized PDEs represent efficient techniques for the approximation of the parameterized PDE solution. Although several works have focused on problems related to both hemodynamics [5, 6, 43] and the simulation of cardiac function [12, 11, 21, 27, 41, 47], applying state-of-the-art ROMs is not straightforward for cardiac problems because of (i) nonlinear behavior (like sharp moving fronts in the case of cardiac electrophysiology) and (ii) parameterization of complex geometries. For these reasons, suitable strategies must be devised to build low-dimensional RB spaces able to capture the manifold of the

problem solutions when varying the parameters, by keeping the cost of the ROM construction sufficiently low. Local (or nonlinear) techniques to build RB spaces and efficient, purely algebraic hyperreduction strategies can help.

In this chapter we review the current state-of-the-art construction of efficient and accurate RB methods for the solution of parameterized problems dealing with (i) arterial fluid dynamics and (ii) the electric activity of the heart. Throughout the chapter, $\boldsymbol{\mu} \in \mathcal{P}$ denotes a parameter vector entering in the definition of the PDE model, whose components might represent physical and/or geometrical features of interest; $\mathcal{P} \subset \mathbb{R}^p$ denotes the parameter space.

The structure of this chapter is as follows. In Section 8.2 some insights on the two problems we focus on – namely, arterial blood flow and cardiac electrophysiology – are provided. In Section 8.3 we sketch the main difficulties arising when dealing with the construction of ROMs for these two problems, addressing some possible strategies to enhance computational efficiency and numerical accuracy of ROMs. Numerical results related to arterial blood flow and cardiac electrophysiology are presented in Sections 8.4 and 8.5, respectively. Finally, open critical issues and future perspectives are briefly outlined in Section 8.6. For the sake of space, we do not report the full description of the FOMs and the ROMs involved in the solution of the proposed problems; the interested reader can find further details in [25, 47].

8.2 Two relevant cardiovascular applications

8.2.1 Arterial blood flow

The cardiovascular system is a closed circuit that carries oxygenated blood to all the tissues and organs of the body. The systemic circulation is made up of the arteries, carrying oxygenated blood ejected by the left heart to the living tissues, and the veins, allowing nonoxygenated blood to return to the right heart. In large arteries, blood behaves like a Newtonian fluid, modeled by unsteady Navier–Stokes equations, with pulsatile input. The different velocity of blood flow in the arteries of the systemic circulation results in different values of the Reynolds number $\mathrm{Re} = \rho DU/\mu$ (where D and U are characteristic vessel dimension and blood velocity, respectively), a dimensionless quantity which highlights the importance of the inertial terms over the viscous ones. Here we assume to deal with laminar flows. Indeed, blood experiences a wide range of Reynolds numbers; moreover, the pulsatile nature of blood flow does not allow the onset of fully turbulent flow in healthy conditions. This is not necessarily the case for some pathological conditions, such as (severe) carotid stenosis, yielding a narrowing of the vessel lumen and increased complexity of the geometry together with higher Reynolds numbers [38]. Occlusions (or stenoses) at the carotid bifurcation are caused by the accumulation of fatty material in the internal layer of the vessel wall, progres-

sively leading to plaque formation and atherosclerosis. The main complications are partial occlusion of the lumen with possible generation of cerebral ischemia, or even total occlusion, resulting in cerebral infarction. The role of blood fluid dynamics has been recognized as crucial for the development of such a disease [67, 39, 63]. In particular, WSSs, that is, the viscous/friction forces exerted by the blood on the vessel wall, despite being 100 times smaller in magnitude than pressure, play an important role in atherosclerosis. For the case at hand, we will consider less severe occlusions occurring at the carotid bifurcation, thus justifying the assumption of laminar flow.

In particular, we are interested to characterize blood flow patterns in different geometrical configurations obtained by increasing the degree of stenosis of a subject-specific carotid bifurcation geometry, for a wide range of flow conditions. Hence, both geometrical and physical parameters are considered: The former are related with the computational domain, the latter with the problem's data (affecting the Reynolds number). Given an open bounded and $\boldsymbol{\mu}$-dependent domain $\Omega(\boldsymbol{\mu}) \subset \mathbb{R}^d$, $d = 2, 3$, such that $\partial\Omega(\boldsymbol{\mu}) = \Gamma_{\text{out}}(\boldsymbol{\mu}) \cup \Gamma_{\text{in}}(\boldsymbol{\mu}) \cup \Gamma_w(\boldsymbol{\mu})$ and $\mathring{\Gamma}_{\text{out}}(\boldsymbol{\mu}) \cap \mathring{\Gamma}_{\text{in}}(\boldsymbol{\mu}) = \mathring{\Gamma}_w(\boldsymbol{\mu}) \cap \mathring{\Gamma}_{\text{in}}(\boldsymbol{\mu}) = \mathring{\Gamma}_{\text{out}}(\boldsymbol{\mu}) \cap \mathring{\Gamma}_w(\boldsymbol{\mu}) = \emptyset$, and a final time $T > 0$, unsteady Navier–Stokes equations for Newtonian, incompressible fluids read as follows: For each $t \in (0, T)$,

$$
\begin{cases}
\dfrac{\partial \vec{u}(\boldsymbol{\mu})}{\partial t} + \vec{u}(\boldsymbol{\mu}) \cdot \nabla \vec{u}(\boldsymbol{\mu}) - \nabla \cdot \boldsymbol{\sigma}(\vec{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) + \nabla p(\boldsymbol{\mu}) = \vec{0} & \text{in } \Omega(\boldsymbol{\mu}), \\
\nabla \cdot \vec{u}(\boldsymbol{\mu}) = 0 & \text{in } \Omega(\boldsymbol{\mu}), \\
\vec{u}(\boldsymbol{\mu}) = \vec{0} & \text{on } \Gamma_w(\boldsymbol{\mu}), \\
\vec{u}(\boldsymbol{\mu}) = \vec{g}_{\text{NS}}(\boldsymbol{\mu}) & \text{on } \Gamma_{\text{in}}(\boldsymbol{\mu}), \\
\boldsymbol{\sigma}(\vec{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))\vec{n}(\boldsymbol{\mu}) = \vec{0} & \text{on } \Gamma_{\text{out}}(\boldsymbol{\mu}), \\
\vec{u}(\boldsymbol{\mu}) = \vec{u}_0 & \text{in } \Omega(\boldsymbol{\mu}), \text{ at } t = 0.
\end{cases}
\tag{8.1}
$$

Here $\vec{u}(\boldsymbol{\mu})$ and $p(\boldsymbol{\mu})$ are the velocity and the pressure of the fluid; $\boldsymbol{\sigma}(\vec{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) = -p(\boldsymbol{\mu})\mathbf{I} + 2\nu\boldsymbol{\varepsilon}(\vec{u}(\boldsymbol{\mu}))$ denotes the stress tensor; and $\nu = \mu/\rho$ denotes the kinematic viscosity of the fluid, ρ being the blood density and ν its viscosity. The strain tensor is given by $\boldsymbol{\varepsilon}(\vec{u}(\boldsymbol{\mu})) = \frac{1}{2}(\nabla \vec{u}(\boldsymbol{\mu}) + \nabla \vec{u}(\boldsymbol{\mu})^T)$. Note that also the Dirichlet boundary condition $\vec{g}_{\text{NS}}(\boldsymbol{\mu})$ might depend on $\boldsymbol{\mu}$. We avoid to deal with fluid–structure interaction (FSI) for the problem at hand; indeed, arterial vessels are compliant, yielding quite large wall displacements (reaching up to 10 % of the lumen diameter). The reduction of FSI problems has only been partially addressed in few works, focusing on simplified geometries, and without addressing efficient hyperreduction techniques; see, e. g., [37, 8].

Problem (8.1) is first discretized in space by means of the FE method, relying on quadratic and linear FEs for velocity and pressure, respectively, and then in time with a semi-implicit backward differentiation formula (BDF) scheme of order $\sigma = 2$. We introduce a partition of $[0, T]$ in N_t subintervals of equal size $\Delta t = T/N_t$, such that

$t_n = n\Delta t$, and approximate the time derivative of the FOM velocity $\vec{u}_h(\boldsymbol{\mu})$ at t_n as

$$\frac{d\vec{u}_h(\boldsymbol{\mu})}{dt} \approx \frac{\frac{3}{2}\vec{u}_h^{n+1}(\boldsymbol{\mu}) - \vec{u}_h^{n,\sigma}(\boldsymbol{\mu})}{\Delta t}, \quad \vec{u}_h^{n,\sigma}(\boldsymbol{\mu}) = 2\vec{u}_h^{n}(\boldsymbol{\mu}) - \frac{1}{2}\vec{u}_h^{n-1}(\boldsymbol{\mu}). \tag{8.2}$$

This yields a sequence in time of parameterized linear systems of the form

$$\mathbb{N}(\mathbf{u}^{n,*}(\boldsymbol{\mu});\boldsymbol{\mu}) \begin{bmatrix} \mathbf{u}^{n+1}(\boldsymbol{\mu}) \\ \mathbf{p}^{n+1}(\boldsymbol{\mu}) \end{bmatrix} = \mathbf{g}^{n+1}(\boldsymbol{\mu}), \quad n = 0, \dots, N_t - 1, \tag{8.3}$$

where $\mathbf{u}^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h^u}$ and $\mathbf{p}^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h^p}$ denote the FOM vector representation of the velocity and the pressure, at time t_n, $\mathbf{u}^0(\boldsymbol{\mu}) = \mathbf{u}_0 \in \mathbb{R}^{N_h^u}$ is the initial condition, and $\mathbb{N}(\mathbf{u}^{n,*}(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{g}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are given by

$$\begin{aligned} \mathbb{N}(\mathbf{u}^{n,*}(\boldsymbol{\mu});\boldsymbol{\mu}) &= \begin{bmatrix} \frac{\alpha_1}{\Delta t}\mathbb{M}^u(\boldsymbol{\mu}) + \mathbb{D}(\boldsymbol{\mu}) + \mathbb{C}(\mathbf{u}^{n,*}(\boldsymbol{\mu});\boldsymbol{\mu}) & \mathbb{B}^T(\boldsymbol{\mu}) \\ \mathbb{B}(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix}, \\ \mathbf{g}^{n+1}(\boldsymbol{\mu}) &= \begin{bmatrix} \frac{1}{\Delta t}\mathbb{M}^u(\boldsymbol{\mu})\mathbf{u}^{n,\sigma}(\boldsymbol{\mu}) + \mathbf{f}_1^{n+1}(\boldsymbol{\mu}) \\ \mathbf{f}_2^{n+1}(\boldsymbol{\mu}) \end{bmatrix}. \end{aligned} \tag{8.4}$$

Here $\mathbb{M}^u(\boldsymbol{\mu}) \in \mathbb{R}^{N_h^u \times N_h^u}$ is the velocity mass matrix; $\mathbb{D}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h^u \times N_h^u}$ and $\mathbb{B}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h^p \times N_h^u}$ encode the velocity stiffness and the divergence operator, whereas $\mathbb{C}(\mathbf{u}^{n,*}(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{N_h^u \times N_h^u}$ arises from the linearization (about $\mathbf{u}^{n,*}(\boldsymbol{\mu}) = 2\mathbf{u}^n(\boldsymbol{\mu}) - \mathbf{u}^{n-1}(\boldsymbol{\mu})$) of the nonlinear term.

To deal with complex domains and their deformations in a flexible way, we exploit a general mesh deformation technique (see, e. g., [60]), in which domain deformations result from an additional FE problem providing either an harmonic [4] or an elastic [64, 62, 61] deformation by properly extending boundary displacements; see also [42] for further details. The corresponding meshes are also obtained as a deformation of a reference mesh, hence not affecting the topology of the degrees of freedom. See also [13] for the case of a parameterized ROM for Navier–Stokes equations in domains with variable shapes, relying on an FOM based on a finite volume discretization, built for accelerating the calculation of pressure drop along blood vessels.

8.2.2 Cardiac electrophysiology

The propagation of the electric signal through cardiac cells is the main mechanism responsible for their contraction, finally resulting in atrial and ventricular contractions. Mathematical models of cardiac electrophysiology describe the action potential mechanism of depolarization and polarization of the cardiac cells, which consists of rapid variations of the cell membrane electric potential with respect to a resting potential. In particular, cellular models characterize the electric potential of a single cell, whereas

physiological models provide a quantitative description of the action potential propagation at the tissue level. The former are based on systems of ODEs that describe the variation of ionic species and ionic currents; the latter are derived from the former by means of suitable homogenization procedures.

Here we are interested to characterize the evolution of the electric potential for a wide range of physical parameters affecting both electric conductivities at the tissue level and ion dynamics at the cell level. Coupling the monodomain model for the transmembrane potential $u(\boldsymbol{\mu})$ with a phenomenological model for the ionic currents – here involving a single gating variable $w(\boldsymbol{\mu})$ – in a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, representing, e. g., a portion of the myocardium (or the whole left ventricle) results in the following time-dependent nonlinear differential system: For each $t \in (0, T)$,

$$\begin{aligned} \frac{\partial u(\boldsymbol{\mu})}{\partial t} - \operatorname{div}(\mathbf{D}(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) + I_{\text{ion}}(u(\boldsymbol{\mu}), w(\boldsymbol{\mu}); \boldsymbol{\mu}) &= I_{\text{app}}(t; \boldsymbol{\mu}) \quad \text{in } \Omega, \\ \frac{\partial w(\boldsymbol{\mu})}{\partial t} + g(u(\boldsymbol{\mu}), w(\boldsymbol{\mu}); \boldsymbol{\mu}) &= 0 \quad \text{in } \Omega, \\ \frac{\partial u(\boldsymbol{\mu})}{\partial \vec{n}} &= 0 \quad \text{on } \partial\Omega, \\ u(\boldsymbol{\mu}) = u_0, \quad w(\boldsymbol{\mu}) &= w_0 \quad \text{in } \Omega, \text{ at } t = 0. \end{aligned} \tag{8.5}$$

Here t denotes a rescaled time, I_{app} is an applied current representing the initial activation of the tissue, I_{ion} and g model the cellular bioelectric activity, and $\vec{d}$ denotes the diffusivity tensor. For the case at hand, we aim at estimating the effect of (i) anisotropic conductivity, (ii) ion dynamics, and (iii) activation times on the electric conduction by parameterizing the tensor $\vec{d}$, the functions I_{ion} and g, and the source term I_{app}, respectively.

We model the cardiac tissue as composed of fibers (the cardiomyocytes) whose orientation varies from the epicardium to the endocardium due to the laminar organization in sheets of the tissue [58]. At the macroscopic level, this structure yields preferential directions for the action potential traveling front. Therefore, at any point $\vec{x}$, an orthonormal local reference system is described by the principal axes $\vec{f}_0(\vec{x})$, $\vec{s}_0(\vec{x})$, and $\vec{n}_0(\vec{x})$, with $\vec{f}_0(\vec{x})$ parallel to the fiber direction and with $\vec{s}_0(\vec{x})$ and $\vec{n}_0(\vec{x})$ orthogonal and tangent to the sheet direction. Denoting by σ_l, σ_t, and σ_v the conductivity coefficients measured along the corresponding directions $\vec{f}_0$, $\vec{s}_0$, and $\vec{n}_0$, the anisotropic conductivity tensor is

$$\vec{d}(\boldsymbol{\mu}) = \sigma_l \vec{f}_0 \otimes \vec{f}_0 + \sigma_t \vec{s}_0 \otimes \vec{s}_0 + \sigma_v \vec{n}_0 \otimes \vec{n}_0.$$

We assume that the left ventricle tissue is an axisymmetric anisotropic medium ($\sigma_t = \sigma_v$), so that the previous relation simplifies as follows:

$$\vec{d}(\boldsymbol{\mu}) = \sigma_t \mathbf{I} + (\sigma_l - \sigma_t) \vec{f}_0 \otimes \vec{f}_0,$$

where the fibers' structure is considered as (spatially dependent, but) $\boldsymbol{\mu}$-independent. The reaction term I_{ion} and the function g depend on both u and w, thus making the

PDE and the ODE two-ways coupled. In this chapter, we focus on the Aliev–Panfilov model, for which

$$\begin{aligned} I_{\text{ion}}(u, w; \boldsymbol{\mu}) &= Ku(u-a)(u-1) + wu, \\ g(u, w; \boldsymbol{\mu}) &= \left(\varepsilon_0 + \frac{c_1 w}{c_2 + u}\right)(-w - Ku(u-b-1)); \end{aligned} \tag{8.6}$$

the (parametric) coefficients K, a, b, ε_0, c_1, and c_2 are related to the cell.

Finally, we model the electric activation as the combination of three applied current stimuli, $\{\phi_i(\vec{x})\}_{i=1}^3$, at three fixed spatial locations $\{\vec{x}_i\}_{i=1}^3$, with t_1, t_2 expressing a time delay

$$I_{\text{app}}(t; \boldsymbol{\mu}) = \phi_1(\vec{x})\mathbb{I}_{[0,2]}(t) + \phi_2(\vec{x})\mathbb{I}_{[t_1, t_1+2]}(t) + \phi_3(\vec{x})\mathbb{I}_{[t_2, t_2+2]}(t),$$

where $\mathbb{I}_A(t)$ denotes the indicator function of the set A, that is, $\mathbb{I}_A(t) = 1$ if $t \in A$, and $\mathbb{I}_A(t) = 0$ otherwise.

To summarize, we consider $p = 9$ parameters for this test case: the two conductivities σ_l and σ_t; five parameters affecting the ionic model, K, a, b, c_1, ε; and two characteristic times t_1, t_2 affecting the electric activation.

Problem (8.5) is first discretized in space by means of the FE method, using linear FEs for the transmembrane potential; a semi-implicit, first-order, one-step scheme is then used for time discretization [19], in which the nonlinear vector $\mathbf{I}_{\text{ion}} \in \mathbb{R}^{N_h}$ at time t_{n+1} is calculated using the solution computed at the previous time t_n. This decouples the PDE from the ODE leading to a linear system to be solved at each time step. At each time step t_n, $n = 0, \ldots, N_t - 1$, a system of N_h (independent) nonlinear equations must be solved, arising from the backward (implicit) Euler method, under the form

$$\mathbf{w}^{n+1}(\boldsymbol{\mu}) - \Delta t\, \mathbf{g}(\mathbf{u}^n(\boldsymbol{\mu}), \mathbf{w}^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{w}^n(\boldsymbol{\mu}), \quad n = 0, \ldots, N_t - 1, \tag{8.7}$$

given $\mathbf{w}^0(\boldsymbol{\mu}) = \mathbf{w}_0(\boldsymbol{\mu})$. The so-called ionic current interpolation strategy is used to evaluate the ionic current term, so that only the nodal values are used to build a (piecewise linear) interpolant of the ionic current. This yields a sequence in time of parameterized linear systems of the form

$$\begin{aligned} &\left(\frac{\mathbb{M}(\boldsymbol{\mu})}{\Delta t} + \mathbb{A}(\boldsymbol{\mu})\right)\mathbf{u}^{n+1}(\boldsymbol{\mu}) + \mathbf{I}_{\text{ion}}(\mathbf{u}^n(\boldsymbol{\mu}), \mathbf{w}^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &\quad = \frac{\mathbb{M}(\boldsymbol{\mu})}{\Delta t}\mathbf{u}^n(\boldsymbol{\mu}) + \mathbf{I}_{\text{app}}^{n+1}(\boldsymbol{\mu}), \quad n = 0, \ldots, N_t - 1, \end{aligned} \tag{8.8}$$

where $\mathbf{u}^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $\mathbf{w}^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ denote the FOM vector representation of the transmembrane potential and the gating variable, respectively, at time t_n, and $\mathbf{u}^0 = \mathbf{u}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, $\mathbf{w}^0 = \mathbf{w}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are the initial conditions. Here $\mathbb{M}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the mass matrix and $\mathbb{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ encodes the diffusion operator, whereas $\mathbf{I}_{\text{app}}^{n+1} \in \mathbb{R}^{N_h}$ encodes the applied current at time t_{n+1}.

The major computational costs are entailed by assembling the terms $\mathbf{I}_{\text{ion}}$ and $\mathbf{g}$ at each time step and by the solution of the linear system (8.8); strong constraints on the spatial mesh size have to be taken into account due to the propagation of very steep fronts [18, 49, 28], yielding a very large dimension N_h. On its turn, the time step Δt is required to be sufficiently small to capture the fast dynamics characterizing the propagation of the electric signal [26].

8.3 Reduced-order modeling

ROMs aim at reducing the computational complexity and costs entailed by the repeated solution of PDE problems [10, 52]. In the case of parameterized PDEs, the RB method is a remarkable instance of ROM that allows to dramatically reduce the dimension of the discrete problems arising from numerical approximation – from millions to hundreds, or thousands at most, degrees of freedom. Proper orthogonal decomposition (POD) is a general-purpose technique widely used to build RB spaces; later, a (Petrov–)Galerkin projection onto the RB space is employed to generate the ROM. Whenever cheaply computable a posteriori error bounds are available, greedy algorithms can be used as an alternative strategy.

Parameterized blood flows in idealized cardiovascular geometries have been addressed by means of ROMs in [37, 43] and in [5] by taking into account more complex (and computationally challenging) subject-specific configurations; in all these cases, solutions of Navier–Stokes equations are computed with respect to inflow and/or geometrical parameters, however without exploiting efficient, purely algebraic hyperreduction techniques. Applications to PDE-constrained optimization problems arising in the context of optimal design of prosthetic devices can be found, e. g., in [44, 36]. An application of POD to the analysis of transient turbulence in a stenosed carotid artery, however without dealing with parameterized flows and the solution of the problem for new parameter instances, has been considered in [29]. Except for few cases [12, 14, 15, 21, 27], which however do not systematically explore parameter-dependent problems, the application of the RB method to the simulation of the cardiac function in subject-specific configurations is work in progress.

In this chapter we provide a quick overview of the way the most relevant difficulties related to the application of the RB method to the two problems introduced in the previous sections have been tackled. In both cases we will rely on POD for constructing the RB spaces and on suitable hyperreduction techniques to deal with the efficient assembling of nonlinear or nonaffinely parameter-dependent terms appearing in the ROM.

8.3.1 Navier–Stokes equations

The most relevant issues related to the reduction of the parameterized Navier–Stokes system (8.1) for the application at hand are (i) the construction of flexible, low-dimensional *shape parameterization* to describe complex shapes in terms of few parameters and (ii) the *stability* of the ROM (in the sense of a suitable inf-sup stability condition) when dealing with the approximation of both velocity and pressure.

Expressing the RB approximation of velocity and pressure fields at time t_n as a linear combination of the RB basis functions,

$$\mathbf{u}^n(\boldsymbol{\mu}) \approx \mathbb{V}_u \mathbf{u}_N^n(\boldsymbol{\mu}), \quad \mathbf{p}^n(\boldsymbol{\mu}) \approx \mathbb{V}_p \mathbf{p}_N^n(\boldsymbol{\mu}), \tag{8.9}$$

where $\mathbb{V}_u \in \mathbb{R}^{N_h^u \times N_u}$ and $\mathbb{V}_p \in \mathbb{R}^{N_h^p \times N_p}$ collect the (degrees of freedom of the) basis functions, a Galerkin projection yields the following Galerkin-RB problem: Given $\boldsymbol{\mu} \in \mathcal{P}$, $\mathbf{u}_N^{n-1}$, $\mathbf{u}_N^n$, find $(\mathbf{u}_N^{n+1}(\boldsymbol{\mu}), \mathbf{p}_N^{n+1}(\boldsymbol{\mu})) \in \mathbb{R}^{N_u} \times \mathbb{R}^{N_p}$ such that $\mathbf{u}_N^0(\boldsymbol{\mu}) = \mathbf{u}_{N,0}$ and

$$\mathbb{N}_N(\mathbb{V}_u \mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu}) \begin{bmatrix} \mathbf{u}_N^{n+1}(\boldsymbol{\mu}) \\ \mathbf{p}_N^{n+1}(\boldsymbol{\mu}) \end{bmatrix} = \mathbf{g}_N^{n+1}(\boldsymbol{\mu}), \quad n = 1, \dots, N_t - 1, \tag{8.10}$$

where $\mathbf{u}_{N,0} = \mathbb{V}_u^T \mathbf{u}_0$,

$$\begin{aligned} \mathbb{N}_N(\mathbb{V}_u \mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \begin{bmatrix} \frac{\alpha_1}{\Delta t} \mathbb{M}_N^u(\boldsymbol{\mu}) + \mathbb{D}_N(\boldsymbol{\mu}) + \mathbb{C}_N(\mathbb{V}_u \mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu}) & \mathbb{B}_N^T(\boldsymbol{\mu}) \\ \mathbb{B}_N(\boldsymbol{\mu}) & 0 \end{bmatrix}, \\ \mathbf{g}_N^{n+1}(\boldsymbol{\mu}) &= \begin{bmatrix} \frac{1}{\Delta t} \mathbb{M}_N^u(\boldsymbol{\mu}) \mathbf{u}_N^{n,\sigma}(\boldsymbol{\mu}) + \mathbf{f}_{N1}^{n+1}(\boldsymbol{\mu}) \\ \mathbf{f}_{N2}^{n+1}(\boldsymbol{\mu}) \end{bmatrix}, \end{aligned} \tag{8.11}$$

and

$$\begin{gathered} \mathbb{D}_N(\boldsymbol{\mu}) = \mathbb{V}_u^T \mathbb{D}(\boldsymbol{\mu}) \mathbb{V}_u, \quad \mathbb{M}_N^u(\boldsymbol{\mu}) = \mathbb{V}_u^T \mathbb{M}^u(\boldsymbol{\mu}) \mathbb{V}_u, \quad \mathbb{B}_N(\boldsymbol{\mu}) = \mathbb{V}_p^T \mathbb{B}(\boldsymbol{\mu}) \mathbb{V}_u, \\ \mathbb{C}_N(\mathbb{V}_u \mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{V}_u^T \mathbb{C}(\mathbb{V}_u \mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbb{V}_u, \\ \mathbf{f}_{N1}^{n+1}(\boldsymbol{\mu}) = \mathbb{V}_u^T \mathbf{f}_1^{n+1}(\boldsymbol{\mu}), \quad \mathbf{f}_{N2}^{n+1}(\boldsymbol{\mu}) = \mathbb{V}_p^T \mathbf{f}_2^{n+1}(\boldsymbol{\mu}). \end{gathered}$$

To construct the RB matrices $\mathbb{V}_u$ and $\mathbb{V}_p$ we rely on POD – separately on velocity and pressure variables – by collecting snapshots of the FOM solution for a sample of selected parameter values $\boldsymbol{\mu}_i$, $i = 1, \dots, n_s$, and computing, for $n = 0, \dots, N_t - 1$, the solution of the FOM (8.3). In particular, POD is first performed with respect to the time trajectory (for a fixed $\boldsymbol{\mu}_i$) and then with respect to $\boldsymbol{\mu}$. Three issues must then be taken into account:

1. *ROM stability*. Performing, as in (8.10), a Galerkin projection onto the RB space built through the POD procedure above does not automatically ensure the stability of the resulting RB problem (in the sense of the fulfillment of an *inf-sup* condition at the reduced level). This yields a potentially singular matrix $\mathbb{N}(\mathbf{u}^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu})$.

Several strategies can be employed to overcome this shortcoming; here we augment the velocity space by means of a set of *enriching* basis functions computed through the pressure supremizing operator. For each $i = 1, \dots, n_s$ and for each $n = 1, \dots, N_t$ we compute the so-called supremizers by solving

$$\mathbb{X}_u(\boldsymbol{\mu}_i)\mathbf{t}_p^n(\boldsymbol{\mu}_i) = \mathbb{B}^T(\boldsymbol{\mu}_i)\mathbf{p}^n(\boldsymbol{\mu}_i), \tag{8.12}$$

where $\mathbb{X}_u \in \mathbb{R}^{N_h^u \times N_h^u}$ encodes the scalar product over the velocity space. Then, POD is performed to extract an enriching basis $\mathbb{V}_s \in \mathbb{R}^{N_h^u \times N_s}$, which is then merged (through a Gram–Schmidt orthonormalization) with the columns of $\mathbb{V}_u$. This strategy leads to an RB problem which is *inf-sup* stable in practice, but whose well-posedness is not rigorously proven [56, 7]. Another option to enforce the inf-sup stability would rely on a coarse algebraic least-squares RB method; however, this strategy has only been investigated in the case of steady Stokes problems so far; see [23].

2. *Efficient assembling of nonlinear terms*. Because of the $\boldsymbol{\mu}$ dependence induced by the geometry deformation, all the matrices and vectors in the ROM (8.10) depend nonaffinely on the parameter $\boldsymbol{\mu}$; moreover, a critical issue is represented by the linearized term $\mathbb{C}_N(\mathbb{V}_u\mathbf{u}_N^{n,*}(\boldsymbol{\mu}); \boldsymbol{\mu})$ appearing in (8.4). To assemble it efficiently, we rely on the matrix version of the discrete empirical interpolation method (DEIM). Such a procedure requires the evaluation of a sample of system (vectors and matrices) snapshots, followed by a POD on vectors and vectorized matrices and a further selection of a set of well-chosen interpolation points; see, e. g., [46]. The matrix DEIM (MDEIM) is another technique also employed to compute an approximated affine decomposition of the diffusion $\mathbb{D}(\boldsymbol{\mu})$, the pressure-divergence $\mathbb{B}(\boldsymbol{\mu})$, and the velocity mass $\mathbb{M}^u(\boldsymbol{\mu})$ matrices.
3. *Low-dimensional parameterization of the computational domain*. To deal with complex domains and their deformations in an extremely flexible way, we exploit a general mesh deformation technique, in which deformations result from an additional FE problem describing either the behavior of the structure with respect to given inputs or an harmonic extension of boundary data. These are called *mesh-based variational methods*, and are often referred to as *solid-extension mesh moving techniques*. The corresponding meshes are also taken as a deformation of a reference mesh, without affecting the topology of the degrees of freedom. Denoting by Ω_h^0 the computational mesh over which the state problem is solved, a deformed volumetric mesh is obtained as $\Omega_h(\boldsymbol{\mu}) = \{\vec{x}_h \in \mathbb{R}^3 : \vec{x}_h(\boldsymbol{\mu}) = \vec{x}_h + \vec{d}(\boldsymbol{\mu}), \vec{x}_h \in \Omega_h^0\}$, where the nodes position is modified (so that $\Omega_h(\boldsymbol{\mu})$ conforms to the updated boundary) while keeping the mesh connectivity fixed; the domain deformation $\vec{d}(\boldsymbol{\mu})$ can then be approximated by an RB method, before constructing the ROM for the fluid flow problem; see, e. g., [25, 42] for further details.

8.3.2 Monodomain system

To ensure the efficiency of ROMs when dealing with the parameterized coupled monodomain-ionic model (8.5) a generalization of the POD approach is envisaged, requiring the construction of *local RB spaces*. Additionally, hyperreduction techniques such as DEIM and its matrix version MDEIM are employed to enhance the construction of nonaffine and nonlinear terms. Regarding the PDE system (8.8) for the transmembrane potential, we assume that the RB approximation of the transmembrane potential at time t_n is expressed by a linear combination of the RB basis functions,

$$\mathbf{u}^n(\boldsymbol{\mu}) \approx \mathbb{V}\mathbf{u}_N^n(\boldsymbol{\mu}), \tag{8.13}$$

where $\mathbb{V} \in \mathbb{R}^{N_h \times N}$ collects the (degrees of freedom of the) basis functions.

If the gating variable has already been updated to its current value $\mathbf{w}^{n+1}(\boldsymbol{\mu})$ at time t^{n+1} by solving (8.7), the Galerkin-RB problem reads

$$\begin{aligned}\left(\frac{\mathbb{M}_N(\boldsymbol{\mu})}{\Delta t} + \mathbb{A}_N(\boldsymbol{\mu})\right)\mathbf{u}_N^{n+1} + \mathbb{V}^T\mathbf{I}_{\text{ion}}(\mathbb{V}\mathbf{u}_N^n, \mathbf{w}^{n+1}; \boldsymbol{\mu}) \\ = \frac{\mathbb{M}_N(\boldsymbol{\mu})}{\Delta t}\mathbf{u}_N^n + \mathbf{I}_{\text{app}}^{n+1}(\boldsymbol{\mu}), \quad n = 0, \dots, N_t - 1,\end{aligned} \tag{8.14}$$

where $\mathbb{A}_N(\boldsymbol{\mu}) = \mathbb{V}^T\mathbb{A}(\boldsymbol{\mu})\mathbb{V}$ and $\mathbb{M}_N(\boldsymbol{\mu}) = \mathbb{V}^T\mathbb{M}(\boldsymbol{\mu})\mathbb{V}$; also in this case, since the $\boldsymbol{\mu}$ dependence shown by these matrices is nonaffine, we rely on the MDEIM to get an approximate affine expansion.

Two issues arise when dealing with this problem:

1. *PDE–ODE coupling*. We can take advantage of the DEIM to avoid the evaluation of the full-order array $\mathbf{I}_{\text{ion}} \in \mathbb{R}^{N_h}$, which would compromise the overall ROM efficiency. Hence, we approximate

$$\mathbf{I}_{\text{ion}}(\mathbb{V}\mathbf{u}_n^n, \mathbf{w}_h^{n+1}; \boldsymbol{\mu}) \approx \tilde{\mathbf{I}}_{\text{ion}}(\boldsymbol{\mu}) = \sum_{q=1}^{m} \theta_q(t^n; \boldsymbol{\mu})\,\mathbf{z}_q \tag{8.15}$$

once $m \ll N_h$ $\boldsymbol{\mu}$-independent vectors and $\mathbf{z}_q \in \mathbb{R}^{N_h}$, $1 \le q \le m$, basis vectors have been calculated from a set of snapshots $\{\mathbf{I}_{\text{ion}}(\mathbb{V}\mathbf{u}_n^n(\boldsymbol{\mu}^k), \mathbf{w}_h^{n+1}(\boldsymbol{\mu}^k); \boldsymbol{\mu}^k),\ k = 1, \dots, N_s,\ \ell = 0, \dots, N_t - 1\}$; $\boldsymbol{\mu}$-dependent weights $\theta_q : \mathcal{P} \mapsto \mathbb{R}$ are then computed by imposing m interpolation constraints. Basis vectors are computed by means of POD [16], whereas the set of points (in the physical domain) where interpolation constraints are being imposed are iteratively selected by employing the so-called *magic points* algorithm [9, 40].

The ionic term in the potential equation can then be approximated by

$$\mathbb{V}^T\mathbf{I}_{\text{ion}}(\mathbb{V}\mathbf{u}^n(\boldsymbol{\mu}), \mathbf{w}^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \underbrace{\mathbb{V}^T\boldsymbol{\Phi}(\mathbb{P}^T\boldsymbol{\Phi})^{-1}}_{N_h \times m}\underbrace{\mathbf{I}_{\text{ion}}(\mathbb{P}^T\mathbb{V}\mathbf{u}^n(\boldsymbol{\mu}), \mathbb{P}^T\mathbf{w}^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu})}_{m \times 1},$$

where $\mathbf{\Phi} = [\mathbf{z}_1|\dots|\mathbf{z}_m] \in \mathbb{R}^{N_h \times m}$ and $\mathbb{P} = [\mathbf{e}_{\mathcal{I}_1} \mid \dots \mid \mathbf{e}_{\mathcal{I}_m}] \in \mathbb{R}^{N_h \times m}$, with $\mathbf{e}_{\mathcal{I}_i} = [0,\dots,0,1,0,\dots,0]^T \in \mathbb{R}^{N_h}$, $\mathcal{I}$ being the set of m interpolation indices $\mathcal{I} \subset \{1,\dots,N_h\}$, with $|\mathcal{I}| = m$. Note that the matrix $\Phi_m = \mathbb{V}^T\mathbf{\Phi}(\mathbb{P}^T\mathbf{\Phi})^{-1}$ is $\boldsymbol{\mu}$-independent and can be assembled once for all. As a matter of fact, this procedure also enables a reduction of the computational complexity entailed by the solution of the ODE system (8.7). Indeed, its pointwise approximation can be advanced in time only on the m degrees of freedom $\mathcal{I}_1,\dots,\mathcal{I}_m$, thus resulting in a reduced ODE system for the vector $\mathbf{w}_m^n = \mathbb{P}^T\mathbf{w}^n \in \mathbb{R}^m$.
Finally, the ROM for the monodomain system (8.5) reads as follows: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $(\mathbf{u}_N^{n+1}(\boldsymbol{\mu}), \mathbf{w}_m^{n+1}(\boldsymbol{\mu})) \in \mathbb{R}^{N_u} \times \mathbb{R}^m$ such that $\mathbf{u}_N^0(\boldsymbol{\mu})(\boldsymbol{\mu}) = \mathbb{V}^T\mathbf{u}_0(\boldsymbol{\mu})$, $\mathbf{w}_m^0 = \mathbb{P}^T\mathbf{w}_0(\boldsymbol{\mu})$, and, for $n = 0,\dots,N_t-1$,

$$\begin{aligned}
&\mathbf{w}_m^{n+1}(\boldsymbol{\mu}) - \Delta t\,\mathbf{g}(\mathbb{P}^T\mathbb{V}\mathbf{u}_N^n(\boldsymbol{\mu}), \mathbf{w}_m^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{w}_h^n(\boldsymbol{\mu}),\\
&\left(\frac{\mathbb{M}_N(\boldsymbol{\mu})}{\Delta t} + \mathbb{A}_N(\boldsymbol{\mu})\right)\mathbf{u}_N^{n+1}(\boldsymbol{\mu})\\
&\quad + \Phi_m\mathbf{I}_{\text{ion}}(\mathbb{P}^T\mathbb{V}\mathbf{u}_N^n(\boldsymbol{\mu}), \mathbf{w}_m^{n+1}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{\mathbb{M}_N(\boldsymbol{\mu})}{\Delta t}\mathbf{u}_N^n(\boldsymbol{\mu}) + \mathbb{V}^T\mathbf{I}_{\text{app}}^{n+1}(\boldsymbol{\mu}).
\end{aligned} \tag{8.16}$$

2. *RB space construction*. Relying on POD on global RB spaces yields accurate approximations only if very large dimensions (up to some hundreds) N and m of the POD expansion and of the DEIM approximation, respectively, are considered, ultimately yielding negligible speedups compared to the FOM. Indeed, parameterized problems in cardiac electrophysiology might easily yield solutions showing a remarkable variability over the parameter space, because of the huge sensitivity of the solution with respect to variations of parameters representing conduction velocities, fiber structures, and tissue anisotropy.
Multiple local subspaces must be generated when performing the RB approximation of the PDE solution, and the DEIM approximation can be used for the nonlinear term. Approximating the whole solution set by a series of subspaces of smaller dimension results in a more efficient approach than building a single subspace of larger dimension. For this reason, clustering (or partitioning) algorithms are employed, prior to performing POD, aiming at collecting snapshots (of both the solution and the nonlinear terms) into clusters; then, a local RB is built for each cluster through POD. Here we consider the approach proposed in [2] based on the *k-means* algorithm, to address the construction of local ROMs in the *state space* and further extended in [65, 3, 1]; see, e. g., [47] for further details.

Finally, we highlight that if other models of cellular bioelectric activity were considered, based on either a single ODE (such as FitzHugh–Nagumo, Rogers–McCulloch, or Mitchell–Schaeffer models) or a system of ODEs (such as the Fenton–Karma model), the construction of ROMs would not change; see, e. g., [17, 20] for a review.

8.4 Computation of wall shear stress in a carotid artery bifurcation

The first application we deal with is related to blood flow in a subject-specific three-dimensional carotid bifurcation, in terms of both flow variables and derived outputs of interest. The carotid bifurcation is located along the sides of the neck and supplies blood to the face and the brain [66]; it is composed by the common carotid artery (CCA), which then splits in the internal carotid artery (ICA) and the external carotid artery (ECA) (Figure 8.1a). In adult age, the carotid bifurcation might be affected by atherosclerosis, that is, a progressive narrowing of the artery, which might ultimately lead to stroke. Blood dynamics play an important role in the development of such disease, and one of the main indicators employed in the risk analysis is the distribution of the WSS on the vessel wall boundary Γ_w close to the bifurcation [59]. Numerical simulations can provide such quantitative results able to support clinicians and recent results are reported, e. g., in [35, 30]. The WSS is defined as

$$\vec{\tau}_w = (2\bar{\mu}\boldsymbol{\varepsilon}(\vec{u})\vec{n}) \cdot \vec{t} = 2\bar{\mu}(\boldsymbol{\varepsilon}(\vec{u})\vec{n} - (\boldsymbol{\varepsilon}(\vec{u})\vec{n} \cdot \vec{n})\vec{n}),$$

where $\vec{n}$ and $\vec{t}$ are the (outer) normal and tangential unit vectors on Γ_w, respectively, $\boldsymbol{\varepsilon}$ is the strain tensor, and ν is the dynamic viscosity of the fluid.

Here we exploit the proposed ROM work flow to investigate the behavior of blood flow in different virtual scenarios, described in terms of parameterized domains and inlet boundary conditions. As a result, the WSS distribution will depend on parameters, too.

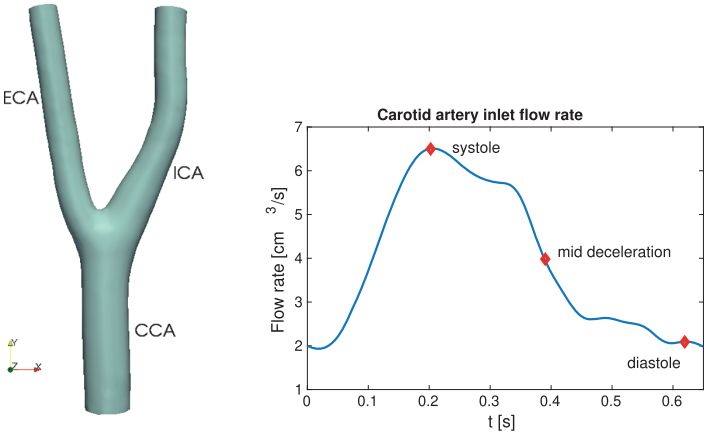


Figure 8.1: Computational domain with common carotid artery (CCA), internal carotid artery (ICA), and external carotid artery (ECA) (left) and reference inlet flow rate $Q_{\mathrm{CCA}}(t)$ [$\mathrm{cm}^3\,\mathrm{s}^{-1}$] with highlighted systole, mid deceleration, and diastole phases (right).

We denote by Γ_{in} the CCA inlet boundary portion, located at the bottom of the bifurcation in Figure 8.1a; Γ_{out} is given by the two ECA and ICA outflow boundaries, located on its top; finally, $\Gamma_w = \partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}})$. By following the setup employed in [25], at the CCA inlet boundary we prescribe a parameterized flow rate $Q_{\text{CCA}}(t;\boldsymbol{\mu})$, obtained as a suitable modification of the reference flow rate $Q^0_{\text{CCA}}(t)$, acquired from echo-color Doppler and reported in Figure 8.1b for a single heartbeat. The resulting inlet velocity $\vec{g}_{\text{NS}}(\boldsymbol{\mu})$ is a parabolic function in the normal direction to $\Gamma_{\text{in}}(\boldsymbol{\mu})$ and vanishing in the tangential ones, such that

$$\int_{\Gamma_{\text{in}}} \vec{g}_{\text{NS}}(t;\boldsymbol{\mu}) \cdot \vec{n} \, \mathrm{d}\Gamma_{\text{in}} = Q_{\text{CCA}}(t;\boldsymbol{\mu}) = \mu_2 \, Q^0_{\text{CCA}}(t). \tag{8.17}$$

For the case at hand $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{P} = [0.2,\ 0.4] \times [0.75, 1.0] \subset \mathbb{R}^2$; the parameter μ_1 tunes the narrowing of the ICA by loading a stress proportional to μ_1 in the region shown in Figure 8.2a. This simulates the effect of a stenosis obstructing the vessel; the larger μ_1, the more emphasized the deformation. See [25] for a detailed description on the way this geometrical parameterization is built. On the other hand, μ_2 determines the magnitude of the inlet flow rate; see (8.17). The radius at the inlet boundary at the entrance of the CCA measures approximately 0.27 cm, leading to a peak of the inlet velocity profile of approximately 59 cm s^{-1}, when $\mu_2 = 1$, during the systolic phase. Two examples of deformation (with respect to the reference domain) are reported in Figure 8.2 for different instances of the parameter $\boldsymbol{\mu}_1 = (0.38, 0.975)$ and $\boldsymbol{\mu}_3 = (0.21, 0.7625)$. Finally, the blood kinematic viscosity is $\nu = 0.035$ cm^2s^{-1}; as a result, Re $\approx$ 450. Taylor–Hood ($\mathbb{P}^2 - \mathbb{P}^1$) FEs are employed for the spatial discretization, leading to $N^u_h = 248{,}019$ degrees of freedom for the velocity and $N^p_h = 11{,}911$ for the pressure, respectively; as a result, $N_h = N^u_h + N^p_h = 259{,}930$. A BDF2 scheme with $\Delta t = 0.01$ has been considered for the time discretization, taking $T = 0.64$ seconds as

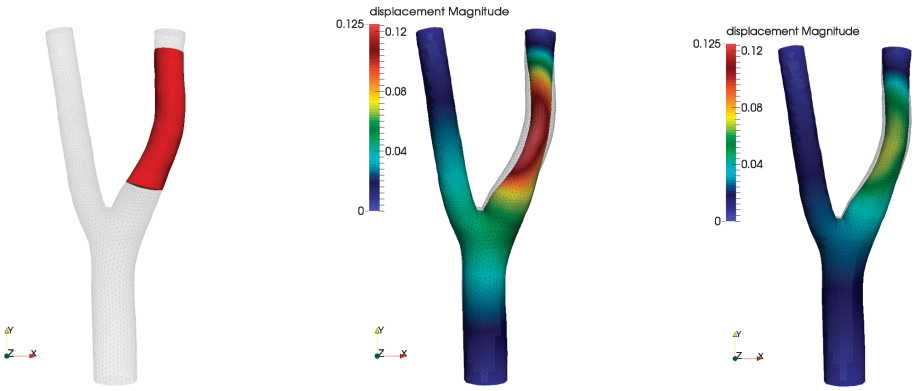


Figure 8.2: Region where the stress is loaded to deform the mesh (left) and examples of deformation for two instances of the parameter $\boldsymbol{\mu}_1 = (0.38,\ 0.975)$ (center) and $\boldsymbol{\mu}_3 = (0.21,\ 0.7625)$ (right).

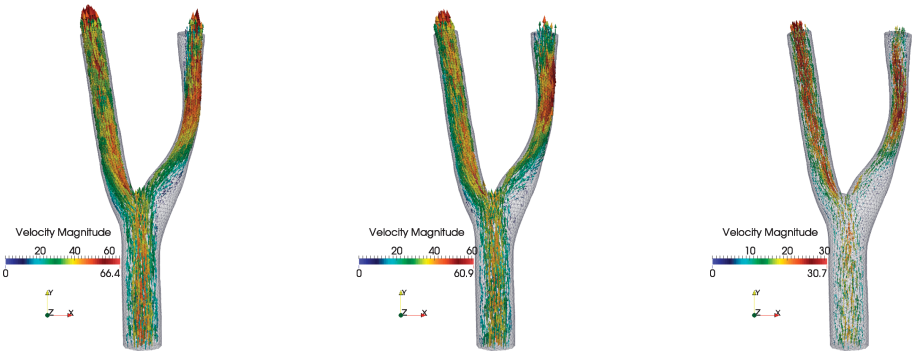


Figure 8.3: Velocity vector for $\boldsymbol{\mu}_1 = (0.38,\ 0.975)$ at different time steps $t = 0.2,\ 0.35,\ 0.5$.

final time to simulate an entire heartbeat. Numerical simulations have been carried out by employing 32 cores.[1]

Regarding system approximation, the MDEIM and DEIM applied to the arrays appearing in (8.11) yield 277 matrix operators and 16 right-hand side vectors. Regarding state reduction, the RB matrices $\mathbb{V}_u$, $\mathbb{V}_p$, $\mathbb{V}_s$ for velocity, pressure, and supremizing functions, respectively, are built with POD; this latter retains $N_u = 836$, $N_p = 506$, and $N_s = 742$ basis functions, respectively. Compared to the dimension $N_h = 259{,}930$ of the original FOM, the reduction in the system size is of about 125.

Examples of ROM solutions for different values of the parameters, computed with the ROM, are reported in Figures 8.3 and 8.4, with an error of the order of 0.1 % if compared with the FE simulation. On average, the Navier–Stokes equations are solved by the ROM with a computational cost of 4.05 seconds per time step, yielding a speedup

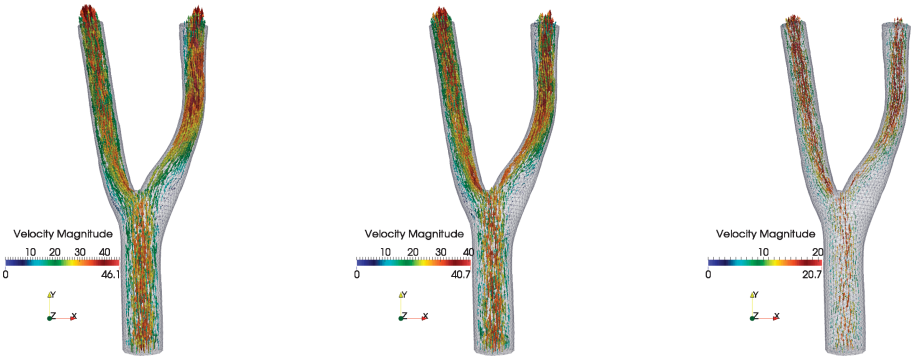


Figure 8.4: Velocity vector for $\boldsymbol{\mu}_3 = (0.21,\ 0.7625)$ at different time steps $t = 0.2,\ 0.35,\ 0.5$.

1 Computations have been performed on the Piz-Daint cluster at the Swiss National Supercomputing Center with Intel® Xeon® E5-2695 v4 @ 2.10 GHz and 64 Gb RAM.

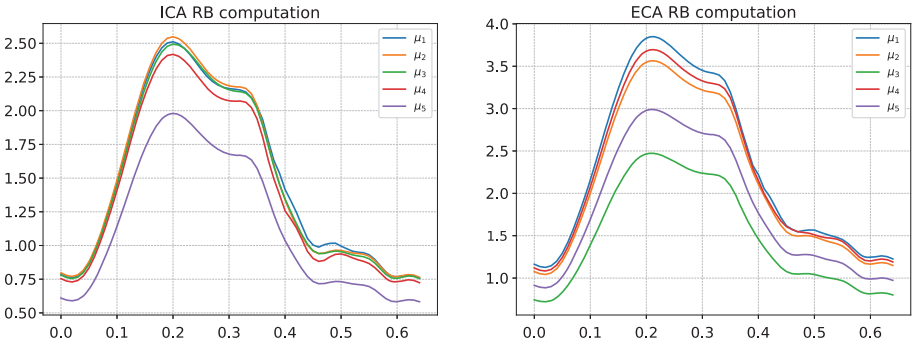


Figure 8.5: Computed flow rate at ICA (left) and ECA (right) for $\boldsymbol{\mu}_1 = (0.38,\ 0.975)$, $\boldsymbol{\mu}_2 = (0.35,\ 0.9375)$, $\boldsymbol{\mu}_3 = (0.21,\ 0.7625)$, $\boldsymbol{\mu}_4 = (0.38,\ 0.9375)$, $\boldsymbol{\mu}_5 = (0.38,\ 0.7625)$.

factor of about 20 with respect to the FOM. The significant gain in the speedup allows for the real-time simulation performed by clinicians, since in only few minutes the computation can be carried out for any new scenario.

The velocity pattern is affected by the parameter values, also influencing the flow rate at the outlet boundaries: In Figure 8.5 we report the outflow rate at the ICA and ECA boundaries as a function of time, for different parameter values. The physical parameter μ_2 mainly impacts the absolute value of the flow rate, whereas the geometrical parameter μ_1 mostly affects the way blood flows split into the two branches: the smaller μ_1, the larger the flow rate through the ICA.

Finally, in Figures 8.6 and 8.7 the WSS magnitude distribution is reported for different values of the parameters and times; as expected, the WSS is higher during the systolic peak and concentrated close to the bifurcation.

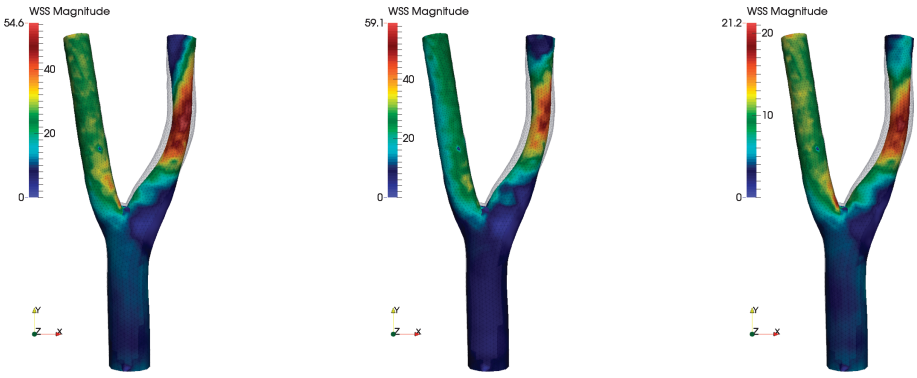


Figure 8.6: WSS for $\boldsymbol{\mu}_1 = (0.38,\ 0.975)$ at different time steps $t = 0.2,\ 0.35,\ 0.5$.

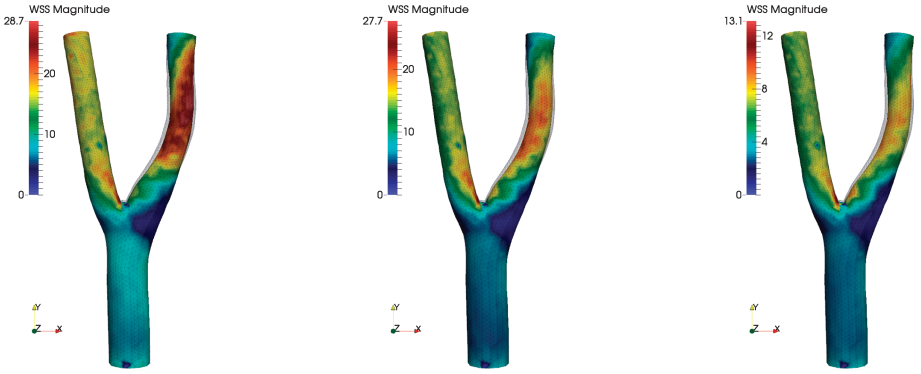


Figure 8.7: WSS for $\boldsymbol{\mu}_3 = (0.21,\ 0.7625)$ at different time steps $t = 0.2,\ 0.35,\ 0.5$.

8.5 Evaluation of activation maps in cardiac electrophysiology

The second application we consider is related to the efficient evaluation of activation maps over a subject-specific left ventricle geometry. These outputs are the *virtual* counterpart of epicardial or endocardial potential recordings (electrograms) obtained by means of (unipolar or bipolar) catheters located inside a cardiac chamber, or on its external surface. Common relevant features extracted from these measurements are, e. g., *voltage maps*, showing the distribution of the electric potential at any given time, and *activation maps*, providing information about the time when the electric wavefront reaches a given point. These information are crucial in the clinical practice, for instance when treating cardiac arrhythmias by radio-frequency catheter ablation. Target sites for ablation, often consisting of slow-conducting narrow isthmuses bordered by areas of scar tissue, are indeed identified by endocardial voltage and activation maps.

The efficient numerical evaluation of several different scenarios in terms of electric potential, by means of accurate ROMs, can open new paths to address the propagation of input uncertainty on the output and the systematic evaluation of the impact of (optimized) standard intervention procedures, aiming at optimizing them. For the case at hand, we consider the monodomain model (8.5) with the Aliev–Panfilov model (8.6) for estimating the effect of (i) anisotropic conductivity, (ii) ion dynamics, and (iii) activation times on the electric conduction.

We consider $p = 9$ parameters for this test case: the two conductivities $\sigma_l \in [12.9 \cdot 0.1, 12.9 \cdot 0.15]$ and $\sigma_t \in [12.9 \cdot 0.05, 12.9 \cdot 0.1]$; five parameters affecting the ionic model, $K \in [7, 9]$, $a, b \in [0.05, 0.15]$, $c_1 \in [0.1, 0.2]$, and $\varepsilon \in [0.005, 0.02]$; and finally, two parameters affecting the electric activation, $t_1 \in [5, 10]$ and $t_2 \in [10, 15]$. Regarding the

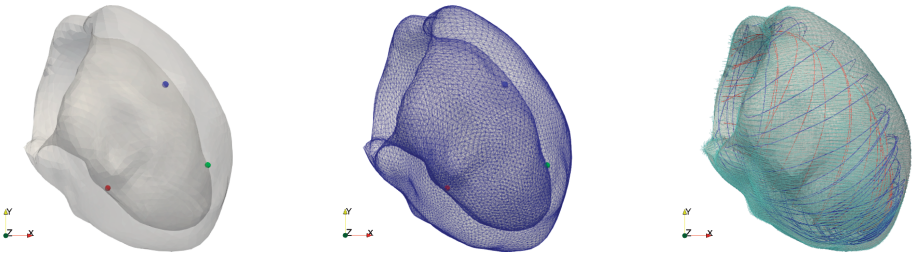


Figure 8.8: Left ventricle geometry: activation points (left), computational mesh (center), and fiber orientation (right).

latter, the three applied current stimuli $\{\phi_i(\vec{x})\}_{i=1}^3$ are

$$\phi_i(\vec{x}) = \frac{1}{2(2\pi)^{\frac{3}{2}}} \exp\left(-\frac{\|\vec{x} - \vec{x}_i\|^2}{4}\right), \quad i = 1, 2, 3.$$

The location of the three activation points, together with the computational mesh of the subject-specific left ventricle geometry we consider, and the fiber orientation are reported in Figure 8.8c. The left ventricle geometry, extracted from the atlas of cardiac geometries described in [31], has been discretized using a three-dimensional mesh with $N_h = 24{,}660$ vertices and 105,904 elements. The time interval is $[0, 600\ \text{ms}]$ and the time step is $\Delta t = 0.25$ ms. A single query to the FOM, based on linear FEs, takes about 23.5 minutes to be computed.[2]

We assess the computational performance and the accuracy of the ROM built by considering the *k-means* algorithm to address the construction of local ROMs in the *state space*. We start from a training sample of 25 parameter vectors, leading to a set of training snapshots of dimension $N_h \times 6 \cdot 10^4$. The 15 clusters formed by the *k-means* algorithm subdivide the snapshots mainly with respect to time. Indeed, as shown in Figure 8.9, the clusters' centroids (or barycenters) reflect, roughly speaking, the evolution of the electric potential over the time interval.

We then compute the relative error

$$e_r = \frac{\sqrt{\sum_{n=1}^{N_t} \|\mathbf{u}^n - \mathbf{u}_N^n\|_{H^1(\Omega)}^2}}{\sqrt{\sum_{n=1}^{N_t} \|\mathbf{u}^n\|_{H^1(\Omega)}^2}},$$

where $\|\cdot\|_{H^1(\Omega)}$ denotes the $H^1(\Omega)$-norm, on three selected test parameter vectors,

$$\boldsymbol{\mu}_1^* = [1.29, 1.29, 7.023, 0.0837, 0.1162, 0.0179, 10, 10]^T, \tag{8.18}$$

2 All timings are obtained by performing calculations on an Intel(R) Core i7-8700K CPU with 64 Gb DDR4 2666 MHz RAM.

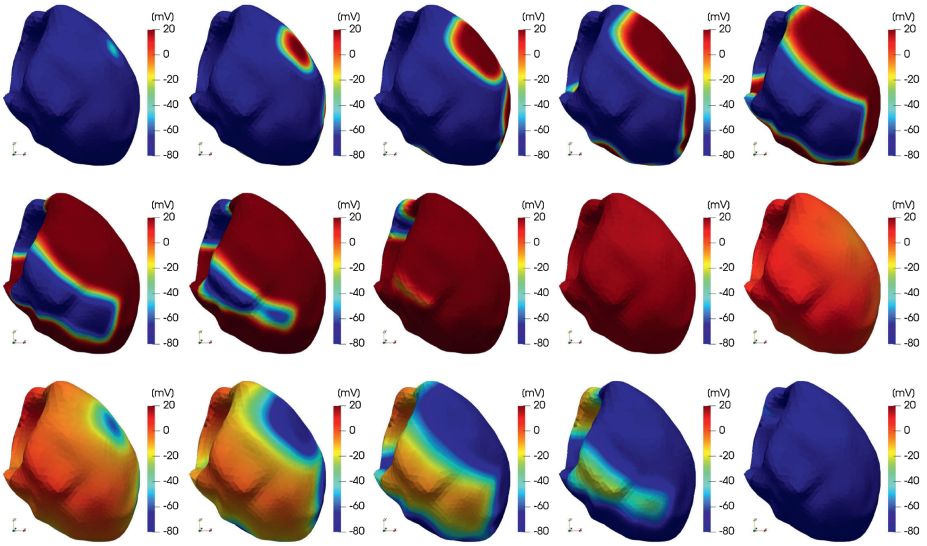


Figure 8.9: Centroids computed by the *k-means* algorithm on the training set.

$$\boldsymbol{\mu}_2^* = [1.3968, 1.0333, 7.5259, 0.1154, 0.1689, 0.0175, 7.2527, 10.4191]^T, \tag{8.19}$$

$$\boldsymbol{\mu}_3^* = [1.4377, 1.2341, 7.3048, 0.1326, 0.1538, 0.02, 5.3909, 12.2134]^T, \tag{8.20}$$

which we also employ for the sake of visualization. Figure 8.10 shows the action potential for $\boldsymbol{\mu} = \boldsymbol{\mu}_i^*$, $i = 1, 2, 3$, at different time steps $t = 60$,100,320 ms.

Table 8.1: Computational performance on the three test parameter vectors $\boldsymbol{\mu}_i^*$, $i = 1, 2, 3$.

Parameters	FOM time	ROM time	Relative error	Speedup
$\boldsymbol{\mu}_1^*$	1409 s	26.1 s	0.0077	54×
$\boldsymbol{\mu}_2^*$	1419 s	33.2 s	0.005	43×
$\boldsymbol{\mu}_3^*$	1398 s	29.4 s	0.0099	48×

Table 8.1 shows that the local ROM provides an average speedup factor of about 48× compared to the FOM, ensuring at the same time a relative error smaller than 1%. The resulting POD bases on each cluster (obtained by considering a tolerance of 10^{-3} on the relative energy content of the discarded POD modes) have dimension ranging from 38 to 162. The same procedure is applied within the DEIM for the approximation of the nonlinear terms; in this case the dimension of the POD bases (obtained with a tolerance of 10^{-8}) ranges from 492 to 1221.

We then show how to recover two outputs of clinical interest from the computed solutions:

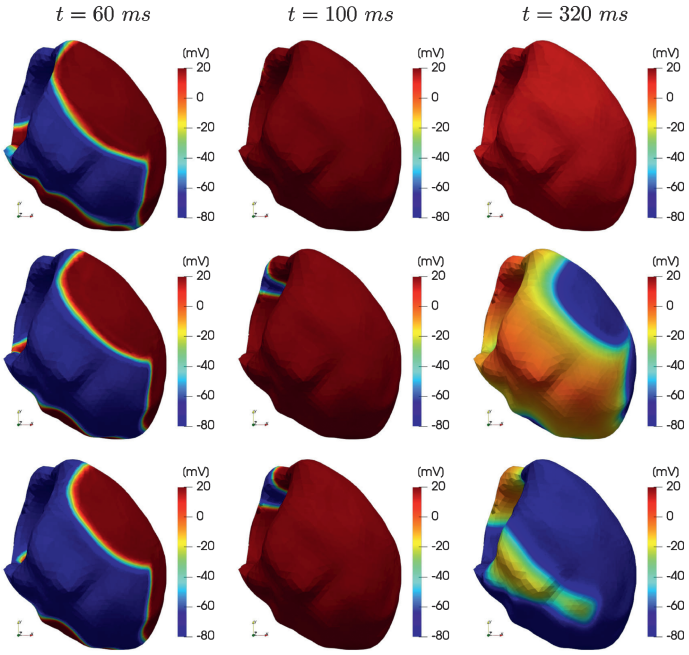


Figure 8.10: Action potential for the three test parameter vectors $\boldsymbol{\mu}_i^*$, $i = 1, 2, 3$, at different time steps $t = 60$,100,320 ms.

– The former is the *activation map*, which is obtained by evaluating the local activation time (LAT) at each vertex of the mesh; see Figure 8.11. The LAT at a spatial point is given by the time when the electric wavefront passes through that point, that is, when the maximum negative slope of the electric deflection is measured. Such a map allows one to understand if regions of slow conduction are present in the tissue, for instance.
– The latter is a set of six *simulated electrocardiograms (ECGs)*, representing the unipolar precordial (or chest) leads $V_1, \ldots, V_6$; these signals can be numerically approximated by integrating the projection of the heart vector ∇u onto the direction vector $\nabla(1/\|\vec{r}\|)$, i. e.,

$$V_i = \int_\Omega \nabla u \cdot \nabla\left(\frac{1}{\|\vec{r}\|}\right) d\Omega, \quad \vec{r} = \vec{x} - \vec{x}_i,\ i = 1, \ldots, 6,$$

$\vec{x}_i$ being the positions of the pseudo-electrodes [34, 57]. For the case at hand, we locate the pseudo-electrodes as reported in Figure 8.12 in order to mimic their position on the chest. ECG is a noninvasive test which conveys a large amount of information about the heart conditions. Variations on the uncertain ionic coefficients K, a, b, c_1, and ε_0 induce considerable changes in the T wave and the QT interval, as shown in Figure 8.13.

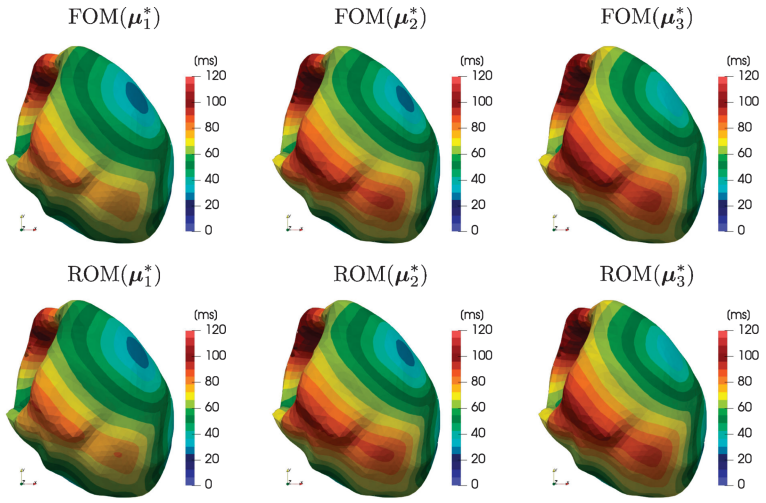


Figure 8.11: Activation maps for the three test parameter vectors $\boldsymbol{\mu}_i^*$, $i = 1, 2, 3$.

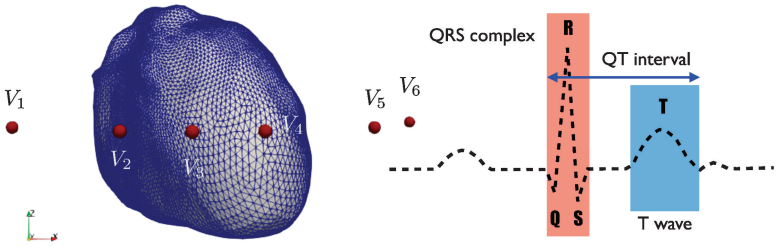


Figure 8.12: Position of the pseudo-electrodes used for the approximation of V_k, $k = 1, \ldots, 6$, and ECG scheme.

Also in this case, outputs of interest can be inexpensively evaluated taking advantage of the significant gain in the speedup provided by the ROM.

8.6 Conclusions and outlook

In this chapter we have shown two examples of applications of state-of-the-art ROM techniques to relevant problems in cardiovascular modeling, dealing with (i) the arterial fluid dynamics and (ii) the electric activity of the heart. In both cases, we suitably combined projection-based ROMs, built through POD, and hyperreduction techniques to enhance the assembling of the reduced-order problem. This allows computational speedup factors of about 20–50 with respect to the high-fidelity, FOM built by the Galerkin FE method, without substantially lowering the accuracy of the FOM approximation. The availability of efficient and reliable ROMs is thus of paramount

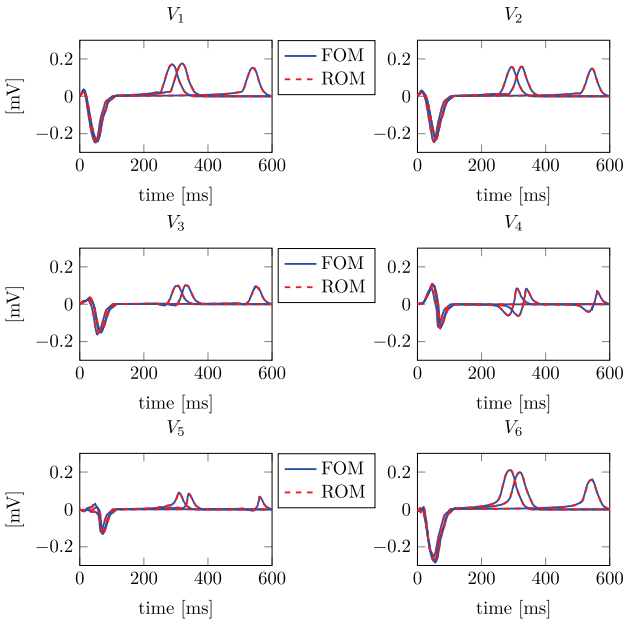


Figure 8.13: Numerical approximation of unipolar precordial leads V_k, $k = 1, \ldots, 6$, for $\boldsymbol{\mu}_i^*$, $i = 1, 2, 3$.

importance to enable parametric studies, sensitivity analysis, and uncertainty quantification in complex, subject-specific scenarios (e. g., [45, 32, 50, 55]), yet unaffordable with standard, high-fidelity techniques even on modern parallel architectures.

Despite huge efforts to enhance ROM efficiency still preserving their accuracy, several challenges remain open when dealing with cardiovascular applications. A nonexhaustive list includes, among others:

- to achieve fine temporal and spatial resolution, the training phase of ROMs (offline) as well as hyperreduction techniques becomes computationally intensive;
- new approaches are needed to address multiphysics and multiscale models, involving a wide range of spatio-temporal scales, more particularly to handle coupled problems;
- a full decoupling between the ROM and the FOM can be rather involved to obtain in the case of fine computational meshes and complex geometries, when dealing with hyperreduction techniques;
- intra-patient and inter-patient variability involves complex parameterizations of the model inputs;
- nonlinearity and high sensitivity of the solution with respect to parameter variations limit computational speedups.

Very often, several of these criticalities are simultaneously present in cardiovascular models, thus making the design of efficient and accurate ROMs a critical task.

On the other hand, an emerging strategy in the field of cardiovascular applications is grounded on the use of purely data-driven surrogate models or emulators, based, e. g., on machine learning (ML) techniques (e. g., [54, 48, 24, 22, 33]), like artificial neural networks, or Gaussian process regression. This approach is especially relevant to approximate input-output maps featuring low-dimensional outputs as quantities of interest. Despite their efficiency at testing time (once a training phase, usually expensive, has been performed) for the sake of output evaluations for new parameter instances, often these techniques lack interpretability. Besides, the lack of error indicators makes their construction tailored on the specific problem at hand. In this respect, the combination of physics-based models (among which we also include projection-based ROMs) with data-driven, ML-based techniques for the sake of efficiency looks promising in view of a future translation of ROMs in clinical practice.

Bibliography

[1] D. Amsallem and B. Haasdonk, PEBL-ROM: Projection-error based local reduced-order models, *Adv. Model. Simul. Eng. Sci.*, **3** (1) (2016), 6.

[2] D. Amsallem, M. J. Zahr, and C. Farhat, Nonlinear model order reduction based on local reduced-order bases, *Int. J. Numer. Methods Eng.*, **92** (10) (2012), 891–916.

[3] D. Amsallem, M. J. Zahr, and K. Washabaugh, Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction, *Adv. Comput. Math.*, **41** (5) (2015), 1187–1230.

[4] T. J. Baker, Mesh movement and metamorphosis, *Eng. Comput.*, **18** (3) (2002), 188–198.

[5] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, and R. Scrofani, Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization, *J. Comput. Phys.*, **315** (2016), 609–628.

[6] F. Ballarin, E. Faggiano, A. Manzoni, A. Quarteroni, G. Rozza, S. Ippolito, C. Antona, and R. Scrofani, Numerical modeling of hemodynamics scenarios of patient-specific coronary artery bypass grafts, *Biomech. Model. Mechanobiol.*, **16** (4) (2017), 1373–1399.

[7] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1136–1161.

[8] F. Ballarin and G. Rozza, POD–Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems, *Int. J. Numer. Methods Fluids*, **82** (12) (2016), 1010–1034.

[9] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math. Acad. Sci. Paris*, **339** (9) (2004), 667–672.

[10] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, vol. 15, SIAM, 2017.

[11] D. Bonomi, A. Manzoni, and A. Quarteroni, A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics, *Comput. Methods Appl. Mech. Eng.*, **324** (2017), 300–326.

[12] M. Boulakia, E. Schenone, and J.-F. Gerbeau, Reduced-order modeling for cardiac electrophysiology. Application to parameter identification, *Int. J. Numer. Methods Biomed. Eng.*, **28** (6–7) (2012), 727–744.

[13] S. Buoso, A. Manzoni, H. Alkadhi, A. Plass, A. Quarteroni, and V. Kurtcuoglu, Reduced-order modeling of blood flow for non-invasive functional evaluation of coronary artery disease in clinical routine, *Biomech. Model. Mechanobiol.*, **18** (6) (2019), 1867–1881.

[14] D. Chapelle, A. Gariah, P. Moireau, and J. Sainte-Marie, A Galerkin strategy with Proper Orthogonal Decomposition for parameter-dependent problems – Analysis, assessments and applications to parameter estimation, *ESAIM: Math. Model. Numer. Anal.*, **47** (6) (2013), 1821–1843.

[15] D. Chapelle, A. Gariah, and J. Sainte-Marie, Galerkin approximation with Proper Orthogonal Decomposition: new error estimates and illustrative examples, *ESAIM: Math. Model. Numer. Anal.*, **46** (2012), 731–757.

[16] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[17] R. Clayton, O. Bernus, E. Cherry, H. Dierckx, F. Fenton, L. Mirabella, A. Panfilov, F. Sachse, G. Seemann, and H. Zhang, Models of cardiac tissue electrophysiology: Progress, challenges and open questions, *Prog. Biophys. Mol. Biol.*, **104** (1) (2011), 22–48.

[18] P. Colli Franzone, L. Pavarino, and B. Taccardi, Simulating patterns of excitation, repolarization and action potential duration with cardiac bidomain and monodomain models, *Math. Biosci.*, **197** (1) (2005), 35–66.

[19] P. Colli Franzone and L. F. Pavarino, A parallel solver for reaction–diffusion systems in computational electrocardiology, *Math. Models Methods Appl. Sci.*, **14** (06) (2004), 883–911.

[20] P. Colli Franzone, L. F. Pavarino, and S. Scacchi, *Mathematical Cardiac Electrophysiology*, *MS&A*, vol. 13, Springer, 2014.

[21] C. Corrado, J. Lassoued, M. Mahjoub, and N. Zemzemi, Stability analysis of the POD reduced order method for solving the bidomain model in cardiac electrophysiology, *Math. Biosci.*, **272** (2016), 81–91.

[22] F. Costabal, K. Matsuno, J. Yao, P. Perdikaris, and E. Kuhl, Machine learning in drug development: Characterizing the effect of 30 drugs on the QT interval using Gaussian process regression, sensitivity analysis, and uncertainty quantification, *Comput. Methods Appl. Mech. Eng.*, **348** (2019), 313–333.

[23] N. Dal Santo, S. Deparis, A. Manzoni, and A. Quarteroni, An algebraic least squares reduced basis method for the solution of nonaffinely parametrized Stokes equations, *Comput. Methods Appl. Mech. Eng.*, **344** (2019), 186–208.

[24] N. Dal Santo, S. Deparis, and L. Pegolotti, Data driven approximation of parametrized PDEs by Reduced Basis and Neural Networks, arXiv preprint arXiv:1904.01514, 2019.

[25] N. Dal Santo and A. Manzoni, Hyper-reduced order models for parametrized unsteady Navier–Stokes equations on domains with variable shape, *Adv. Comput. Math.*, **45** (5) (2019), 2463–2501.

[26] M. Ethier and Y. Bourgault, Semi-implicit time-discretization schemes for the bidomain model, *SIAM J. Numer. Anal.*, **46** (5) (2008), 2443–2468.

[27] J.-F. Gerbeau, D. Lombardi, and E. Schenone, Reduced order model in cardiac electrophysiology with approximated Lax pairs, *Adv. Comput. Math.*, **41** (5) (2015), 1103–1130.

[28] S. Goktepe and E. Kuhl, Computational modeling of cardiac electrophysiology: a novel finite element approach, *Int. J. Numer. Methods Eng.*, **79** (2) (2009), 156–178.

[29] L. Grinberg, A. Yakhot, and G. Karniadakis, Analyzing transient turbulence in a stenosed carotid artery by proper orthogonal decomposition, *Ann. Boomed. Eng.*, **37** (11) (2009), 2200–2217.

[30] B. Guerciotti, C. Vergara, L. Azzimonti, L. Forzenigo, A. Buora, P. Biondetti, and M. Domanin, Computational study of the fluid-dynamics in carotids before and after endarterectomy, *J. Biomech.*, **49** (1) (2016), 26–38.

[31] C. Hoogendoorn, N. Duchateau, D. Sánchez-Quintana, T. Whitmarsh, F. M. Sukno, M. De Craene, K. Lekadir, and A. F. Frangi, A high-resolution atlas and statistical model of the human heart from multislice CT, *IEEE Trans. Med. Imaging*, **32** (1) (2013), 28–44.

[32] D. Hurtado, S. Castro, and P. Madrid, Uncertainty quantification of two models of cardiac electromechanics, *Int. J. Numer. Methods Biomed. Eng.*, **33** (12) (2017), e2894.

[33] G. Kissas, Y. Yang, E. Hwuang, W. Witschey, J. Detre, and P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, *Comput. Methods Appl. Mech. Eng.*, **358** (2020), 112623.

[34] S. Krishnamoorthi, L. E. Perotti, N. P. Borgstrom, O. A. Ajijola, A. Frid, A. V. Ponnaluri, J. N. Weiss, Z. Qu, W. S. Klug, D. B. Ennis, and A. Garfinkel, Simulation methods and validation criteria for modeling cardiac ventricular electrophysiology, *PLoS ONE*, **9** (12) (2014), 1–29.

[35] R. M. Lancellotti, C. Vergara, L. Valdettaro, S. Bose, and A. Quarteroni, Large eddy simulations for blood dynamics in realistic stenotic carotids, *Int. J. Numer. Methods Biomed. Eng.*, **33** (11) (2017).

[36] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza, Boundary control and shape optimization for the robust design of bypass anastomoses under uncertainty, *ESAIM: Math. Model. Numer. Anal.*, **47** (4) (2013), 1107–1131.

[37] T. Lassila, A. Quarteroni, and G. Rozza, A reduced basis model with parametric coupling for fluid-structure interaction problem, *SIAM J. Sci. Comput.*, **34** (2) (2012), A1187–A1213.

[38] S.-W. Lee, L. Antiga, J. D. Spence, and D. A. Steinman, Geometry of the carotid bifurcation predicts its exposure to disturbed flow, *Stroke*, **39** (8) (2008), 2341–2347.

[39] S.-W. Lee, L. Antiga, and D. A. Steinman, Correlations among indicators of disturbed flow at the normal carotid bifurcation, *J. Biomech. Eng.*, **131** (6) (2009), 061013.

[40] Y. Maday, N. C. Nguyen, A. T. Patera, and S. H. Pau, A general multipurpose interpolation procedure: the magic points, *Commun. Pure Appl. Anal.*, **8** (1) (2009), 383–404.

[41] A. Manzoni, D. Bonomi, and A. Quarteroni, Reduced order modeling for cardiac electrophysiology and mechanics: New methodologies, challenges and perspectives, in D. Boffi, L. Pavarino, G. Rozza, S. Scacchi, and C. Vergara (eds.) *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*, SEMA SIMAI Springer Series, pp. 115–166, Springer International Publisher, 2018.

[42] A. Manzoni and F. Negri, Efficient reduction of PDEs defined on domains with variable shape, in P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban (eds.) *Model Reduction of Parametrized Systems*, vol. 17, pp. 183–199, Springer, Cham, 2017.

[43] A. Manzoni, A. Quarteroni, and G. Rozza, Model reduction techniques for fast blood flow simulation in parametrized geometries, *Int. J. Numer. Methods Biomed. Eng.*, **28** (6–7) (2012), 604–625.

[44] A. Manzoni, A. Quarteroni, and G. Rozza, Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques, *Int. J. Numer. Methods Fluids*, **70** (5) (2012), 646–670.

[45] G. Mirams, P. Pathmanathan, R. Gray, P. Challenor, and R. Clayton, Uncertainty and variability in computational and mathematical models of cardiac physiology, *J. Physiol.*, **594** (23) (2016), 6833–6847.

[46] F. Negri, A. Manzoni, and D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, *J. Comput. Phys.*, **303** (2015), 431–454.

[47] S. Pagani, A. Manzoni, and A. Quarteroni, Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method, *Comput. Methods Appl. Mech. Eng.*, **340** (2018), 530–558.

[48] M. Peirlinck, F. Costabal, K. Sack, J. Choy, G. Kassab, J. Guccione, M. De Beule, P. Segers, and E. Kuhl, Using machine learning to characterize heart failure across the scales, *Biomech. Model. Mechanobiol.*, **18** (6) (1987–2001), 2019.

[49] G. Plank, M. Liebmann, R. W. dos Santos, E. J. Vigmond, and G. Haase, Algebraic multigrid preconditioner for the cardiac bidomain model, *IEEE Trans. Biomed. Eng.*, **54** (4) (2007), 585–596.

[50] A. Quaglino, S. Pezzuto, P.-S. Koutsourelakis, A. Auricchio, and R. Krause, Fast uncertainty quantification of activation sequences in patient-specific cardiac electrophysiology meeting clinical time constraints, *Int. J. Numer. Methods Biomed. Eng.*, **34** (7) (2018), e2985.

[51] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-Baier, Integrated heart – coupling multiscale and multiphysics models for the simulation of the cardiac function, *Comput. Methods Appl. Mech. Eng.*, **314** (2017), 345–407.

[52] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations. An Introduction*, Springer International Publishing, 2016.

[53] A. Quarteroni, A. Manzoni, and C. Vergara, The cardiovascular system: Mathematical modeling, numerical algorithms, clinical applications, *Acta Numer.*, **26** (2017), 365–590.

[54] M. Raissi, A. Yazdani, and G. E. Karniadakis, Hidden fluid mechanics: A Navier–Stokes informed deep learning framework for assimilating flow visualization data, arXiv preprint arXiv:1808.04327, 2018.

[55] R. Rodriguez-Cantano, J. Sundnes, and M. Rognes, Uncertainty in cardiac myofiber orientation and stiffnesses dominate the variability of left ventricle deformation response, *Int. J. Numer. Methods Biomed. Eng.*, **35** (5) (2019), e3178.

[56] G. Rozza, D. Huynh, and A. Manzoni, Reduced basis approximation and error bounds for Stokes flows in parametrized geometries: roles of the inf–sup stability constants, *Numer. Math.*, **125** (1) (2013), 115–152.

[57] F. Sahli Costabal, J. Yao, and E. Kuhl, Predicting drug-induced arrhythmias by multiscale modeling, *Int. J. Numer. Methods Biomed. Eng.*, **34** (5) (2018), 29–64.

[58] P. P. Sengupta, J. Korinek, M. Belohlavek, J. Narula, M. A. Vannan, A. Jahangir, and B. K. Khandheria, Left ventricular structure and function: basic science for cardiac imaging, *J. Am. Coll. Cardiol.*, **48** (10) (2006), 1988–2001.

[59] C. Slager, J. Wentzel, F. Gijsen, A. Thury, A. Van der Wal, J. Schaar, and P. Serruys, The role of shear stress in the destabilization of vulnerable plaques and related therapeutic implications, *Nat. Rev. Cardiol.*, **2** (9) (2005), 456.

[60] M. L. Staten, S. J. Owen, S. M. Shontz, A. G. Salinger, and T. S. Coffey, A comparison of mesh morphing methods for 3D shape optimization, in *Proceedings of the 20th International Meshing Roundtable*, pp. 293–311, Springer, 2011.

[61] K. Stein, T. Tezduyar, and R. Benney, Mesh moving techniques for fluid-structure interactions with large displacements, *J. Appl. Mech.*, **70** (1) (2003), 58–63.

[62] K. Stein, T. E. Tezduyar, and R. Benney, Automatic mesh update with the solid-extension mesh moving technique, *Comput. Methods Appl. Mech. Eng.*, **193** (21–22) (2004), 2004–2032.

[63] J. Stroud, S. Berger, and D. Saloner, Numerical analysis of flow through a severely stenotic carotid artery bifurcation, *J. Biomech. Eng.*, **124** (1) (2002), 9–20.

[64] T. Tezduyar, M. Behr, S. Mittal, and A. Johnson, Computation of unsteady incompressible flows with the stabilized finite element methods: Space-time formulations, iterative strategies and massively parallel implementations, in *New Methods in Transient Analysis*, vol. 246/AMD, pp. 7–24, ASME, New York, 1992.

[65] K. Washabaugh, D. Amsallem, M. Zahr, and C. Farhat, Nonlinear model reduction for CFD problems using local reduced order bases, *AIAA Paper 2012-2686*, pp. 1–16, 2012.

[66] D. M. Wootton and D. N. Ku, Fluid mechanics of vascular systems, diseases, and thrombosis, *Annu. Rev. Biomed. Eng.*, **1** (1) (1999), 299–329.

[67] C. K. Zarins, D. P. Giddens, B. Bharadvaj, V. S. Sottiurai, R. F. Mabon, and S. Glagov, Carotid bifurcation atherosclerosis. Quantitative correlation of plaque localization with flow velocity profiles and wall shear stress, *Circ. Res.*, **53** (4) (1983), 502–514.

Jean-Christophe Loiseau, Steven L. Brunton, and Bernd R. Noack

9 From the POD-Galerkin method to sparse manifold models

Abstract: Reduced-order models are essential for the accurate and efficient prediction, estimation, and control of complex systems. This is especially true in fluid dynamics, where the fully resolved state space may easily contain millions or billions of degrees of freedom. Because these systems typically evolve on a low-dimensional attractor, model reduction is defined by two essential steps: (1) identifying a good state space for the attractor and (2) identifying the dynamics on this attractor. The leading method for model reduction in fluids is Galerkin projection of the Navier–Stokes equations onto a linear subspace of modes obtained via proper orthogonal decomposition (POD). However, there are serious challenges in this approach, including truncation errors, stability issues, difficulty handling transients, and mode deformation with changing boundaries and operating conditions. Many of these challenges result from the choice of a linear POD subspace in which to represent the dynamics. In this chapter, we describe an alternative approach, feature-based manifold modeling (FeMM), in which the low-dimensional attractor and nonlinear dynamics are characterized from typical experimental data: time-resolved sensor data and optional nontime-resolved particle image velocimetry (PIV) snapshots. FeMM consists of three steps: First, the sensor signals are lifted to a dynamic feature space. Second, we identify a sparse human-interpretable nonlinear dynamical system for the feature state based on the sparse identification of nonlinear dynamics (SINDy). Third, if PIV snapshots are available, a local linear mapping from the feature state to the velocity field is performed to reconstruct the full state of the system. We demonstrate this approach, and compare with POD-Galerkin modeling, on the incompressible two-dimensional flow around a circular cylinder. Best practices and perspectives for future research are also included, along with open-source code for this example.

Keywords: Reduced-order models, manifold, SINDy, dynamical systems, nonlinear dynamics

MSC 2010: 70K50, 70K70, 76D25, 76E30

Jean-Christophe Loiseau, Laboratoire DynFluid, Arts et Métiers ParisTech, 75013 Paris, France
Steven L. Brunton, Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA
Bernd R. Noack, Center for Turbulence Control, Harbin Institute of Technology, Shenzhen Campus, Room 311 of C block HIT Campus, University Town, Xili, Shenzhen, 518055 People's Republic of China; and Institut für Strömungsmechanik und Technische Akustik (ISTA), Technische Universität Berlin, Müller-Breslau-Straße 8, 10623 Berlin, Germany

https://doi.org/10.1515/9783110499001-009

9.1 Introduction

Understanding, modeling, and controlling complex fluid flows is a central focus in many scientific, technological, and industrial applications, including energy (e. g., wind, tidal, and combustion), transportation (e. g., planes, trains, and automobiles), security (e. g., airborne contamination), and medicine (e. g., artificial hearts and artificial respiration). Improved models of engineering flows have the potential to dramatically improve performance in these systems through optimization and control, resulting in practical gains such as drag reduction, lift increase, and mixing enhancement [38, 21, 98, 85, 25]. Although the Navier–Stokes equations provide a detailed mathematical model, this representation may be difficult to use for engineering design, optimization, and control. Instead, they are commonly discretized into a high-dimensional, nonlinear dynamical system with many degrees of freedom and multiscale interactions. These equations are nonetheless expensive to simulate, making them unwieldy for iterative optimization or in-time control. They may also obscure the underlying physics, which often evolves on a low-dimensional attractor [49, 77]. The various fidelities of model description were described by [115]: *white-box* describes an accurate evolution equation based on first principles (e. g., Navier–Stokes discretization), *gray-box* describes a low-dimensional model approximating the full state (e. g., proper orthogonal decomposition [POD]-Galerkin models), and *black-box* describes input–output models that lack a connection to the full state space (e. g., neural networks).

In the following, we outline related reduced-order models as our point of departure in Section 9.1.1 and foreshadow proposed innovations of this study in Section 9.1.2.

9.1.1 Related reduced-order models as point of departure

Reduced-order models provide low-dimensional descriptions of the underlying fluid behavior in a compact and computationally efficient representation. This is illustrated in Figure 9.1, where, starting from full-state velocity snapshots obtained from direct numerical simulation, one extracts the leading coherent structures in order to obtain a low-dimensional representation of the system's dynamics. There are many techniques for reduced-order modeling, ranging from physical reductions to purely data-driven methods, and nearly everything in between. POD [100, 14, 49] provides a low-rank modal decomposition of fluid flow field data, extracting the most energetic modes. It is then possible to Galerkin project the Navier–Stokes equations onto these modes, resulting in an approximate, low-dimensional model in terms of mode coefficients [78, 28]. POD-Galerkin models are widely used, as they are interpretable, gray-box models, and it is straightforward to reconstruct the high-dimensional flow field from the low-dimensional model via POD modes. The first pioneering example of [4] featured wall

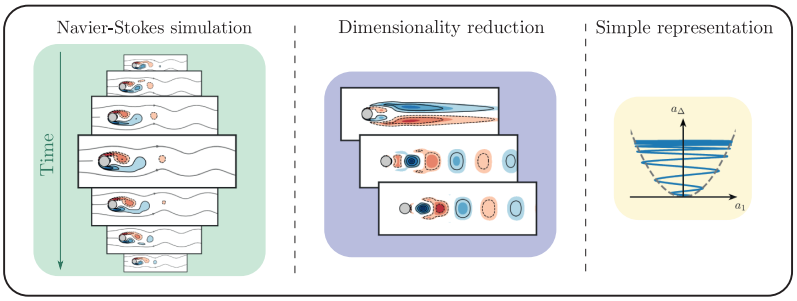


Figure 9.1: Illustration of reduced-order modeling. Starting from a direct numerical simulation of the Navier–Stokes equations (left), the dominant spatio-temporal coherent structures are extracted from a set of velocity snapshots (center). The temporal evolution of these structures then provides a simplified representation of the system's dynamics (right) amenable to modeling.

turbulence, almost three decades ago. Subsequent POD models have been developed for the transitional boundary layer [83], the mixing layer [111, 114], the cylinder wake [33, 42], and the Ahmed body wake [80], to name only a few.

POD-Galerkin modeling is challenging for changing domains [18], changing boundary conditions [45], and slow deformation of the modal basis [5]. Standard Galerkin projection can also be expected to suffer from stability issues [82, 90, 29], although including energy-preserving constraints may improve the long-time stability and performance of nonlinear models [7, 31]. POD-Galerkin models tend to be valid for a narrow range of operating conditions, near those of the data set used to generate the POD modes. Transients also pose a challenge to POD modeling. Refs. [77] and [106] demonstrate the ability of a low-dimensional model to reproduce nonlinear transients of the von Kármán vortex shedding past a two-dimensional cylinder, provided the projection basis includes a *shift mode* quantifying the distortion between the linearly unstable base flow and marginally stable mean flow. These techniques have been extended to include the effect of wall actuation [45, 81].

In addition to the physics-informed Galerkin projection, data-driven modeling approaches are prevalent in fluid dynamics [21, 85]. For example, dynamic mode decomposition (DMD) [50, 86, 55], the eigensystem realization algorithm (ERA) [51], Koopman analysis [72, 73, 109, 116], cluster-based reduced-order models [53], NARMAX models [15, 95, 120, 44], and network analysis [76] have all been used to identify dynamical systems models from fluids data, without relying on prior knowledge of the underlying Navier–Stokes equations. DMD models are readily obtained directly from data, and they provide interpretability in terms of flow structures, but the resulting models are linear, and the connection to nonlinear systems is tenuous unless DMD is enriched with nonlinear functions of the data [116, 55]. Neural networks have long been used for flow modeling and control [74, 122, 56, 54], and recently deep neural networks have been used for Reynolds-averaged turbulence modeling [59]. However, many machine learning methods may be prone to overfitting, have limited

interpretability, and make it difficult to incorporate known physical constraints. Parsimony has thus become an overarching goal when using machine learning to model nonlinear dynamics. In the seminal work of [16] and [91] governing dynamics and conservation laws are discovered using genetic programming along with a Pareto analysis to balance model accuracy and complexity, preventing overfitting.

Recently, [22] introduced the sparse identification of nonlinear dynamics (SINDy), which identifies parsimonious nonlinear models from data. SINDy follows the principle of Ockham's razor, resting on the assumption that there are only a few important terms that govern the dynamics of a system, so that the equations are sparse in the space of possible functions. Sparse regression is then used to efficiently determine the fewest terms in the dynamics required to accurately represent the data, preventing overfitting. Because SINDy is based on linear algebra (i. e., the nonlinear dynamics are represented as a linear combination of candidate nonlinear functions), the method is readily extended to incorporate known physical constraints [61]. In general, it is possible to obtain nonlinear models using genetic programming or SINDy on POD or DMD mode coefficients, which make these methods *gray* box, having a transformation from the model back to the high-dimensional, interpretable state space. However, models developed on POD/DMD mode coefficients may still suffer from fundamental challenges of traditional POD-Galerkin models, such as capturing changing boundary conditions, moving geometry, and varying operating condition.

9.1.2 Contribution of this work

In this work, we introduce a new gray-box modeling procedure that yields interpretable nonlinear models from measurement data. The method is applied to the well-investigated two-dimensional transient flow past a circular cylinder with slow change of the base flow and varying coherent structures [105]. In particular, we develop sparse interpretable nonlinear models only from the temporal amplitudes $a_1(t)$ and $a_2(t)$ of the leading vortex shedding POD modes, hereafter denoted as our features. Second, a sparse dynamical model is identified in this feature space. For the following step, full-state measurement data are assumed to be available. Combining the nonlinear correlations existing between the various POD modes with techniques from Grassmann manifold interpolation enables us to obtain highly accurate estimates of the flow field both in the vicinity of the linearly unstable base flow and the marginally stable flow. This mapping provides significantly more accurate flow reconstruction, as compared to a POD-Galerkin model of the same order. To summarize, the resulting gray-box modeling procedure has the following beneficial features: (i) it captures nonlinear physics, (ii) it is based on a simple, noninvasive computational algorithm, (iii) the resulting model is interpretable in terms of nonlinear interaction physics and generalized modes (optional with full-state data), and (iv) modeling

feature vectors is more robust to mode deformation, moving geometry, and varying operating condition.

The chapter is organized as follows: Section 9.2 provides an overview of the flow configuration considered in this work, namely, the incompressible, two-dimensional flow past a circular cylinder at Re = 100. Based on velocity snapshots obtained from direct numerical simulations, two different reduced-order modeling strategies are presented in Sections 9.3 and 9.4. First, Section 9.3 introduces the canonical POD-Galerkin reduced-order model and discusses its main limitations. Then, Section 9.4 presents a highly accurate low-order model identified using recent advances in machine learning. Finally, Sections 9.5 and 9.6 summarize our key findings, highlight some connections with previous works, and provide the reader with good practices and possible future directions to extend this work.

This contribution closely relates to three chapters of Volumes 1 and 2 of this handbook. Starting point is the POD-Galerkin method [12, Chapter 2]. A transient cylinder wake illustrates the benefits from manifold interpolation [124, Chapter 7]: A two-dimensional manifold is more accurate than a POD expansion with 50 modes. The resulting dynamical system on this manifold is significantly simplified by SINDy [12, Chapter 7].

9.2 Benchmark configuration and dynamics

The flow configuration considered is the canonical two-dimensional incompressible viscous flow past a circular cylinder at Re = 100, based on the free-stream velocity U_∞, the cylinder diameter D, and the kinematic viscosity ν. This Reynolds number is well above the critical Reynolds number ($\mathrm{Re}_c = 48$) for the onset of the two-dimensional vortex shedding [118, 104, 94] and below the critical Reynolds number ($\mathrm{Re}_c = 188$) for the onset of three-dimensional instabilities [119, 8, 117]. Its dynamics are governed by the incompressible Navier-Stokes equations

$$\begin{aligned} &\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}, \\ &\nabla \cdot \boldsymbol{u} = 0, \end{aligned} \tag{9.1}$$

where $\boldsymbol{u} = (u, v)^T$ and p are the velocity and pressure fields, respectively. The center of the cylinder has been chosen as the origin of the reference frame $\boldsymbol{x} = (x, y)$, where x denotes the streamwise coordinate and y denotes the spanwise coordinate. This study considers the same computational domain as in [77, 61, 63], extending from $x = -5$ to $x = 15$ in the streamwise direction and from $y = -5$ to $y = 5$ in the spanwise direction. A uniform velocity profile is prescribed at the inflow, a classical stress-free boundary condition is used at the outflow, and free-slip boundary conditions are used on

the lateral boundaries of the computational domain. The open-source spectral element solver Nek5000 [41] is used to solve the equations with a third-order accurate temporal integration. For the sake of reproducibility, all of the files required to rerun the simulations presented in this work are freely available at the following address: https://www.github.com/loiseaujc along with an illustrative Jupyter Notebook.

9.2.1 Direct numerical simulation

Figure 9.2 depicts the evolution of the lift coefficient C_L as a function of time. This direct numerical simulation (DNS) has been initialized with

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_b + \epsilon \Re(\hat{\boldsymbol{u}})(\boldsymbol{x}),$$

where $\boldsymbol{u}_b$ is the linearly unstable base flow and $\Re(\hat{\boldsymbol{u}})(\boldsymbol{x})$ is the real part of the linearly unstable eigenmode normalized such that its amplitude is equal to unity (see Section 9.2.2 for more details). The parameter ϵ, fixing the initial amplitude of the perturbation, was set such that the initial energy of the perturbation is of the order 10^{-6}.

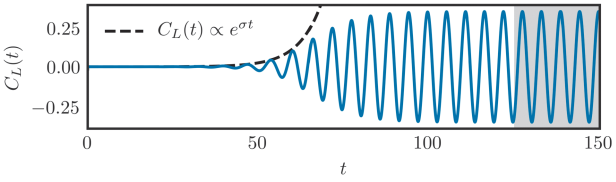


Figure 9.2: Time series of the instantaneous lift coefficient $C_L(t)$, from the linearly unstable base flow to the marginally stable mean flow, obtained by direct numerical simulation. The black dashed line depicts the exponential growth predicted by linear stability analysis while the gray shaded region highlights the window over which flow snapshots have been collected for the POD analysis presented in Section 9.3.

Three different phases are clearly visible in the time evolution of $C_L(t)$, namely, a period of exponential growth for $0 \leq t \leq 60$, the onset of nonlinear saturation for $60 \leq t \leq 100$, and finally the constant amplitude quasi-harmonic oscillatory regime for $t \geq 100$ characteristic of the von Kármán vortex street. The nonlinear saturation mechanism is briefly described hereafter. The nonlinear interaction of the instability mode with itself produces Reynolds stresses that distort the underlying base flow which, in turn, modifies the shape of the instability mode. This distortion also induces a frequency shift, the flow oscillating at a frequency almost 30 % larger in its final saturated state compared to that predicted by linear stability analysis of the base flow. This process continues until an equilibrium is achieved, balancing the influence of the perturbation's Reynolds stresses onto the instantaneous mean flow and the feedback this mean flow has onto the instantaneous growth rate of the perturbation. When

this equilibrium is reached, the flow is in a marginally stable state [9] and the amplitude of the perturbation no longer grows. For a complete description of this stabilizing nonlinear feedback mechanism, interested readers are referred to the self-consistent model presented in [68] or the weakly nonlinear analyses conducted by [96] and [27].

9.2.2 Stability of the steady solution

Given a fixed point $\boldsymbol{u}_b$ of the Navier–Stokes equations, the dynamics of an infinitesimal perturbation $\boldsymbol{u}'$ evolving in its vicinity are governed by

$$\begin{aligned} &\frac{\partial \boldsymbol{u}'}{\partial t} + \nabla \cdot (\boldsymbol{u}_b \otimes \boldsymbol{u}' + \boldsymbol{u}' \otimes \boldsymbol{u}_b) = -\nabla p' + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}', \\ &\nabla \cdot \boldsymbol{u}' = 0. \end{aligned} \tag{9.2}$$

Introducing the normal mode ansatz $\boldsymbol{u}'(\boldsymbol{x}, t) = \hat{\boldsymbol{u}}(\boldsymbol{x}) e^{\lambda t}$, this set of equations can be recast into the following generalized eigenvalue problem:

$$\lambda \begin{bmatrix} \mathcal{I} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} -\nabla \cdot (\boldsymbol{u}_b \otimes \cdot + \cdot \otimes \boldsymbol{u}_b) + \frac{1}{\mathrm{Re}} & -\nabla \\ \nabla \cdot & 0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{p} \end{bmatrix}. \tag{9.3}$$

The linear stability of the base flow $\boldsymbol{u}_b$ is then governed by the real part of the eigenvalue λ. In the rest of this work, the linearly unstable flow $\boldsymbol{u}_b$ has been obtained using the selective damping approach [1] while the eigenpairs of the linearized Navier–Stokes operator have been computed using a time stepper Arnoldi algorithm [37, 6, 60, 62]. Interested readers are referred to [30, 108, 97] for exhaustive reviews about hydrodynamic instabilities.

The vorticity field of the linearly unstable base flow $\boldsymbol{u}_b$ at $\mathrm{Re} = 100$ is depicted in Figure 9.3a. To the best of our knowledge, this is the only fixed point of the Navier–Stokes equations known for this flow configuration. Its linear stability has been extensively investigated [43, 96, 68, 27], and it is now well known that the bifurcation occurring at $\mathrm{Re}_c \simeq 48$ is a supercritical Andronov–Poincaré–Hopf bifurcation eventually giving rise to the canonical Bénard–von Kármán vortex street. The vorticity field of the corresponding unstable eigenmode is shown in Figure 9.3b. This complex–conjugate pair of eigenmodes is the only unstable pair before the onset of three-dimensionality.

From a dynamical system point of view, one thus concludes that, although our discretized system is of the order 10^6 dimensions, the unstable linear subspace of the fixed point is only two-dimensional, i. e., only two degrees of freedom are required to describe the evolution of the system within this linear subspace. Let us furthermore consider the following *stable and unstable manifold theorem* [46].

Theorem 1. *Let E be an open subset of $\mathbb{R}^n$ containing the origin, let $f \in C^1(E)$, and let ϕ_t be the flow of the nonlinear system*

$$\frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} = f(\boldsymbol{a}).$$

Suppose that $f(0) = 0$ *and that the Jacobian matrix* $\boldsymbol{L} = Df(0)$ *has* k *eigenvalues with negative real part and* $n - k$ *eigenvalues with positive real part. Then, there exists a* k*-dimensional manifold* W^s *tangent to the stable subspace* E^s *of the linear system*

$$\frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} = \boldsymbol{L}\boldsymbol{a}$$

at $\boldsymbol{a}_0 = 0$. *Similarly, there exists an* $(n - k)$*-dimensional unstable manifold* W^u *tangent to the unstable subspace* E^u.

This theorem is of crucial importance for the understanding of the reduced-order model to be discussed in Section 9.4. Indeed, although we will eventually consider the nonlinear evolution of our 10^6-dimensional system, we will see that this evolution can be described by a very simple dynamical system evolving onto a two-dimensional parabolic manifold originating from the aforementioned unstable subspace E^u.

9.2.3 Stability of the mean flow

For the flow configuration considered herein, the linearly unstable base flow $\boldsymbol{u}_b(\boldsymbol{x})$ and the mean flow $\bar{\boldsymbol{u}}(\boldsymbol{x})$ computed from DNS differ quite significantly from one another, notably in the size of the recirculation bubble (see Figures 9.3a and 9.4a). Consequently, predictions of the spatio-temporal characteristics of the fluctuation obtained by linear stability analysis of the base flow might be misleading.

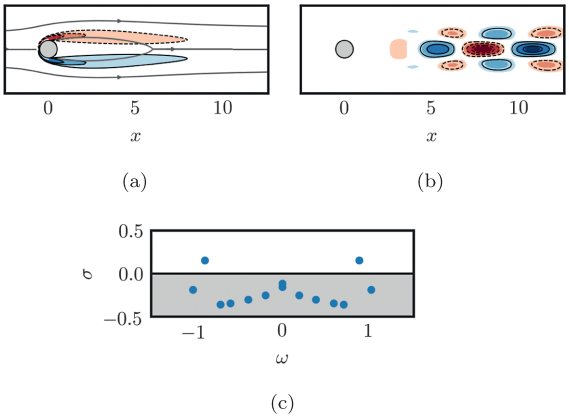


Figure 9.3: (a) Vorticity field of the linearly unstable base flow for the two-dimensional cylinder flow at Re = 100. (b) Real part of the leading unstable mode's vorticity field. In both figures, blue shaded contours (solid lines) highlight regions of positive vorticity, while red shaded ones (dashed lines) highlight those of negative vorticity. In (a), a few streamlines are plotted (light gray) to highlight the extent of the recirculation bubble. (c) Eigenspectrum of the corresponding linearized Navier–Stokes operator.

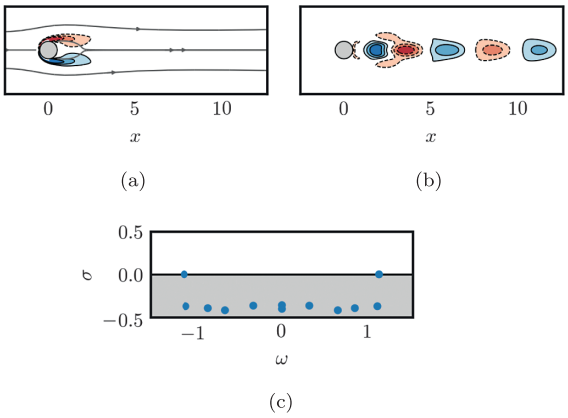


Figure 9.4: (a) Vorticity field of the marginally stable mean flow for the two-dimensional cylinder flow at Re = 100. (b) Real part of the marginal mode's vorticity field. In both figures, blue shaded contours (solid lines) highlight regions of positive vorticity, while red shaded ones (dashed lines) highlight those of negative vorticity. In (a), a few streamlines are plotted (light gray) to highlight the extent of the recirculation bubble. (c) Eigenspectrum of the corresponding linearized Navier–Stokes operator.

Even though the mean flow $\bar{\boldsymbol{u}}(\boldsymbol{x})$ is not a solution of the stationary Navier–Stokes equations, it has now become quite standard nonetheless to linearize the Navier–Stokes equations in its vicinity as to study its linear stability [9, 110, 11]. The eigenspectrum of the corresponding linearized Navier–Stokes operator is depicted in Figure 9.4c. As shown in [9], the leading eigenvalues have a zero real part, indicating that this mean flow is marginally stable. Moreover, while the frequency predicted by linear stability analysis of the base flow differs by almost 30 % from the one recorded in direct numerical simulation, the one predicted by stability analysis of the mean flow almost is a perfect match. This mismatch results from the strong distortion induced by the instability mode as it saturates nonlinearly. Similarly, the eigenmode shown in Figure 9.4b provides a much better representation of the spatial characteristics of the fluctuations observed in DNS. For extensive details and theoretical justifications about mean flow stability analysis, interested readers are referred to [65, 9, 96, 68, 69, 110, 71, 11, 70] and references therein.

9.3 POD-Galerkin projection of the Navier–Stokes equations

POD [100, 49] provides a low-rank modal decomposition of fluid flow field data, extracting the most energetic modes. It is then possible to project the Navier–Stokes onto the span of these POD modes, resulting in an approximate low-dimensional model

governing the evolution of the mode coefficients. POD-Galerkin models are widely used as they are interpretable gray-box models and it is straightforward to reconstruct the high-dimensional state vector of the original system from the low-dimensional model via the POD modes. The first pioneering example of [4] featured wall turbulence, over three decades ago. Subsequent POD models have been developed for the transitional boundary layer [83], the mixing layer [111, 114], the cylinder wake [33, 77, 42], and the Ahmed body wake [80], to name only a few. In the present section, dimensionality reduction via POD analysis is first presented in Section 9.3.1. Then, Sections 9.3.2 to 9.3.5 discuss the derivation of the reduced-order model from the Navier–Stokes equations and its properties, as well as its accuracy and limitations.

9.3.1 Dimensionality reduction – POD analysis

A large number of systems, including but not limited to fluid flows, are governed by high-dimensional nonlinear dynamics. Nonetheless, because most of these nonlinear dynamical systems are dissipative by nature, their dynamics are likely to evolve onto a lower-dimensional attractor characterized by a few dominant coherent structures containing a significant portion of the system's energy [49]. Given a high-dimensional data set, the aim of *dimensionality reduction* is thus to extract a low-dimensional embedding capturing most of the variability of the original data. One of the most widely used techniques for dimensionality reduction is *POD*. It is also known as principal component analysis (PCA) in statistics and machine learning, as Kosambi–Karhunen–Loève transform in signal processing, or as empirical orthogonal functions in meteorological science, and it is closely related to singular value decomposition (see Figure 9.5). For the sake of conciseness, the mathematical details of POD will not be discussed herein. For more details, interested readers are referred to [100] and [14]. Note additionally that POD is discussed at length in this book series; see for instance Chapters 2 and 12 of Volume 1.

The gray shaded region in Figure 9.2 highlights the window over which snapshots of the base flow-subtracted fluctuation have been collected for the present POD analysis at a sampling rate approximately 25 times higher than the circular frequency of the natural vortex shedding. Figure 9.6a depicts the fraction of the fluctuation's kinetic energy captured by each of the first 10 POD modes along with its cumulative sum. Note that, because we have considered base flow-subtracted fluctuations rather than mean flow-subtracted ones, the leading POD mode corresponds to the shift mode [77]. This mode captures the distortion between the base flow and the mean flow (Figure 9.6b) and accounts for 46 % of the whole kinetic energy in our snapshots data set. Considering the second and third POD modes, related to the vortex shedding (Figure 9.6c), 97.7 % of the total kinetic energy is captured. Finally, less than 1 % of the kinetic energy is discarded if one considers the first five POD modes, and less than 0.1 % if the first

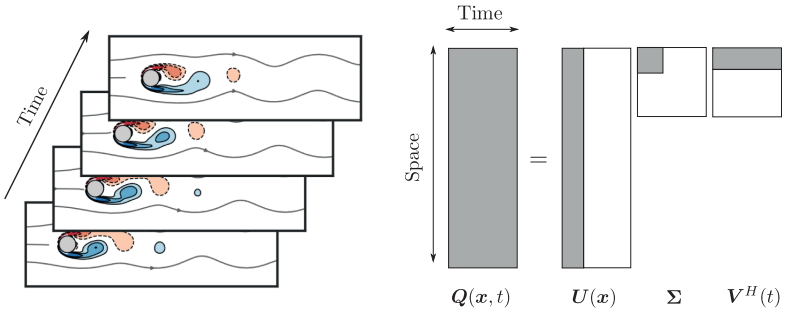


Figure 9.5: Schematic representation of the low-rank approximation of the data matrix $\boldsymbol{Q}$ by means of singular value decomposition. Each column of $\boldsymbol{Q}$ contains one snapshot obtained from direct numerical simulation. The matrix $\boldsymbol{U}$ contains the space-dependent POD modes $\boldsymbol{u}_i(\boldsymbol{x})$ while $\boldsymbol{V}$ contains the associated temporal evolutions, with superscript H denoting the Hermitian (i. e., complex conjugate transpose) operation. Finally, the diagonal matrix $\boldsymbol{\Sigma}$ contains the singular values whose square characterizes the amount of variance explained by the associated singular pairs.

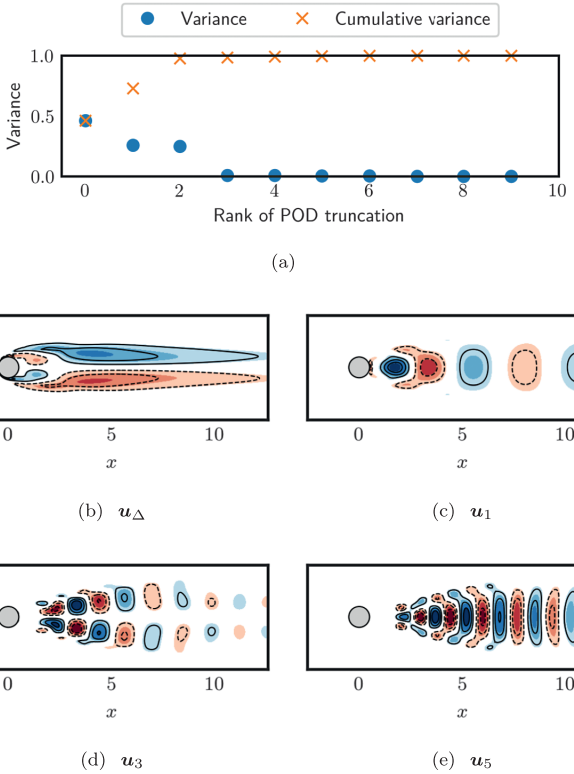


Figure 9.6: (a) Fraction of the total variance (•) explained by each POD mode and the corresponding cumulative variance (×). This POD analysis has been performed using base flow-subtracted snapshots collected during the gray shaded window in Figure 9.2. The zeroth POD eigenvalue in this plot is associated to the shift mode $\boldsymbol{u}_\Delta$. Figures (b) to (e) depict the vorticity distribution of the shift mode and the first, third, and fifth POD modes, respectively. Only a subset of the whole computational domain is depicted.

seven ones are considered. For the sake of completeness, the vorticity field of selected POD modes are shown in Figure 9.6b–e.

Figure 9.7 depicts the phase plots of these various POD modes. For Figure 9.7a–c, only the evolution of the flow once it has reached the limit cycle is shown. It can be seen that, within the (a_1, a_2)-plane, the evolution of the flow traces a perfect circle underlining the periodic nature of the saturated vortex shedding for the Reynolds number considered. Additionally, the phase plots shown in Figure 9.7 highlight that the third and fourth POD modes correspond to the second harmonics of the vortex shedding, while the fifth and sixth modes capture its third harmonics. Finally, Figure 9.7d shows the whole evolution of the system, from the base flow to the mean flow, projected onto the (a_1, a_Δ)-plane. As expected, one recovers the well-known low-dimensional parabolic manifold [77] characteristic of a large number of wake flows. It is these dynamics that we wish to capture in Section 9.3.2 using a POD-Galerkin reduced-order model.

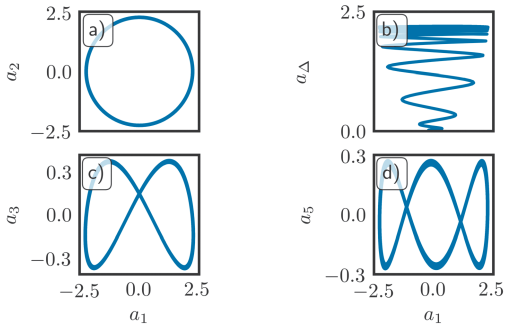


Figure 9.7: Phase plots of various POD modes. For (a), (c), and (d), only the evolution once the flow has reached the limit cycle is depicted. In (b), the whole evolution is shown, from the linearly unstable base flow to the marginally stable mean flow.

9.3.2 Reduced-order modeling strategy – Galerkin projection

The POD analysis performed in the previous section has revealed that close to 97.5 % of the base flow subtracted fluctuation's kinetic energy is captured by considering only the shift mode and the first pair of POD modes. Starting from this observation, it thus appears reasonable to approximate the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ using the following Galerkin expansion:

$$\boldsymbol{u}(\boldsymbol{x}, t) \simeq \boldsymbol{u}_b(\boldsymbol{x}) + \boldsymbol{u}_\Delta(\boldsymbol{x}) a_\Delta(t) + \boldsymbol{u}_1(\boldsymbol{x}) a_1(t) + \boldsymbol{u}_2(\boldsymbol{x}) a_2(t), \tag{9.4}$$

where $\boldsymbol{u}_b(\boldsymbol{x})$ is the linearly unstable fixed point of the Navier–Stokes equations, while $\boldsymbol{u}_\Delta(\boldsymbol{x})$, $\boldsymbol{u}_1(\boldsymbol{x})$ and $\boldsymbol{u}_2(\boldsymbol{x})$ are the velocity fields associated with the shift mode and the first

two POD modes, respectively. Starting from the Navier–Stokes equations, our goal is thus to derive a low-dimensional system of nonlinearly coupled ordinary differential equations governing the evolution of the POD modes' amplitudes $a_i(t)$. Introducing our Galerkin expansion ansatz into the Navier–Stokes equations and projecting the latter onto the span of our POD basis (this process is known as *Galerkin projection*), we obtain evolution equations for each amplitude $a_i(t)$ of the form

$$\frac{\mathrm{d}a_i}{\mathrm{d}t} = \sum_j L_{ij} a_j + \sum_j \sum_k Q_{ijk} a_j a_k, \tag{9.5}$$

with $i, j, k = \Delta, 1, 2$. By convention, the coefficient a_0 associated to the base flow $\boldsymbol{u}_b(\boldsymbol{x})$ is set to $a_0 = 1$. In the above equation, the linear term is given by

$$L_{ij} = \left\langle \boldsymbol{u}_i \middle| -\nabla \cdot (\boldsymbol{u}_b \otimes \boldsymbol{u}_j + \boldsymbol{u}_j \otimes \boldsymbol{u}_b) + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}_j \right\rangle,$$

while the quadratic one is

$$Q_{ijk} = -\langle \boldsymbol{u}_i | \nabla \cdot (\boldsymbol{u}_j \otimes \boldsymbol{u}_k) \rangle,$$

where $\langle \boldsymbol{a} | \boldsymbol{b} \rangle$ denotes the inner product

$$\langle \boldsymbol{a} | \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a} \cdot \boldsymbol{b} \, \mathrm{d}\Omega.$$

Note that, as in [77], we did not explicitly account for the pressure term. For the present case, this omission however hardly changes the prediction of the reduced-order model. For a detailed discussion about the importance (or insignificance) of the pressure term in POD-Galerkin projection reduced-order models, interested readers are referred to [79].

9.3.3 Does the model capture the key physics?

Before discussing whether the reduced-order model derived by POD-Galerkin projection is accurate or not, let us first investigate whether it captures the key physics of the problem. In the present case, this would imply that:

1. The reduced-order model has a single fixed point located at $\boldsymbol{a} = 0$.
2. The unstable subspace E^u of the reduced-order model linearized in the vicinity of $\boldsymbol{a} = 0$ is two-dimensional and associated with a complex–conjugate eigenpair.
3. As $t \to \infty$, the system eventually evolves toward a structurally stable limit cycle.

It must be emphasized that if the reduced-order model fails to comply with any of these requirements, then it fails at capturing the key physics of the problem.

Given the low-dimensionality of the present model, condition 1 can easily be (and has been) checked by performing an extensive Newton search. As expected, the only fixed point admitted by our reduced-order model is $\boldsymbol{a} = \boldsymbol{0}$. The linearization of our model in the vicinity of this fixed point is given by

$$\frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} = \boldsymbol{L}\boldsymbol{a},$$

with $\boldsymbol{a} = [a_1\ a_2\ a_\Delta]^T$ and

$$\boldsymbol{L} = \begin{bmatrix} 0.042 & -0.986 & 0 \\ 0.959 & 0.046 & 0 \\ 0 & 0 & -0.047 \end{bmatrix}.$$

Spectral decomposition of this matrix reveals that its eigenvalues are

$$\boldsymbol{\Lambda} = \{\lambda_1 = 0.044 + i0.972,\ \lambda_2 = 0.044 - i0.972,\ \lambda_\Delta = -0.047\}, \tag{9.6}$$

while the corresponding set of eigenvectors is

$$E_{\boldsymbol{\Lambda}} = \left\{ \hat{\boldsymbol{a}}_1 = \begin{bmatrix} 1 \\ -i \\ 0 \end{bmatrix}, \hat{\boldsymbol{a}}_2 = \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix}, \hat{\boldsymbol{a}}_\Delta = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{9.7}$$

Looking at these eigenpairs, it is clear that, as for the original Navier–Stokes equations, the fixed point $\boldsymbol{a} = \boldsymbol{0}$ of our POD-Galerkin reduced-order model is linearly unstable. Moreover, its unstable subspace E^u is also two-dimensional and associated with complex–conjugate eigenvalues and eigenvectors corresponding to oscillatory dynamics in the (a_1, a_2)-plane while it is stable along the direction corresponding to the shift mode. Condition 2 is thus also fulfilled.

The last condition that needs to be checked is whether or not the system naturally evolves toward a stable limit cycle as $t \to \infty$. To do so, we integrate in time our reduced-order model using a fourth-order accurate Runge–Kutta scheme. Figure 9.8 depicts the predicted asymptotic evolution. As can be observed, this reduced-order model does evolve toward a stable limit cycle, although its amplitude is slightly larger than the amplitude of the limit cycle obtained from direct numerical simulation of the Navier–Stokes equations. Our reduced-order model thus fulfills all three necessary conditions we stated at the beginning of this section and, as such, captures qualitatively the key physics of the two-dimensional cylinder flow. Consequently, the only question that remains to be answered is the following: How accurate is this reduced-order model? The answer to this question is the subject of Section 9.3.4.

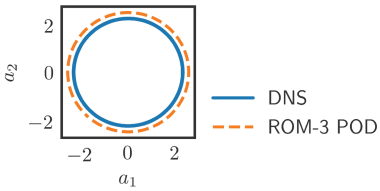


Figure 9.8: Comparison of the limit cycles observed in DNS (–) and predicted by the three-POD mode reduced-order model (orange ––).

9.3.4 How accurate is it?

We have shown in the previous section that a reduced-order model derived from the Navier–Stokes equations by means of a POD-Galerkin projection procedure qualitatively captures the key physics of the problem considered, namely:

Property 1: It has a single fixed point at the origin.

Property 2: This fixed point is linearly unstable and the associated unstable subspace is two-dimensional.

Property 3: As $t \to \infty$, the reduced-order model predicts that the system naturally evolves toward a periodic limit cycle.

Let us now try to further characterize the accuracy of said reduced-order model. In particular, we will focus our attention on two critical aspects:

1. Does it appropriately capture the transient dynamics of the flow as it evolves from the linearly stable base flow to the marginally stable mean flow?
2. How good are its flow reconstruction capabilities?

As to answer to these questions, the reduced-order model is fed with a random initial condition having the same initial energy as that used in the direct numerical simulation described in Section 9.2, i. e.,

$$\boldsymbol{a}(0) = \alpha\hat{\boldsymbol{a}}_1 + \beta\hat{\boldsymbol{a}}_2,$$

such that $\|\boldsymbol{a}(0)\|_2^2 = 10^{-6}$. Figure 9.9 depicts the evolution of the fluctuation's kinetic energy as a function of time obtained from direct numerical simulation and predicted by our POD-Galerkin reduced-order model. Although our low-order model qualitatively captures the transient dynamics of the flow, i. e., a period of exponential growth followed by nonlinear saturation, it is clear that it largely overestimates the transients duration. Moreover, as nonlinear saturation occurs, the reduced-order model predicts an energy overshoot before it saturates at a level higher than that observed in DNS. These two observations put in the limelight two critical issues of a large number of reduced-order models derived from the Navier–Stokes equations by a POD-Galerkin procedure.

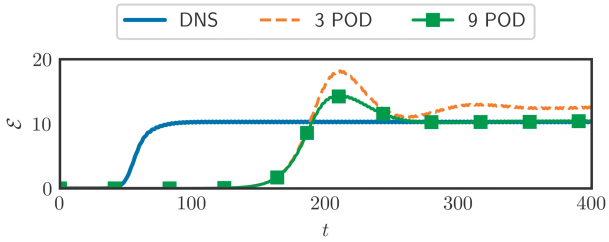


Figure 9.9: Evolution as a function of time of the kinetic energy $\mathcal{E}(t)$ of the base flow-subtracted fluctuation for the DNS and two Galerkin projection reduced-order models using either the first three or the first nine POD modes.

Let us first consider the problem of the overestimation of the transients duration. This problem finds its roots in the major difference that exists between the POD modes associated with the first harmonics of the vortex shedding and the eigenmodes of the linearized Navier–Stokes operator. Looking at Figure 9.3b and c, it can be seen that the POD modes are located further upstream compared to the instability modes. Consequently, while the projection of the linearized Navier–Stokes operator onto the span of the POD modes reasonably approximates the dynamics of the system in the vicinity of the mean flow, it provides a very crude approximation of the dynamics of the system when close to its fixed point, notably in terms of the instability growth rate. This is a structural problem of POD-Galerkin reduced-order models. Indeed, from a physical point of view, the instability modes continuously deform into the POD modes as the amplitude of the fluctuation grows. However, fixing the projection basis a priori using solely the POD modes prevents the reduced-order model from being able to capture this mode deformation and the continuous change of dynamics associated with it. As to alleviate this problem, [77] explicitly included the instability modes into the projection basis. Although this trick partially solves the problem, it unnecessarily increases the dimensionality of the reduced-order model.

The second problem of the present low-dimensional model is the energy overshoot and the subsequent saturation to a higher level than the one observed in DNS. This problem arises from the projection of the Navier–Stokes equations onto a finite number of basis vectors and thus from the chosen truncation of the POD basis. In the present case, our projection basis consists only of the shift mode (quantifying the distortion between the base flow and the mean flow) and the POD modes associated with the first harmonics of the vortex shedding. Because of this choice, the energy cascade from the large scales to the small scales is truncated early on. As a consequence, the energy extracted by the leading POD modes from the underlying unstable base flow cannot be transferred correctly to smaller-scale structures, hence growing beyond their expected amplitudes and causing the energy overshoot observed in Figure 9.9. This excess energy is eventually absorbed by the mean flow distortion until an equilibrium is reached, even though the final kinetic energy of the reduced-order

model nonetheless saturates at a higher level that the one observed in DNS. A naive approach to fix this issue would be to include more POD modes in the projection basis. This is illustrated in Figure 9.9, where the evolution of the kinetic energy predicted by a reduced-order model derived using a projection basis that includes the POD modes associated with the second, third, and fourth harmonics of the vortex shedding is also shown. Although increasing the rank of the POD basis from 3 to 9 mitigates the problem, the energy overshoot still exists. Moreover, including these higher-order modes in the projection basis also modifies the properties of the linearized dynamics in the vicinity of the fixed point. In the present case, including the POD modes associated with the second harmonics of the vortex shedding actually increases the dimensionality of the unstable subspace E^u from 2 to 4. In the vicinity of the fixed point, the properties of the linearized reduced-order model thus become inconsistent with those of the linearized Navier–Stokes operator.

9.3.5 Limitations of this approach

Although the POD-Galerkin approach to reduced-order modeling has had considerable success over the years, it nonetheless suffers from major limitations, even for a flow configuration as simple as the two-dimensional cylinder flow. For the case considered herein, four major limitations can be listed:

1. In order to accurately capture the dynamics of the system once on the limit cycle, the projection basis had to include a relatively large number of modes (i. e., eight) despite the simplicity of the dynamics, including very low energy modes.
2. The low-dimensional system tends to exhibit an energy overshoot as nonlinear saturation occurs because of the truncation of the energy cascade. This truncation of the energy cascade results from the projection of the nonlinear partial differential equations onto a finite set of basis vectors.
3. Because of the difference between the linear instability and the POD modes obtained from the limit cycle, the reduced-order model largely overestimates the transients duration unless the instability modes are explicitly included into the projection basis.
4. Finally, it can hardly account for the continuous mode deformation taking place as the flow evolves from the vicinity of the linearly unstable base flow to that of the marginally stable mean flow. A similar problem arises if one varies the Reynolds number slowly in time.

Since the generalized mean field model of Noack et al. [77], various attempts have been made to limit these shortcomings. For instance, [99] and [113] used eddy viscosity models to account for the added diffusion induced by the truncated modes, while [75] and [103] used linear interpolation to partially capture the continuous mode deformation. Recently, [34] have used *sparse coding* to obtain a nonorthonormal projection basis

for the turbulent lid-driven cavity flow that nonetheless included some of the small-scale structures needed for the energy cascade, while [40] combined POD-Galerkin projection with constrained convex optimization techniques to ensure that the statistical properties of the POD amplitudes predicted by the reduced-order model were consistent with those obtained from direct numerical simulations. These works however still had to include dozens of POD modes for numerical stability although the dynamics of the system are lower-dimensional. Despite all these attempts to increase the range of validity of the POD-Galerkin projection approach, one must not forget that it still suffers from one critical limitation that cannot be overcome within this particular framework: The governing equations of the high-dimensional system (in our case the Navier–Stokes equations) need to be known before one even tries to perform model reduction.

9.4 Manifold model

The approach described in the previous section can be understood as a semi-empirical or partially data-driven approach. Indeed, while on the one hand the projection basis is obtained via POD of a snapshots data matrix, the Galerkin projection procedure relies on a priori knowledge of the high-dimensional system's governing equations. Let us now consider a fully data-driven model of the flow that leverages the existence of a low-dimensional nonlinear manifold. Starting from the POD analysis presented in the previous section, Section 9.4.1 illustrates how one can further reduce the dimensionality of the problem by considering the nonlinear correlations existing between the various POD mode amplitudes. As a second step, a low-dimensional system is obtained using recent system identification techniques in Section 9.4.2. Finally, given that the system under consideration evolves on a low-dimensional manifold, Section 9.4.5 highlights how one can use Grassmannian manifolds to solve the continuous mode deformation problem when reconstructing the high-dimensional state vector of the full-order model, while Section 9.4.6 discusses some of the limitations of the approach proposed herein.

9.4.1 Looking for nonlinear correlations

PCA (equivalent to POD in mechanical engineering) is one of the most popular dimensionality reduction techniques. One of the key reasons for this widespread usage is that PCA finds its root in statistics. Moreover, when formulated as a singular value decomposition, PCA can be understood as an optimal low-rank matrix approximation and can thus leverage highly performing and scalable algorithms to handle extremely large data sets. Considering only the first few principal components (i. e.,

the leading left singular vectors of the data matrix), one can define an optimal linear subspace onto which the data can be orthogonally projected while minimizing (and quantifying) the amount of information lost in the process. From a statistical point of view, this orthogonal projection provides linearly uncorrelated features. Despite its optimality properties, PCA unfortunately cannot unravel nonlinear correlations in the data and postanalyses are thus required. Accounting for such nonlinear correlations may however be beneficial to further reduce the dimensionality of the problem.

Over the years, various alternatives have been proposed to overcome this major limitation in order to be able to capture nonlinear manifolds. One can cite for instance kernel PCA (kPCA) [92], Isomap [107], locally linear embedding (LLE) and its variants [84, 121, 35], spectral embedding [10], multidimensional scaling (MDS) [17], or all the variants of autoencoders recently reviewed in [13]. All these techniques are part of a domain now known as *manifold learning* or *representation learning*. However, for the particular problem considered herein, the dynamics are sufficiently simple so that we can assess the existence of nonlinear correlations directly from time series of POD modes' amplitudes. From a practical point of view, the existence of a clear pattern in a phase plot (a_i-a_j) implies the existence of such nonlinear correlations (see Figure 9.7 for examples).

The POD analysis performed in Section 9.3.1 has revealed that less than 0.1 % of the total kinetic energy in our training data set is discarded if we only consider the shift mode and the first six POD modes. Given the Fourier-like nature of the POD coefficients once the flow evolves on the limit cycle, these can be approximated by

$$\begin{aligned} a_\Delta(t) &\simeq \hat{A}_\Delta, \\ a_1(t) \pm i a_2(t) &\simeq \hat{A}_1 e^{\pm i\omega t}, \\ a_3(t) \pm i a_4(t) &\simeq \hat{A}_2 e^{\pm i2\omega t}, \\ a_5(t) \pm i a_6(t) &\simeq \hat{A}_3 e^{\pm i3\omega t}, \end{aligned}$$

where ω is the fundamental frequency of the vortex shedding, $\hat{A}_\Delta$ is the amplitude of the shift mode in the saturated stage, and $\hat{A}_1$, $\hat{A}_2$, and $\hat{A}_3$ are the amplitudes of the first, second, and third pairs of POD modes, respectively. Guided by physical intuition, Figure 9.10 summarizes some of the possible triadic interactions arising from the nonlinear convective term $\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})$ of the Navier–Stokes equations. Looking at these triadic interactions, it thus appears that the dynamics of the shift mode and of the second pair of POD modes both result from quadratic interactions of the first pair of POD modes with itself. Similarly, the dynamics of the third pair of POD modes result from the interaction of the first pair with the second pair of modes. Alternatively, this last quadratic interaction can also be understood as a cubic interaction of the first pair with itself. These intuitions are further confirmed by looking at the correlation matrix depicted in Figure 9.11.

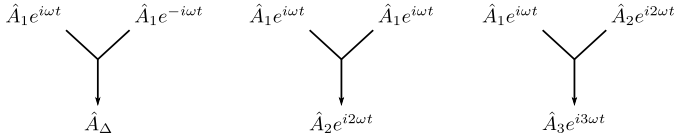


Figure 9.10: Some of the possible triadic interactions arising from the nonlinear convective term $\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})$ of the Navier–Stokes equations. These triadic interactions will guide us to determine the form of nonlinear correlations existing between the amplitudes of the various POD modes considered.

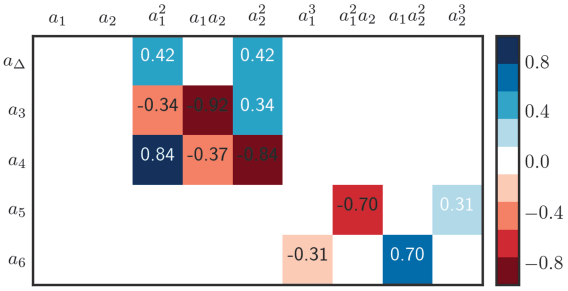


Figure 9.11: Pearson's ρ correlation coefficient between various monomials of a_1 and a_2 and the amplitude a_Δ of the shift mode or the amplitudes a_3 to a_6 of the higher-order POD modes. Blue denotes strong positive linear correlation, red denotes strong negative correlation, and white implies no linear correlation between the two variables considered.

The exact form of these nonlinear correlations can be unraveled by polynomial regression. Doing so, we obtain the following relationships:

$$\begin{aligned} a_\Delta &= 0.41(a_1^2 + a_2^2), \\ a_3 &= -0.028(a_1^2 - a_2^2) - 0.13a_1a_2, \\ a_4 &= 0.065(a_1^2 - a_2^2) - 0.056a_1a_2, \\ a_5 &= -0.065a_1^2a_2 + 0.022a_2^3, \\ a_6 &= -0.021a_1^3 + 0.066a_2^2a_1. \end{aligned} \tag{9.8}$$

Figure 9.12 provides a comparison of the evolution of the various POD modes' amplitudes obtained from DNS and the ones predicted by the nonlinear correlations identified. As can be observed, these quadratic and cubic correlations accurately capture the evolution of the higher-order POD modes as well as the existence of the paraboloid manifold. Hence, it is clear that, although POD analysis reveals that seven POD modes need to be considered to accurately reconstruct the flow, only two of these modes are actual degrees of freedom of the system while the rest of them are entirely slaved to these two. This observation is consistent with the fact that, as shown in Section 9.2, the unstable subspace of the Navier–Stokes operator linearized in the vicinity of the

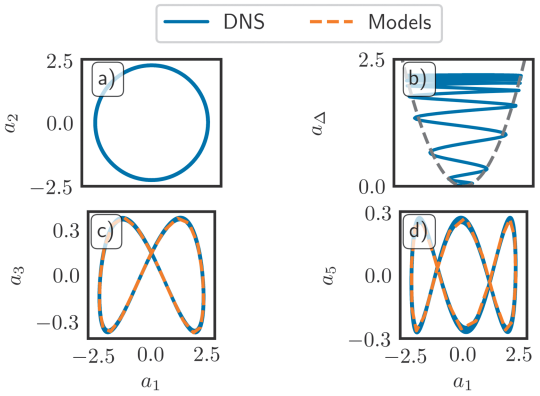


Figure 9.12: Same as Figure 9.7. The evolution of the coefficients a_3 and a_5 predicted by the nonlinear correlation models is also reported. In (d), only the parabola $a_\Delta = 0.41a_1^2$ (i. e., a slice of the paraboloid manifold in the $a_2 = 0$ plane) is shown.

unstable fixed point is only two-dimensional. The coming section is then devoted to the identification of the dynamical system governing the dynamics of a_1 and a_2.

9.4.2 Low-dimensional system identification – SINDy

Advanced regression methods from statistics, such as genetic programming or sparse regression, are driving new algorithms that identify parsimonious nonlinear dynamics from measurements of complex systems. Bongard and Lipson [16] and Schmidt and Lipson [91] introduced nonlinear system identification based on genetic programming, which has been used in numerous practical applications in aerospace engineering, the petroleum industry, and finance. More recently, Brunton et al. [22] have proposed a system identification approach based on sparse regression known as *sparse identification of nonlinear dynamics* (SINDy). Following the principle of Ockham's razor, SINDy rests on the assumption that there are only a few important terms that govern the dynamics of a given system so that the equations are sparse in the space of possible functions. Sparse regression is then used to determine the fewest terms in a dynamical system required to accurately represent the data. The resulting models are parsimonious, balancing model complexity with descriptive power while avoiding overfitting and remaining interpretable. For more details about SINDy, interested readers are referred to Chapter 12 of Volume 1 of the present book series as well as to the increasing body of literature on the subject [22, 66, 23, 87, 101, 32, 89, 88, 67, 61, 63, 52, 24, 48].

The nonlinear correlation analysis conducted in the previous section has revealed that the only true degrees of freedom of the system are the POD amplitudes a_1 and a_2.

Thus, we now aim to find a nonlinear dynamical system

$$\begin{aligned}\frac{\mathrm{d}a_1}{\mathrm{d}t} &= f_1(a_1, a_2),\\ \frac{\mathrm{d}a_2}{\mathrm{d}t} &= f_2(a_1, a_2),\end{aligned} \tag{9.9}$$

where $f_1 : \mathbb{R}^2 \to \mathbb{R}$ and $f_2 : \mathbb{R}^2 \to \mathbb{R}$ are two unknown functions to be identified with SINDy. For the sake of simplicity, we will assume that these two functions are polynomial functions of a_1 and a_2. In general, any basis functions may be used in the SINDy library, although polynomials appear to be a reasonable choice for fluid systems, based on the quadratic nonlinearity in the Navier–Stokes equations. Given time series of a_1 and a_2, we thus define a library of candidate atoms

$$\Theta(a_1, a_2) = \begin{bmatrix}1 & a_1 & a_2 & a_1^2 & a_1a_2 & a_2^2 & a_1^3 & a_1^2a_2 & a_1a_2^2 & a_2^3\end{bmatrix}$$

so that the unknown system can be recast as

$$\begin{aligned}\frac{\mathrm{d}a_1}{\mathrm{d}t} &= \Theta(a_1, a_2)\boldsymbol{\xi}_1,\\ \frac{\mathrm{d}a_2}{\mathrm{d}t} &= \Theta(a_1, a_2)\boldsymbol{\xi}_2,\end{aligned} \tag{9.10}$$

where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are the solutions of a sparsity-promoting regression problem. After some cross-validation, the following system has been identified:

$$\begin{aligned}\frac{\mathrm{d}a_1}{\mathrm{d}t} &= 0.09a_1 - 0.77a_2 - 0.016(a_1^2 + a_2^2)a_1 - 0.07(a_1^2 + a_2^2)a_2,\\ \frac{\mathrm{d}a_2}{\mathrm{d}t} &= 0.8a_1 + 0.18a_2 + 0.06(a_1^2 + a_2^2)a_1 - 0.03(a_1^2 + a_2^2)a_2.\end{aligned} \tag{9.11}$$

As for the POD-Galerkin reduced-order model derived in Section 9.3, let us first investigate whether the identified model captures the key physics of the problem before discussing its accuracy.

9.4.3 Does the model capture the key physics?

In order to capture the key physics, the identified model (9.11) needs to fulfill the same conditions as those fulfilled by the POD-Galerkin reduced-order model, namely:

Property 1: The model has a single fixed point located at $\boldsymbol{a} = \mathbf{0}$.

Property 2: The unstable subspace E^u of the model linearized in the vicinity of $\boldsymbol{a} = \mathbf{0}$ is two-dimensional and associated to a complex–conjugate eigenpair.

Property 3: As $t \to \infty$, the system eventually evolves toward a structurally stable limit cycle.

Anyone familiar with dynamical system theory might recognize that the model (9.11) identified with SINDy corresponds to the normal form of a supercritical Andronov–Poincaré–Hopf bifurcation whose phase portrait is depicted in Figure 9.13. As such, the identified model fulfills all three conditions at once and thus captures the key physics of the problem. Identifying such a normal form is consistent with earlier works on the same flow configuration [102, 94, 123, 77].

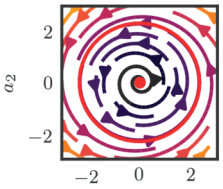


Figure 9.13: Phase plane of the low-order model identified using SINDy. The red dot indicates the linearly unstable fixed point while the red circle highlights the attracting limit cycle.

Before discussing its accuracy, let us make use of the nonlinear correlations identified in Section 9.4.1 to recast the present model as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} &= \begin{bmatrix} 0.09(1-0.19a_\Delta) & -0.77(1+0.09a_\Delta) \\ 0.8(1+0.07a_\Delta) & 0.18(1-0.18a_\Delta) \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \\ a_\Delta &= 0.41(a_1^2 + a_2^2). \end{aligned} \tag{9.12}$$

In this form, the identified model strongly underlines the nonlinear feedback mechanism existing between the vortex shedding described by a_1 and a_2 and the induced distortion characterized by a_Δ. It can moreover be understood as a low-dimensional counterpart of the self-consistent model proposed by Mantič-Lugo et al. [68] wherein the "instantaneous" mean flow $\bar{\boldsymbol{u}}$ is governed by

$$\nabla \cdot (\bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{u}}) + \nabla \bar{p} - \frac{1}{\mathrm{Re}}\nabla^2 \bar{\boldsymbol{u}} = -\nabla \cdot (\overline{\boldsymbol{u}' \otimes \boldsymbol{u}'}),$$

with $\overline{\boldsymbol{u}' \otimes \boldsymbol{u}'}$ being the fluctuation's Reynolds stress tensor, while the fluctuation itself is governed by the Navier–Stokes equations linearized in the vicinity of the "instantaneous" mean flow

$$\frac{\partial \boldsymbol{u}'}{\partial t} + \nabla \cdot (\bar{\boldsymbol{u}} \otimes \boldsymbol{u}' + \boldsymbol{u}' \otimes \bar{\boldsymbol{u}}) = -\nabla p' + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{u}'.$$

Comparing these two models, it is quite striking that they have a similar structure and thus both describe the same physics. If one considers an infinitesimal perturbation $\boldsymbol{u}'$, its Reynolds stresses become negligible and the instantaneous mean flow $\bar{\boldsymbol{u}}$ is nothing but the linearly unstable base flow $\boldsymbol{u}_b$. However, as the amplitude of the fluctuation

grows, so do its Reynolds stresses, causing the instantaneous mean flow $\bar{\boldsymbol{u}}$ to slowly deviate from the base flow $\boldsymbol{u}_b$. Concurrently, this distortion impacts the dynamics of the fluctuation through the linearized convective term $\nabla \cdot (\bar{\boldsymbol{u}} \otimes \boldsymbol{u}' + \boldsymbol{u}' \otimes \bar{\boldsymbol{u}})$. This process then continues until the distortion $\bar{\boldsymbol{u}} - \boldsymbol{u}_b$ is such that the instantaneous growth rate of the fluctuation is zero (i. e., the amplitude of the fluctuation no longer grows), hence resulting in the marginally stable mean flow. Using the identified model, this evolution of the instantaneous growth rate of the instability as a function of the distortion is illustrated in Figure 9.14.

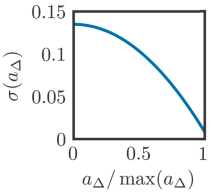


Figure 9.14: Evolution of the instantaneous growth rate σ as a function of the distortion a_Δ. As the distortion increases, the flow evolves from the linearly unstable base flow to the marginally stable mean flow.

9.4.4 How accurate is it?

Let us now assess the accuracy of the identified model compared to direct numerical simulation. The initial velocity field used in our DNS is first projected onto the span of the leading POD modes. The corresponding POD coefficients $a_1(0)$ and $a_2(0)$ are then used as the initial condition for our reduced-order model. Figure 9.15 provides a comparison of the trajectory of the system in the phase plane (a_1, a_2) obtained from direct numerical simulation (–) and predicted by our reduced-order model (– –). Surprisingly, an almost perfect agreement is obtained. Note however that this is no overfitting. Indeed, even though the two trajectories overlap in the (a_1, a_2)-plane, the corresponding temporal evolutions slightly differ due to a small underestimation of the instability growth rate as discussed shortly.

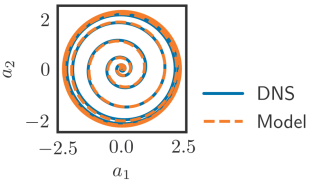


Figure 9.15: Comparison of the evolution of a_1 and a_2 obtained from direct numerical simulation (–) and predicted by the identified low-order model (– –).

Since both the identified model (9.11) and the nonlinear correlations (9.8) are solely defined in terms of the POD coefficients, it is thus quite straightforward to reconstruct an estimate of the flow field as done for the POD-Galerkin reduced-order model. Figure 9.16 depicts the evolution of the base flow-subtracted fluctuation's kinetic energy as a function of time observed in direct numerical simulation as well as the evolution predicted by the POD-Galerkin reduced-order model derived in Section 9.3 and by the present combination of the manifold model and associated nonlinear correlations. Quite clearly, the accuracy of the model proposed in the present section largely outperforms that of the classical POD-Galerkin reduced-order model. In particular, our model does not suffer from the energy overshoot as nonlinear saturation occurs nor does it display the saturation to a higher energy level once the system evolves onto the final limit cycle. However, because we use POD modes computed from the limit cycle dynamics, the flow reconstructed in the vicinity of the fixed point actually differs from the true one since these POD modes provide only a crude approximation of the instability modes. This continuous mode deformation problem can however be solved using Grassmann manifold interpolation techniques discussed in the upcoming section. Finally, Figure 9.16 also highlights that the growth rate of the instability is slightly underestimated by our model, although nothing comparable to the underestimation of the POD-Galerkin ROM. Two different approaches can be used to correct this minor flaw:

1. Instead of restricting ourselves to cubic monomials in a_1 and a_2, one can include up to seventh-order monomials in the library $\Theta(a_1, a_2)$ used for the system identification. The resulting model then corresponds to a higher-order expansion of the supercritical Hopf bifurcation normal form.
2. Alternatively, if the growth rate of the instability is known a priori, one can force the linearized low-dimensional operator to have the same eigenvalues as its high-dimensional counterpart. Such an approach then relies on constrained optimization techniques discussed in [61] and [63].

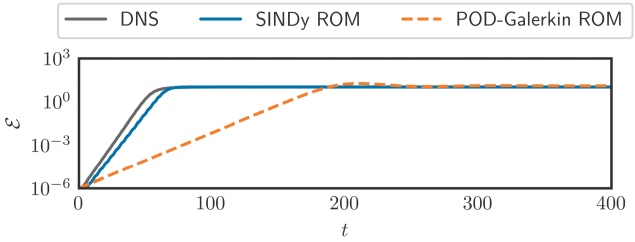


Figure 9.16: Evolution as a function of time of the base flow-subtracted fluctuation's kinetic energy $\mathcal{E}(t)$ for the DNS, the POD-Galerkin ROM derived in Section 9.3, and the model identified using SINDy. Note that, for the latter, the model predicts only the evolution of the a_1 and a_2 POD coefficients. The other coefficients (a_Δ, a_3, and a_4) are then reconstructed using the nonlinear correlations identified previously.

Although not discussed herein, both approaches have been tested and are illustrated in the accompanying Jupyter Notebook. Both of them result in a more accurate low-order model even though the resulting model is either more complex (i. e., includes higher-order terms) or requires more advanced computational techniques for the identification (i. e., constrained ℓ_1-penalized regression).

9.4.5 Solving the continuous mode deformation problem: Grassmann manifold interpolation

The previous section highlighted how the transient and posttransient dynamics of the two-dimensional cylinder flow could be modeled by a simple self-exciting self limiting quasi-harmonic oscillator whose degrees of freedom correspond to the amplitudes $a_1(t)$ and $a_2(t)$ of the two leading POD modes. If one considers only the shift mode and the first two pairs of POD modes computed from the limit cycle dynamics, the instantaneous fluctuating velocity field $\boldsymbol{u}'(\boldsymbol{x}, t)$ is then approximated by

$$\boldsymbol{u}'(\boldsymbol{x}, t) \simeq \boldsymbol{u}_{\text{pod}}(\boldsymbol{x}, t) = \boldsymbol{u}_\Delta(\boldsymbol{x}) a_\Delta(t) + \sum_{i=1}^{4} \boldsymbol{u}_i(\boldsymbol{x}) a_i(t). \tag{9.13}$$

It must be noted, however, that while the above Galerkin expansion provides a highly accurate approximation of the velocity field once the flow evolves onto the limit cycle, it poorly approximates the fluctuation's velocity field during the phase of exponential growth. This is illustrated in Figure 9.19, which depicts the instantaneous relative error

$$\text{Err}(t) = \frac{\|\boldsymbol{u}'(\boldsymbol{x}, t) - \boldsymbol{u}_{\text{pod}}(\boldsymbol{x}, t)\|^2}{\|\boldsymbol{u}'(\boldsymbol{x}, t)\|^2}.$$

As shown, the relative error for the POD reconstruction during the initial stage of transition is of the order of 50 %. This mismatch results from the inability of the Galerkin expansion (9.13) to capture the continuous mode deformation taking place as the system evolves from the vicinity of the base flow to that of the mean flow.

One way to circumvent this issue is to reconstruct the flow field based on the following parameterized Galerkin expansion

$$\boldsymbol{u}'(\boldsymbol{x}, t) \simeq \boldsymbol{u}_{\mathcal{G}}(\boldsymbol{x}, t) = \boldsymbol{u}_\Delta(\boldsymbol{x}, a_\Delta) a_\Delta(t) + \sum_{i=1}^{4} \boldsymbol{u}_i(\boldsymbol{x}, a_\Delta) a_i(t). \tag{9.14}$$

In [75, 57, 103], the parameterized expansion modes were computed simply by linearly interpolating between the instability modes obtained from linear stability analysis and the POD modes from the limit cycle dynamics. Although extremely simple to implement, the elements of the resulting reduced-order basis unfortunately do not form in general an orthonormal set of vectors. Taking into account the fact that the instability modes continuously deform into the POD modes as the system evolves onto the

low-dimensional manifold structuring its phase space, a better reduced-order basis can however be obtained using so-called *Grassmann manifold interpolation*. Such an interpolation technique has been used in [3, 2] to derive linear parameterized reduced-order models for aeroelastic problems. Detailed mathematical derivation of the interpolation scheme is beyond the scope of the present contribution and only the resulting algorithmic implementation will be described hereafter. Interested readers are referred to the PhD thesis of Amsallem [2] for more details. Note moreover that Grassmann manifold interpolation is also covered in Chapter 9 of Volume 1 of the present book series.

Let us consider the linearly unstable base flow and the marginally stable mean flow as two different operating points of the same system parameterized by the relative distortion $s = \frac{a_\Delta}{\max a_\Delta}$. The base flow thus corresponds to $s_0 = 0$, while the mean flow corresponds to $s_1 = 1$. Furthermore, let us denote by $\mathbf{\Phi}_0 \in \mathbb{R}^{n\times 5}$ a basis of POD modes computed from the snapshots taken during the phase of exponential growth (hereafter denoted as *weakly nonlinear POD modes*, see the first row of Figure 9.18), while the POD basis computed from the mean flow will be denoted as $\mathbf{\Phi}_1 \in \mathbb{R}^{n\times 5}$. Finally, let us introduce the Grassmann manifold of $n \times 5$ orthonormal matrices $\mathcal{G}(n, 5)$ and denote by ϕ_0 and ϕ_1 the coordinates associated with our two previous bases on this manifold. Given $\mathbf{\Phi}_0$ and $\mathbf{\Phi}_1$, our goal is thus to compute $\mathbf{\Phi}(s)$, i. e., the reduced-order basis for $s \in [0, 1]$, under the constraint that it has to live onto $\mathcal{G}(n, 5)$. A simple three-step procedure has been derived by [3] for that purpose:

1. Compute the projection of $\mathbf{\Phi}_1$ onto the tangent space of the Grassmann manifold $\mathcal{G}(n, 5)$ at the point ϕ_0. This projection onto the tangent space is given by the so-called logarithmic operator at point ϕ_0

$$\begin{aligned}(\mathcal{I} - \mathbf{\Phi}_0\mathbf{\Phi}_0^T)\mathbf{\Phi}_1(\mathbf{\Phi}_0^T\mathbf{\Phi}_1)^{-1} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,\\ \mathbf{\Gamma} &= \boldsymbol{U}\tan^{-1}(\boldsymbol{\Sigma})\boldsymbol{V}^T,\end{aligned} \tag{9.15}$$

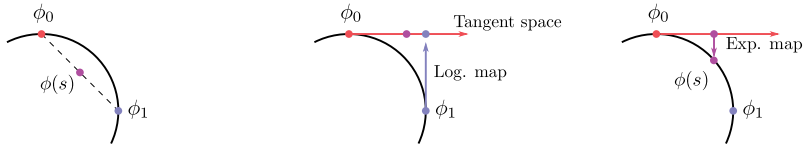
   with $\mathbf{\Gamma}$ being the projection of $\mathbf{\Phi}_1$ onto the tangent space considered.
2. Because this tangent space is flat, one can use simple linear interpolation to obtain $\mathbf{\Gamma}(s)$, i. e., the projection of the yet-unknown basis $\mathbf{\Phi}(s)$ onto the tangent space of the Grassmann manifold at ϕ_0. We then have

$$\mathbf{\Gamma}(s) = \boldsymbol{U}(s\ \tan^{-1}(\boldsymbol{\Sigma}))\boldsymbol{V}^T. \tag{9.16}$$

   Note that, by construction, $\mathbf{\Gamma}(0) = \mathbf{0}$.
3. Finally, the projection back onto the Grassmann manifold $\mathcal{G}$ is computed by the so-called exponential operator at point ϕ_0 given by

$$\mathbf{\Phi}(s) = \mathbf{\Phi}_0\boldsymbol{V}\cos(s\ \tan^{-1}(\boldsymbol{\Sigma})) + \boldsymbol{U}\sin(s\ \tan^{-1}(\boldsymbol{\Sigma})). \tag{9.17}$$

The overall procedure is schematically represented in Figure 9.17b. Note that, by construction, the reduced-order basis $\mathbf{\Phi}(s)$ is orthonormal and continuously varies from



(a) Linear interpolation (b) Grassmann manifold interpolation

Figure 9.17: Illustration of different reduced-order basis interpolation techniques; $\boldsymbol{\phi}_0$ denotes our reference point (i. e., the weakly nonlinear POD basis) and $\boldsymbol{\phi}_1$ corresponds to the mean flow operating condition for which we use the classical POD modes. The parameter s is the relative amplitude of the distortion for which we want to interpolate the corresponding reduced-order basis $\boldsymbol{\phi}(s)$. The black thick line highlights the manifold onto which our reduced-order bases should live.

$\boldsymbol{\Phi}_0$ for $s = 0$ to $\boldsymbol{\Phi}_1$ for $s = 1$. This is illustrated in Figure 9.18 wherein the vorticity field of the instantaneous shift mode and the corresponding first and second harmonics of the vortex shedding are shown for various values of the relative distortion s, namely, $s = 0$, 0.25, 0.5, 0.75, and 1. Finally, Figure 9.19a depicts the evolution as a function of time of the relative projection error

$$\mathrm{Err}(t) = \frac{\|(\mathcal{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T)\boldsymbol{u}'(\boldsymbol{x}, t)\|^2}{\|\boldsymbol{u}'(\boldsymbol{x}, t)\|^2},$$

where $\boldsymbol{\Phi}$ is either given by the classical POD basis $\boldsymbol{\Phi}_1$ or the one obtained from Grassmann manifold interpolation $\boldsymbol{\Phi}(s)$. Although both bases have the same cardinality, the one parameterized by the instantaneous relative distortion s largely outperforms the classical POD one in terms of reconstruction accuracy, notably during the phase of exponential growth. This is particularly visible in Figure 9.19b depicting the spatial distribution of the projection error. These results further confirm the inherent low-dimensionality of the problem considered despite the continuous mode deformation occurring as nonlinear saturation takes place.

9.4.6 Limitations of the present approach

Although the POD-Galerkin reduced-order model derived in Section 9.3 was able to capture the key physics of the problem investigated, it nonetheless suffers from a number of major limitations listed in Section 9.3.5. On the other hand, the present section illustrated how one could identify a highly accurate and interpretable low-order model of the system by taking into account nonlinear correlations in the POD decomposition and the existence of a low-dimensional manifold. The existence of this low-dimensional manifold moreover enabled us to propose a highly accurate parameterized projection basis largely outperforming classical POD-Galerkin expansion of the velocity, notably in the initial stage of transition where the fluctuation's velocity field is well approximated by the instability modes rather than the POD ones.

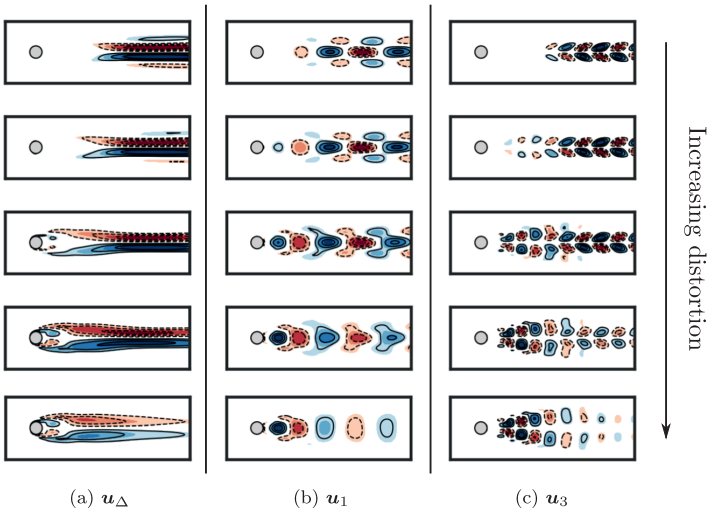


Figure 9.18: Evolution of the different POD modes obtained by Grassmann manifold interpolation as the flow evolves from the linearly unstable base flow (top) to the marginally stable mean flow (bottom). The intermediate rows correspond to a relative distortion of 25%, 50%, and 75%, respectively. Column (a) depicts the shift mode $\boldsymbol{u}_\Delta$, (b) depicts the first harmonics of the vortex shedding, and (c) depicts the second harmonics. Note that, for each value of the relative distortion, these modes form an orthonormal set of vectors.

To the best of our knowledge, the present reduced-order model is the lowest-dimensional and yet most accurate reduced-order model capturing the transient and posttransient dynamics of the two-dimensional cylinder flow. Note moreover that the exact same methodology is likely to be directly applicable to any other flow configuration exhibiting similar dynamics. Despite its impressive accuracy, one must however remain conscious that the methodology proposed herein also has some limitations. First and foremost, the identification of the reduced-order model relied on the existence of a low-dimensional manifold and on our ability to define a corresponding nonlinear embedding of the original high-dimensional data. Although such low-dimensional nonlinear manifolds are likely to exist for a large class of dissipative dynamical systems, they may however be higher-dimensional and/or more complicated to capture. Nonetheless, in such cases one could use advanced techniques from manifold learning such as kPCA [92, 93], Isomap [107], LLE and its variants [84, 121, 35], spectral embedding [10], MDS [17], or autoencoders [13].

Secondly, we assumed that the right-hand side $f(\boldsymbol{a})$ of our low-order model

$$\frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} = f(\boldsymbol{a})$$

could be expressed as a linear combination of monomials in a_1 and a_2. While this choice may be justified for a large class of dynamical systems, the present choice precludes the identification of systems involving other types of nonlinearities, such as

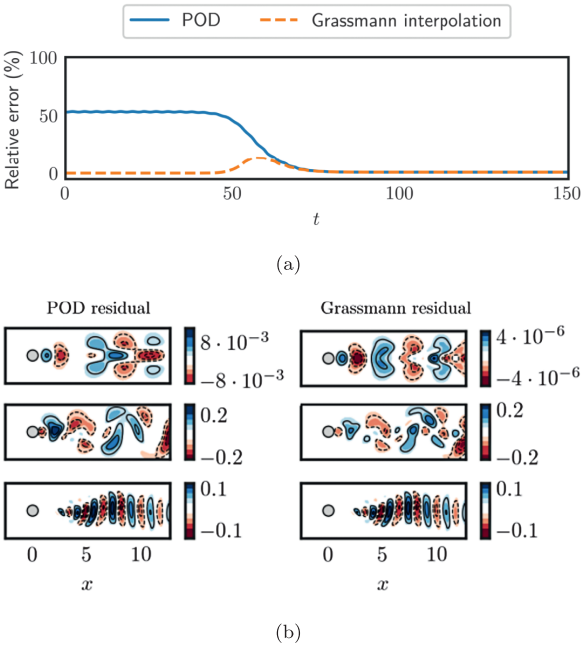


Figure 9.19: (a) Comparison of the relative error for the orthogonal projection of the base flow-subtracted fluctuation's velocity field onto either the leading five POD modes (–) extracted from the limit cycle dynamics or the Grassmann interpolated ones (– –). The direct numerical simulation has been started from an initial condition close to the linearly unstable base flow. (b) Spatial distribution of the projection error at various times. The vertical velocity component is shown. From top to bottom: $t = 6$ (exponential growth of the instability), $t = 60$ (onset of nonlinear saturation), and $t = 120$ (asymptotic limit cycle).

rational functions. It must be noted however that the SINDy framework is quite extensible and various extensions have been proposed since [22] to enable the identification of dynamical systems with exotic nonlinearities; see for instance [66]. Alternatively, if the dynamics appear to be strongly nonlinear and not expressible in terms of classical analytical functions, one could include wavelets in the library $\Theta(\boldsymbol{a})$ used in the identification or turn to a class of neural networks known as *long short-term memory* (LSTM). Although one would sacrifice interpretability by doing so, recent works have shown that such LSTM deep neural networks are able to capture and reproduce the chaotic spatio-temporal dynamics of the Kuramoto-Sivashinky equation [112, 26].

9.5 Good practices

The two-dimensional cylinder flow at $\mathrm{Re} = 100$ is a prototypical example from fluid dynamics capturing the key physics of bluff body flows. Despite the low-dimensionality

of the flow dynamics, it has been shown that a reduced-order model derived from a naive POD-Galerkin projection procedure fails to accurately reproduce the dynamics of the flow, most notably its transient dynamics. The key reasons for this failure, explained in [77], are twofold:

1. Galerkin projection of the Navier–Stokes equations onto the span of a low-dimensional POD basis causes a disruption of the energy cascade, hence giving rise to the energy overshoot illustrated in Figure 9.9.
2. POD modes are classically computed from statistically steady operating conditions. Consequently, this set of modes may provide only a crude approximation of the fluctuation's velocity field during transient dynamics. As a consequence, the corresponding low-dimensional linear operator obtained from Galerkin projection does not correctly capture the spectral properties of its high-dimensional counterpart.

Recent advances in data-driven techniques and machine learning are likely to help overcoming these limitations. It must be emphasized however that, despite their impressive successes regularly reported in mainstream and scientific media, blindly applying techniques from machine learning (and in particular from deep learning) to fluid dynamics problems may give rise to overly complicated models. The aim of this section is to discuss a set of good practices that, according to the authors, are of crucial importance when it comes to data-driven reduced-order modeling.

9.5.1 Dimensionality reduction

The aim of reduced-order modeling is to obtain a low-dimensional representation of the dynamics of the original high-dimensional system. The very first step is thus to apply *dimensionality reduction*. POD, which is discussed at length in this book series, is the standard choice in mechanical engineering due to its ability to rank the modes according to the fraction of the fluctuation's kinetic energy they capture. Once the POD modes have been computed, most of the reduced-order models proposed in the literature then carry on directly with the derivation of the low-dimensional model governing the dynamics of these modes. It must be noted, however, that, as discussed in the previous section, POD analysis provides a set of modes whose temporal evolutions are only linearly uncorrelated. Hence, truncated POD corresponds simply to an optimal linear embedding of our original high-dimensional data set into a lower linear subspace. While this property might be beneficial for reduced-order models of linear systems, dissipative nonlinear dynamical systems are typically characterized by nonlinear correlations across vastly different ranges of temporal and/or spatial scales. Consequently, if the data turn out to live on a low-dimensional nonlinear manifold, POD analysis would then overestimate the number of dimensions required to describe the dynamics of the system. Accounting for these nonlinear correlations is thus a key

problem for standard reduced-order modeling strategies which is often disregarded by practitioners, although it may cause the identified/derived reduced-order model to be unnecessarily complicated.

Looking for nonlinear correlations between the various features of a multivariate time series is obviously significantly more complicated than looking for simple linear correlations. Given the quadratic nature of the nonlinear convective term in the Navier–Stokes equations, it seems however reasonable to restrict ourselves to polynomial correlations. Moreover, when the investigated flow exhibits only periodic dynamics as for the one considered herein, one can simply guess a priori the variables involved in the correlations by considering a limited number of triadic interactions. Polynomial regression can then be used to unravel the exact form of these nonlinear correlations. For more complicated flow configurations (e. g., chaotic and/or higher-dimensional dynamics), this task can however quickly become intractable without further preprocessing. Recently, Lopez-Paz et al. [64] proposed a new correlation metric to unravel whether two features of a multivariate time series are nonlinearly correlated or not: the *randomized dependence coefficient* (RDC). Mathematical derivation of this metric is far beyond the scope of this contribution and interested readers are referred to the original paper [64] for more details. Note that this nonlinear correlation metric is extremely simple to use and can be implemented with less than 10 lines of R or Python. Preliminary results on a high Reynolds number shear-driven cavity flow have shown that the shear-layer dynamics and inner-cavity flow were only weakly nonlinearly correlated, thus considerably simplifying the identification of a reduced-order model with only four degrees of freedom. As an element of comparison, a classical POD-Galerkin reduced-order model would involve 12 to 15 degrees of freedom.

Although the combination of POD, RDC analysis, and polynomial regression has now become one of the standard approaches used by the present authors, it must be noted that numerous other alternatives exist to unravel nonlinear correlations. In the field of machine learning, these tools form a subset known as *manifold learning* or *representation learning*. From the authors' point of view, a particularly interesting technique from manifold learning is the use of so-called *autoencoders*. This is the subject of ongoing investigations by the present authors. For more details about autoencoders and manifold learning, please see the excellent review article by Bengio et al. [13].

9.5.2 System identification

The field of system identification uses statistical methods to build mathematical models of dynamical systems from measured data. With respect to the classification proposed in [115], system identification enables us to obtain either *gray-box* or *black-box* models. Various methods have been proposed over the years. Some of these are classified in Figure 9.20 depending on the complexity (linear or nonlinear, interpretable or noninterpretable) of the resulting model. While the identification of

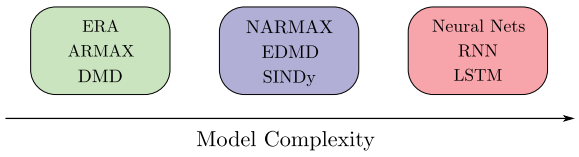


Figure 9.20: Classification of various system identification techniques based on the complexity of the resulting model. On the left, these techniques and their variants enable the identification of linear input–output models. At the center, NARMAX, EDMD, and SINDy allow one to identify interpretable input–output nonlinear dynamical systems. Finally, on the right, neural networks and their variants give rise to black-box strongly nonlinear models.

a linear time-invariant dynamical system has a plethora of theoretical results, theoretical guarantees for nonlinear system recovery are much more scarce. Like many fields, nonlinear system identification has nonetheless been revolutionized with the popularization of deep learning. It must be noted however that, from the authors' point of view, a number of recent studies have put too much emphasis on illustrating deep learning techniques while discarding the possibility that the system considered could be modeled using a much simpler approach, notably studies which have used the two-dimensional cylinder flow as an illustration. Following Ockham's razor, we thus strongly encourage practitioners to try linear system identification first (e. g., ERA, DMD, ARMAX), before moving to interpretable nonlinear system identification (e. g., NARMAX, SINDy) and eventually neural network-based techniques only if the previous two approaches have failed.

9.6 Conclusion

This work proposes a new reduced-order modeling procedure for unsteady fluid flows that yields accurate nonlinear models and insight into relevant flow structures. This procedure identifies sparse interpretable nonlinear models, not on the full fluid state, but from time-resolved measurements of the leading POD coefficients that may be realistically obtained in experiments. The sparsity of the model prevents overfitting and uncovers key nonlinear interaction terms. Although models are data-driven, they are interpretable, and it is also possible to incorporate partial prior knowledge of the physics or constraints to improve the models. If the stability modes are also available, it is possible to estimate the full state from the sparse model using Grassmann manifold interpolation: The full state is expanded in terms of a parameterized reduced-order basis, based on the dynamics.

This methodology is illustrated using the canonical two-dimensional cylinder flow at Re = 100. Despite its simplicity, this flow configuration is a prototypical example capturing the key physics of bluff body flows. Even though this study uses data

from direct numerical simulations, the overall strategy is generally applicable to a real flow experiment with minor modifications. Despite their simplicity, the identified models do not suffer the same drawbacks as reduced-order models obtained from a Galerkin projection procedure, namely, overestimation of the duration of transients and energy overshoots at the onset of nonlinear saturation. Instead, the identified sparse models provide simple explanations for the nonlinear saturation process of globally unstable flows. Moreover, the models are based on sensor measurements, which may include POD coefficients, lift, drag, or pressure measurements that are physically linked to the geometry. Working in these *intrinsic* coordinates has the potential to overcome many of the limitations of classical modal-based projection methods, including mode deformation due to moving geometry and varying parameters.

9.7 Perspectives

The effectiveness of the reduced-order models identified and the modularity of the methodology proposed in the present work suggest a number of exciting future directions. There is significant potential for these methods to be applied broadly to obtain interpretable reduced-order models for a range of flow configurations in simulations and experiments. For example, these manifold models may be applied to develop nonlinear unsteady aerodynamic models, generalizing previous linear and linear parameter-varying models [19, 20, 47].

A key motivation in this work is its extension to flow control. Given a feature vector $\boldsymbol{a}$ and actuators characterized by a control law $\boldsymbol{b}(t)$, one could use SINDy with control [23, 52] in order to identify low-order models

$$\frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b})$$

that incorporate the influence of the actuation $\boldsymbol{b}$ on the dynamics of $\boldsymbol{a}$. Combining such an approach with *machine learning control* [36] may result in interpretable models of entirely new flow behaviors and previously unobserved flow physics that are discovered in the controlled flow. The identified models can then serve as a low-dimensional representation of the actual system in order to facilitate the computation of nonlinear optimal feedback control laws. This is an area of active research by the authors. In the near future, the authors aim to apply the methodology introduced in the present work to the optimal control of experimental flows.

There are a number of methodological extensions that may improve the performance of this sparse modeling framework. First, it will be important to demonstrate that these methods scale favorably to systems with higher-dimensional attractors. Because the algorithms are based on simple regression and sparse optimization, they

should remain computationally tractable. Next, it may be possible to increase the accuracy of the Grassmann interpolation by building local modal libraries in different dynamic regimes (e. g., linear instability, saturated limit cycle, etc.). The storage requirements may further be reduced using compression techniques and sparse sampling. Finally, it has been demonstrated in [63] how such manifold models could be identified directly from sensor measurements such as the lift and drag coefficients. For the present flow configuration, the present authors identified that the dynamical system governing the dynamics of the lift coefficient $C_L(t)$ of the form

$$\frac{\mathrm{d}^2 C_L}{\mathrm{d}t^2} + \left(\sigma - \left[\alpha C_L^2 + \beta\left(\frac{\mathrm{d}C_L}{\mathrm{d}t}\right)^2\right]\right)\frac{\mathrm{d}C_L}{\mathrm{d}t} + \omega_0^2 C_L = 0.$$

Such sensor-based models are strongly related to the existence of a low-dimensional manifold structuring the phase space of the system investigated and to the strong correlations existing between the various sensor measurements considered and the spatio-temporal coherent structures found in the flow. Our ability to identify such sensor-based manifold models may eventually have a major impact in experimental fluid mechanics and flow control.

A data-driven generalization of manifold models are cluster-based network models, where the snapshots are coarse-grained by centroids and the topology is encoded in a transition model between these centroids [58]. Such models may approximate broadband-frequency wall turbulence for dozens of different wall surface actuations [39]. The price for for this conceptually simple, automatable, and robust reduced-order modeling avenue is that the manifold and sparse dynamics still need to be distilled—if they exist.

Bibliography

[1] E. Åkervik, L. Brandt, D. S. Henningson, J. Hœpffner, O. Marxen, and P. Schlatter, Steady solutions of the Navier-Stokes equations by selective frequency damping, *Phys. Fluids*, **18** (6) (2006), 068102.

[2] D. Amsallem, *Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions*. PhD thesis, Stanford University, 2010.

[3] D. Amsallem and C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, *AIAA J.*, **46** (7) (2008), 1803–1813.

[4] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, *J. Fluid Mech.*, **192** (1988), 115–173.

[5] H. Babaee and T. P. Sapsis, A variational principle for the description of time-dependent modes associated with transient instabilities, *Philos. Trans. R. Soc. Lond.*, **472** (2186) (2016), 20150779.

[6] S. Bagheri, E. Åkervik, L. Brandt, and D. S. Henningson, Matrix-free methods for the stability and control of boundary layers, *AIAA J.*, **47** (5) (2009), 1057–1068.

[7] M. Balajewicz, E. H. Dowell, and B. R. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation, *J. Fluid Mech.*, **729** (2013), 285–308.

[8] D. Barkley and R. D. Henderson, Three-dimensional Floquet stability analysis of the wake of a circular cylinder, *J. Fluid Mech.*, **322** (1) (1996), 215.

[9] D. Barkley, Linear analysis of the cylinder wake mean flow, *Europhys. Lett.*, **75** (5) (2006), 750–756.

[10] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.*, **15** (2003), 1373–1396.

[11] S. Beneddine, D. Sipp, A. Arnault, J. Dandois, and L. Lesshafft, Conditions for validity of mean flow stability analysis, *J. Fluid Mech.*, **798** (2016), 485–504.

[12] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Solveira, *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, 2020.

[13] Y. Bengio, A. Courville, and P. Vincent, Representation learning: a review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 1798–1828.

[14] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.*, **25** (1) (1993), 539–575.

[15] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Paperbackshop UK Import, 2013.

[16] J. Bongard and H. Lipson, Automated reverse engineering of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA*, **104** (24) (2007), 9943–9948.

[17] I. Borg and P. Groenen, Modern multidimensional scaling: theory and applications, *J. Educ. Meas.*, **40** (2003), 277–280.

[18] R. Bourguet, M. Braza, and A. Dervieux, Reduced-order modeling of transonic flows around an airfoil submitted to small deformations, *J. Comput. Phys.*, **230** (2011), 159–184.

[19] S. L. Brunton, C. W. Rowley, and D. R. Williams, Reduced-order unsteady aerodynamic models at low Reynolds numbers, *J. Fluid Mech.*, **724** (2013), 203–233.

[20] S. L. Brunton, S. T. M. Dawson, and C. W. Rowley, State-space model identification and feedback control of unsteady aerodynamic forces, *J. Fluids Struct.*, **50** (2014), 253–270.

[21] S. L. Brunton and B. R. Noack, Closed-loop turbulence control: progress and challenges, *Appl. Mech. Rev.*, **67** (5) (2015), 050801.

[22] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA*, **113** (15) (2016), 3932–3937.

[23] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sparse identification of nonlinear dynamics with control (SINDYc), *IFAC-PapersOnLine*, **49** (18) (2016), 710–715.

[24] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Chaos as an intermittently forced linear system, *Nat. Commun.*, **8** (2017), 1.

[25] S. L. Brunton, B. R. Noack, and P. Koumotsakos, Machine learning for fluid dynamics, *Annu. Rev. Fluid Mech.*, **52** (2020), 477–508.

[26] M. A. Bucci, O. Semeraro, A. Allauzen, G. Wisniewski, L. Cordier, and L. Mathelin, Control of chaotic systems by deep reinforcement learning, *Proc. R. Soc. A*, **475** (2231) (2019), 20190351.

[27] M. Carini, F. Auteri, and F. Giannetti, Centre-manifold reduction of bifurcating flows, *J. Fluid Mech.*, **767** (2015), 109–145.

[28] K. Carlberg, R. Tuminaro, and P. Boggs, Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics, *SIAM J. Sci. Comput.*, **37** (2) (2015), B153–B184.

[29] K. Carlberg, M. Barone, and H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, *J. Comput. Phys.*, **330** (2017), 693–734.
[30] J.-M. Chomaz, Global instabilities in spatially developing flows: non-normality and nonlinearity, *Annu. Rev. Fluid Mech.*, **37** (2005), 357–392.
[31] L. Cordier, B. R. Noack, G. Daviller, J. Delvile, G. Lehnasch, G. Tissot, M. Balajewicz, and R. K. Niven, Control-oriented model identification strategy, *Exp. Fluids*, **54** (2013), 1580.
[32] M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model, *Phys. Plasmas*, **24** (2) (2017), 022310.
[33] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders, *Phys. Fluids A*, **3** (1991), 2337–2354.
[34] R. Deshmukh, J. J. McNamara, Z. Liang, J. Z. Kolter, and A. Gogulapati, Model order reduction using sparse coding exemplified for the lid-driven cavity, *J. Fluid Mech.*, **808** (2016), 189–223.
[35] D. L. Donoho and C. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, *Proc. Natl. Acad. Sci. USA*, **100** (2003), 5591–5596.
[36] T. Duriez, S. L. Brunton, and R. R. Noack, *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*, Springer International Publishing, 2017.
[37] W. S. Edwards, L. S. Tuckerman, R. A. Friesner, and D. C. Sorensen, Krylov methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **110** (1) (1994), 82–102.
[38] N. Fabbiane, O. Semeraro, S. Bagheri, and D. S. Henningson, Adaptive and model-based control theory applied to convectively unstable flows, *Appl. Mech. Rev.* (2014).
[39] D. Fernex, R. Semaan, M. Albers, P. S. Meysonnat, R. Ishar, E. Kaiser, W. Schröder, and B. R. Noack, Cluster-based network model for drag reduction mechanisms of an actuated turbulent boundary layer, *Proc. Appl. Math. Mech.*, **19**(1) article e201900219 (2019), 1–2.
[40] L. Fick, Y. Maday, A. T. Patera, and T. Taddei, A stabilized POD model for turbulent flows over a range of Reynolds numbers: optimal parameter sampling and constrained projection, *J. Comput. Phys.*, **371** (2018), 214–243.
[41] P. F. Fischer, J. W. Lottes, and S. G. Kerkemeir, Nek5000 Web pages, 2008. http://nek5000.mcs.anl.gov.
[42] G. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, *J. Fluid Mech.*, **503** (2004), 161–170.
[43] F. Giannetti and P. Luchin, Structural sensitivity of the first instability of the cylinder wake, *J. Fluid Mech.*, **581** (2007), 167.
[44] B. Glaz, L. Liu, and P. P. Friedmann, Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework, *AIAA J.*, **48** (10) (2010), 2418–2429.
[45] W. R. Graham, J. Peraire, and K. Y. Tang, Optimal control of vortex shedding using low-order models. Part I – Open-loop model development, *Int. J. Numer. Methods Eng.*, **44** (1999), 945–972.
[46] J. Guckenheimer and J. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*, Theorem 1.4.2, corrected seventh printing, Springer, 2002.
[47] M. S. Hemati, S. T. M. Dawson, and C. W. Rowley, Parameter-varying aerodynamics models for aggressive pitching-response prediction, *AIAA J.*, **55** (3) (2017), 693–701.
[48] M. Hoffmann, C. Fröhner, and F. Noé, Reactive SINDy: discovering governing reactions from concentration data, *J. Chem. Phys.*, **150** (2) (2019), 025101.
[49] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996.
[50] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.*, **656** (2010), 5–28.

[51] J.-N. Juang and R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, *J. Guid.*, **8** (5) (1985), 620–627.

[52] E. Kaiser, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proc. R. Soc. A*, **474** (2219) (2019), 20180335.

[53] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, and R. K. Niven, Cluster-based reduced-order modelling of a mixing layer, *J. Fluid Mech.*, **754** (2014), 365–414.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (6) (2017), 84–90.

[55] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM-Society for Industrial and Applied Mathematics, 2016.

[56] C. Lee, J. Kim, D. Babcock, and R. Goodman, Application of neural networks to turbulence control for drag reduction, *Phys. Fluids*, **9** (6) (1997), 1740–1747.

[57] O. Lehmann, M. Luchtenburg, B. R. Noack, R. King, M. Morzynski, and G. Tadmor, Wake stabilization using POD Galerkin models with interpolated modes, in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 500–505, IEEE, 2005.

[58] H. Li, D.. Fernex, R. Semaan, J. Tan, M. Morzyński, and B. R. Noack, Cluster-based network model, *J. Fluid Mech.*, **(in print)** (2020), 1–41.

[59] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.*, **807** (2016), 155–166.

[60] J.-Ch. Loiseau, *Dynamics and global stability analysis of three-dimensional flows*. PhD thesis, Ecole Nationale Superieure d'Arts et Metiers, 2014.

[61] J.-Ch. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, *J. Fluid Mech.*, **838** (2018), 42–67.

[62] J.-Ch. Loiseau, M. A. Bucci, S. Cherubini, and J.-Ch. Robinet, Time-stepping and Krylov methods for large-scale instability problems, in *Computational Modelling of Bifurcations and Instailities in Fluid Dynamics*, pp. 33–73, Springer, 2019.

[63] J.-Ch. Loiseau, B. R. Noack, and S. L. Brunton, Sparse reduced-order modelling: sensor-based dynamics to full-state estimation, *J. Fluid Mech.*, **844** (2018), 459–490.

[64] D. Lopez-Paz, P. Hennig, and B. Schölkopf, The randomized dependence coefficient, in C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 1–9, Curran Associates, Inc., 2013.

[65] W. V. R. Malkus, Outline of a theory of turbulent shear flow, *J. Fluid Mech.*, **1** (05) (1956), 521.

[66] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, **2** (1) (2016), 52–63.

[67] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, *Proc. R. Soc. A*, **473** (2204) (2017), 20170009.

[68] V. Mantič-Lugo, V. Arratia, and F. Gallaire, Self-consistent mean flow description of the nonlinear saturation of the vortex shedding in the cylinder wake, *Phys. Rev. Lett.*, **113** (2014), 8.

[69] V. Mantič-Lugo, C. Arratia, and F. Gallaire, A self-consistent model for the saturation dynamics of the vortex shedding around the mean flow in the unsteady cylinder wake, *Phys. Fluids*, **27** (2015), 074103.

[70] P. Meliga, Harmonics generation and the mechanics of saturation in flow over an open cavity: a second-order self-consistent description, *J. Fluid Mech.*, **826** (2017), 503–521.

[71] P. Meliga, E. Boujo, and F. Gallaire, A self-consistent formulation for the sensitivity analysis of finite-amplitude vortex shedding in the cylinder wake, *J. Fluid Mech.*, **800** (2016), 327–357.
[72] I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dyn.*, **41** (1–3) (2005), 309–325.
[73] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, *Annu. Rev. Fluid Mech.*, **45** (1) (2013), 357–378.
[74] M. Milano and P. Koumoutsakos, Neural network modeling for near wall turbulent flow, *J. Comput. Phys.*, **182** (1) (2002), 1–26.
[75] M. Morzynski, W. Stankieciz, B. R. Noack, F. Thiele, R. King, and G. Tadmor, Generalized mean-field model for flow control using a continuous mode interpolation, in *3rd AIAA Flow Control Conference*, American Institute of Aeronautics and Astronautics, 2006.
[76] A. G. Nair and K. Taira, Network-theoretic approach to sparsified discrete vortex dynamics, *J. Fluid Mech.*, **768** (2015), 549–571.
[77] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *J. Fluid Mech.*, **497** (2003), 335–363.
[78] B. R. Noack, M. Morzynski, and G. Tadmor (eds.), *Reduced-Order Modelling for Flow Control*, Springer Vienna, 2011.
[79] B. R. Noack, P. Papas, and P. A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, *J. Fluid Mech.*, **523** (2005), 339–365.
[80] J. Östh, S. Krajnović, B. R. Noack, D. Barros, and J. Borée, On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high Reynolds number flow over an Ahmed body, *J. Fluid Mech.*, **747** (2014), 518–544.
[81] O. K. Rediniotis, J. Ko, and A. J. Kurdila, Reduced order nonlinear Navier-Stokes models for synthetic jets, *J. Fluids Eng.*, **124** (2) (2002), 433–443.
[82] D. Rempfer, On low-dimensional Galerkin models for fluid flow, *Theor. Comput. Fluid Dyn.*, **14** (2000), 75–88.
[83] D. Rempfer and F. H. Fasel, Dynamics of three-dimensional coherent structures in a flat-plate boundary-layer, *J. Fluid Mech.*, **275** (1994), 257–283.
[84] S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (5500) (2000), 2323–2326.
[85] C. W. Rowley and S. T. M. Dawson, Model reduction for flow analysis and control, *Annu. Rev. Fluid Mech.*, **49** (1) (2017), 387–417.
[86] C. W. Rowley, I. Mezic, S. Bagheri, and P. Schlatter, Spectral analysis of nonlinear flows, *J. Fluid Mech.*, **641** (2009), 115.
[87] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.*, **3** (4) (2017), e1602614.
[88] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, *Proc. R. Soc. A*, **473** (2197) (2017), 20160446.
[89] H. Schaeffer and S. G. McCalla, Sparse model selection via integral terms, *Phys. Rev. E*, **96** (2017), 2.
[90] M. Schlegel and B. R. Noack, On long-term boundedness of Galerkin models, *J. Fluid Mech.*, **765** (2015), 325–352.
[91] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science*, **324** (5923) (2009), 81–85.
[92] B. Schölkopf, A. Smola, and K.-R. Müller, Kernel principal component analysis, in *International Conference on Artificial Neural Networks*, pp. 583–588, 1997.
[93] B. Schölkopf, A. Smola, and K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, **10** (5) (1998), 1299–1319.

[94] M. Schumm, E. Berger, and P. A. Monkewitz, Self-excited oscillations in the wake of two-dimensional bluff bodies and their control, *J. Fluid Mech.*, **271** (1994), 17.
[95] O. Semeraro, F. Lusseyran, L. Pastur, and P. Jordan, Qualitative dynamics of wave packets in turbulent jets, *Phys. Rev. Fluids*, **2** (2017), 9.
[96] D. Sipp and A. Lebedev, Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows, *J. Fluid Mech.*, **593** (2007).
[97] D. Sipp, O. Marquet, P. Meliga, and A. Barbagallo, Dynamics and control of global instabilities in open-flows: a linearized approach, *Appl. Mech. Rev.*, **63** (3) (2010), 030801.
[98] D. Sipp and P. J. Schmid, Linear closed-loop control of fluid instabilities and noise-induced perturbations: a review of approaches and tools, *Appl. Mech. Rev.*, **68** (2) (2016), 020801.
[99] S. Sirisup and G. E. Kaniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, *J. Comput. Phys.*, **194** (1) (2004), 92–116.
[100] L. Sirovich, Turbulence and the dynamics of coherent structures. I – Coherent structures. II – Symmetries and transformations. III – Dynamics and scaling, *Q. Appl. Math.*, **45** (1987), 561–571.
[101] M. Sorokina, S. Sygletos, and S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method, *Opt. Express*, **24** (26) (2016), 30433–30443.
[102] K. R. Sreenivasan, P. J. Strykowski, and D. J. Olinger, Hopf bifurcation, Landau equation, and vortex shedding behind circular cylinders, in *Forum on Unsteady Flow Separation*, vol. 1, pp. 1–13, American Society for Mechanical Engineers, Fluids Engineering Division New York, 1987.
[103] W. Stankiewicz, M. Morzynski, K. Kotecki, and B. R. Noack, On the need of mode interpolation for data-driven Galerkin models of a transient flow around a sphere, *Theor. Comput. Fluid Dyn.*, **31** (2) (2016), 111–126.
[104] P. J. Strykowski and K. R. Sreenivasan, On the formation and suppression of vortex 'shedding' at low Reynolds numbers, *J. Fluid Mech.*, **218** (1) (1990), 71.
[105] G. Tadmor, O. Lehmann, B. R. Noack, L. Cordier, J. Delville, J.-P. Bonnet, and M. Morzyński, Reduced order models for closed-loop wake control, *Philos. Trans. R. Soc. A*, **369** (1940) (2011), 1513–1524.
[106] G. Tadmor, O. Lehmann, B. R. Noack, and M. Morzyński, Mean field representation of the natural and actuated cylinder wake, *Phys. Fluids*, **22** (3) (2010), 034102.
[107] J. B. Tenenbaum, V. De Silva, and J. C. Langford, A global parametric framework for nonlinear dimensionality reduction, *Science*, **290** (2000), 2319–2323.
[108] V. Theofilis, Global linear instability, *Annu. Rev. Fluid Mech.*, **43** (2011), 319–352.
[109] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: theory and applications, *J. Comput. Dyn.*, **1** (2) (2014), 391–421.
[110] S. E. Turton, L. S. Tuckerman, and D. Barkley, Prediction of frequencies in thermosolutal convection from mean flows, *Phys. Rev. E*, **91** (4) (2015), 0403009.
[111] L. Ukeiley, L. Cordier, R. Manceau, J. Delville, J. P. Bonnet, and M. Glauser, Examination of large-scale structures in a turbulent plane mixing layer. Part 2. Dynamical systems model, *J. Fluid Mech.*, **441** (2001), 61–108.
[112] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Komoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, *Proc. R. Soc. A*, **474** (2018), 20170844.
[113] Z. Wang, I. Akthar, J. Borggaard, and T. Ilescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, *Comput. Methods Appl. Mech. Eng.*, **237–240** (2012), 10–26.
[114] M. Wei and C. W. Rowley, Low-dimensional models of a temporally evolving free shear layer, *J. Fluid Mech.*, **618** (2009), 113–134.

[115] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine*, 1st, MIT Press, Boston, 1948.
[116] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, A data–driven approximation of the Koopman operator: extending dynamic mode decomposition, *J. Nonlinear Sci.*, **25** (6) (2015), 1307–1346.
[117] C. H. K. Williamson, Vortex dynamics in the cylinder wake, *Annu. Rev. Fluid Mech.*, **28** (1) (1996), 477–539.
[118] A. Zebib, Stability of viscous flow past a circular cylinder, *J. Eng. Math.*, **21** (2) (1987), 155–165.
[119] H.-Q. Zhang, U. Fey, B. R. Noack, M. König, and H. Eckelmann, On the transition of the cylinder wake, *Phys. Fluids*, **7** (4) (1995), 779–794.
[120] W. Zhang, B. Wang, Z. Ye, and J. Quan, Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models, *AIAA J.*, **50** (5) (2012), 1019–1028.
[121] Z. Zhang and J. Wang, MLLE: modified locally linear embedding using multiple weights, in B. Schölkopf, J. C. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems 19*, pp. 1593–1600, MIT Press, 2007.
[122] Z. J. Zhang and K. Duraisamy, Machine learning methods for data-driven turbulence modeling, in *22nd AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2015.
[123] B. J. A. Zielinska and J. E. Wesfreid, On the spatial structure of global modes in wake flow, *Phys. Fluids*, **7** (6) (1995), 1418–1424.
[124] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Solveira, *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, 2020.

Roland Pulch

10 Model order reduction in uncertainty quantification

Abstract: Mathematical models include parameters, which are often affected by uncertainties due to measurement errors or imperfections of an industrial production, for example. In uncertainty quantification (UQ), parameter variations are often described by random variables or random processes. Of course the resulting stochastic model exhibits a higher complexity in comparison to the original model. Thus methods of model order reduction (MOR) become attractive to save computational effort in UQ. We consider dynamical systems consisting of ordinary differential equations or differential algebraic equations. The focus is on linear dynamical systems. On the one hand, state variables and output variables can be expanded into a series with given orthogonal polynomials and unknown coefficient functions. A stochastic Galerkin method yields a high-dimensional deterministic system satisfied by an approximation of the coefficient functions. A stochastic collocation method can also be written as a large weakly coupled deterministic system. We use traditional MOR methods to shrink the dimensionality of the large systems. On the other hand, quantities of interest typically represent probabilistic integrals like moments or failure probabilities, for example. Multidimensional quadrature methods and sampling techniques directly generate approximations of these statistics. Therein, the original dynamical system has to be solved many times for different realizations of the parameters. Thus high-dimensional dynamical systems cause a huge total computational effort. We discuss methods of parametric MOR to reuse a reduced-order model for different parameter values. Finally, numerical results are demonstrated for test examples, where we perform the reduction of large deterministic systems as well as parametric MOR.

Keywords: Model order reduction, uncertainty quantification, polynomial chaos, quadrature, reduced basis method

MSC 2010: 34C20, 37M99, 65D32, 65L99, 93A15

10.1 Introduction

In science and engineering, mathematical modeling often yields systems of ordinary differential equations (ODEs), differential algebraic equations (DAEs), or partial differential equations (PDEs). The systems include physical parameters or geometrical parameters, which exhibit uncertainties due to modeling errors, measurement errors, or

Roland Pulch, Universität Greifswald, Greifswald, Germany

https://doi.org/10.1515/9783110499001-010

imperfections of an industrial production. Uncertainty quantification (UQ) determines the sensitivity of the model outputs with respect to these parameter variations. Often a stochastic modeling is used, where uncertain parameters are replaced by random variables, random processes, or spatial random fields; see [44, 46]. Now the model output also becomes a random process or a random field.

The dynamical systems may be low-dimensional or high-dimensional. In any case, the presence of random variables increases the complexity of the differential equations. In some numerical methods, a stochastic discretization yields a deterministic dynamical system of a much higher dimension. Now methods of model order reduction (MOR) are attractive to decrease the complexity and thus save computational effort in the numerical simulation. Efficient MOR methods are already available for deterministic systems of differential equations or differential algebraic equations; see [2, 6, 7, 9, 16, 40].

We consider a polynomial chaos expansion (see [3, 45]) of the random quantity of interest (QoI) in a small- or medium-sized linear dynamical system. The expansion includes orthogonal basis polynomials and unknown time-dependent coefficient functions. The stochastic Galerkin method yields a coupled deterministic linear dynamical system of high dimensionality, whose solution approximates the coefficient functions; see [21, 31, 42]. MOR methods have been applied to this high-dimensional system in [15, 35, 36, 38], for example. Alternatively, a stochastic collocation technique using a quadrature rule or a sampling method provides approximations of the coefficient functions. We write the stochastic collocation method in the form of a weakly coupled deterministic linear dynamical system; see [33, 34]. Now the system is high-dimensional and thus common MOR methods can be applied.

The concept of parametric MOR (pMOR) becomes attractive in the case of parameter-dependent systems with high dimensionality. Its aim is the efficient computation of reduced-order models (ROMs) for a (possibly) large number of realizations of the parameters. Methods of pMOR and their applications are demonstrated in [5, 8, 13], for example. We address the usage of pMOR in problems of UQ. An ROM represents a surrogate model, which can be solved instead of the original dynamical system. Therein, statistics of the random QoI are computed like the moments, for example. We also discuss reduced basis methods, which can be seen as a class of pMOR methods. Reduced basis techniques are efficient for many spatial problems modeled by PDEs; see [18, 22, 30]. A rigorous investigation of reduced basis methods for UQ can be found in [11].

This chapter is organized as follows. We introduce UQ of dynamical systems and the stochastic modeling in Section 10.2. MOR of deterministic dynamical systems, which are generated by stochastic discretizations, are addressed in Section 10.3. PMOR of dynamical systems with random parameters is considered in Section 10.4. Finally, we illustrate numerical simulations of test examples for both cases in Section 10.5.

10.2 Stochastic models and methods

We review the stochastic modeling and numerical techniques to solve the arising problems in this section.

10.2.1 Dynamical systems

Let a nonlinear dynamical system be given in the form

$$\begin{aligned} E(\boldsymbol{\mu})\dot{\mathbf{x}}(t,\boldsymbol{\mu}) &= A(\boldsymbol{\mu})\mathbf{x}(t,\boldsymbol{\mu}) + \mathbf{F}\big(\mathbf{x}(t,\boldsymbol{\mu}),\boldsymbol{\mu}\big) + B(\boldsymbol{\mu})\mathbf{u}(t), \\ \mathbf{y}(t,\boldsymbol{\mu}) &= C(\boldsymbol{\mu})\mathbf{x}(t,\boldsymbol{\mu}), \end{aligned} \tag{10.1}$$

with matrices and functions depending on physical and/or geometrical parameters $\boldsymbol{\mu} \in \mathcal{D} \subseteq \mathbb{R}^p$. The sizes of the matrices are $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_{\text{in}}}$, $C \in \mathbb{R}^{n_{\text{out}}\times n}$. The system involves a nonlinear function $\mathbf{F} : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}^n$. For nonsingular matrices E, the system consists of ODEs with the state variables $\mathbf{x} : [t_0, T] \times \mathcal{D} \to \mathbb{R}^n$. For singular matrices E, a system of DAEs is given with the inner variables $\mathbf{x}$. We consider initial value problems

$$\mathbf{x}(t_0,\boldsymbol{\mu}) = \mathbf{x}_0(\boldsymbol{\mu}) \quad \text{for } \boldsymbol{\mu} \in \mathcal{D}, \tag{10.2}$$

with a predetermined function $\mathbf{x}_0 : \mathcal{D} \to \mathbb{R}^n$. In the case of DAEs, the initial values have to satisfy consistency conditions and typically depend on the physical parameters of the system.

An input $\mathbf{u} : [t_0, T] \to \mathbb{R}^{n_{\text{in}}}$ is supplied to the system (10.1). An output $\mathbf{y} : [t_0, T] \times \mathcal{D} \to \mathbb{R}^{n_{\text{out}}}$ is defined as a QoI by the state variables or inner variables and the matrix C. Without loss of generality, we restrict the analysis to the case of a single output, i. e., $n_{\text{out}} = 1$.

Efficient methods of MOR are available for linear time-invariant dynamical systems of the form

$$\begin{aligned} E(\boldsymbol{\mu})\dot{\mathbf{x}}(t,\boldsymbol{\mu}) &= A(\boldsymbol{\mu})\mathbf{x}(t,\boldsymbol{\mu}) + B(\boldsymbol{\mu})\mathbf{u}(t), \\ y(t,\boldsymbol{\mu}) &= C(\boldsymbol{\mu})\mathbf{x}(t,\boldsymbol{\mu}). \end{aligned} \tag{10.3}$$

Typical MOR methods are balanced truncation, as described in Chapter 2 of Volume 1 of *Model order reduction*, and Krylov subspace methods, as described in Chapter 3 of Volume 1 of *Model order reduction*. Thus we focus on linear dynamical systems (10.3). We assume that the linear dynamical system (10.3) is asymptotically stable for all $\boldsymbol{\mu} \in \mathcal{D}$. Hence each eigenvalue λ satisfying $\det(\lambda E(\boldsymbol{\mu}) - A(\boldsymbol{\mu})) = 0$ has a strictly negative real part.

In MOR, a dynamical system of a much lower dimension $r \ll n$ is constructed, whose output $\bar{y}$ is still a good approximation of the QoI y in the original system (10.1)

or (10.3). Let a fixed parameter value $\boldsymbol{\mu} \in \mathcal{D}$ be given. The ROM of the linear dynamical system (10.3) reads as

$$\begin{aligned} \bar{E}(\boldsymbol{\mu})\dot{\bar{\mathbf{x}}}(t,\boldsymbol{\mu}) &= \bar{A}(\boldsymbol{\mu})\mathbf{x}(t,\boldsymbol{\mu}) + \bar{B}(\boldsymbol{\mu})\mathbf{u}(t), \\ \bar{y}(t,\boldsymbol{\mu}) &= \bar{C}(\boldsymbol{\mu})\bar{\mathbf{x}}(t,\boldsymbol{\mu}). \end{aligned} \tag{10.4}$$

Projection-based MOR applies two matrices $V, W \in \mathbb{R}^{n\times r}$ of full rank. Typically, an orthogonal matrix V is supposed, i. e., $V^\top V = I_r$ with identity matrix $I_r \in \mathbb{R}^{r\times r}$. The matrices of the linear dynamical system are reduced by

$$\begin{aligned} \bar{A}(\boldsymbol{\mu}) &= W^\top A(\boldsymbol{\mu})V, & \bar{B}(\boldsymbol{\mu}) &= W^\top B(\boldsymbol{\mu}), \\ \bar{C}(\boldsymbol{\mu}) &= C(\boldsymbol{\mu})V, & \bar{E}(\boldsymbol{\mu}) &= W^\top E(\boldsymbol{\mu})V. \end{aligned} \tag{10.5}$$

If the two projection matrices coincide ($V = W$), then the MOR method is of Galerkin type. In common MOR, the projection matrices are computed for each required parameter value $\boldsymbol{\mu} \in \mathcal{D}$ separately. pMOR addresses the parameter variation in a whole domain $\mathcal{D}$, which is considered in Section 10.4.1.

MOR of general nonlinear dynamical systems (10.1) represents a critical task. More efficient MOR methods are available for quadratic-bilinear (QB) systems; see [1, 4] and Chapter 3 of Volume 1 of *Model order reduction*. Sometimes a nonlinear dynamical system can be transformed into an equivalent QB system. Alternatively, the construction of approximative QB systems is feasible.

Furthermore, an overview on software of MOR methods can be found in Chapter 13 of this volume.

10.2.2 Stochastic modeling

The parameters are often affected by uncertainties in the systems (10.1) or (10.3). A common approach in UQ is to consider the parameters as independent random variables $\boldsymbol{\mu} : \Omega \to \mathcal{D}$ on some probability space $(\Omega, \mathcal{A}, P)$ with event space Ω, sigma-algebra $\mathcal{A}$, and probability measure P. Often traditional probability distributions are applied like uniform, beta, Gaussian, etc. We assume that a joint probability density function $\rho : \mathcal{D} \to \mathbb{R}$ is available. Consequently, the output becomes a random process. The expected value of a measurable function $f : \mathcal{D} \to \mathbb{R}$ depending on the random variables reads as

$$\mathbb{E}[f] = \int_\Omega f\big(\boldsymbol{\mu}(\omega)\big)\,\mathrm{d}P(\omega) = \int_{\mathcal{D}} f(\boldsymbol{\mu})\rho(\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\mu}, \tag{10.6}$$

provided that the integral exists. The moments are $\mathbb{E}[f^k]$ for positive integers k. The variance is the second central moment

$$\mathrm{Var}[f] = \mathbb{E}[f^2] - \mathbb{E}[f]^2 \tag{10.7}$$

as usual and the standard deviation is its square root $\sigma[f] = \sqrt{\mathrm{Var}[f]}$. The skewness and the kurtosis also represent interesting statistical quantities. They are given by the third and fourth standardized moments, respectively, i. e.,

$$\eta_j = \frac{\mathbb{E}[(f - \mathbb{E}[f])^j]}{\sigma[f]^j} \tag{10.8}$$

for $j = 3, 4$.

In the dynamical systems (10.1) and (10.3), both the state variables and the output (QoI) change into random processes due to the stochastic modeling. Thus the complexity of the problem increases significantly.

10.2.3 Quadrature and sampling

In a stochastic model, the desired data typically represent probabilistic integrals of the form (10.6). For example, moments are defined by the powers of a function and failure probabilities are given by an indicator function or a characteristic function; see [19]. Since the QoI is the random process y, the integrands are time-dependent. Sometimes just the QoI at a final time $t = T$ is considered.

A well-known approach to discretize a probabilistic integral is a multivariate quadrature rule or a sampling method. Each technique is determined by a set of nodes $\{\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(k)}\} \subset \mathcal{D}$ in the parameter domain and a set of weights $\{\gamma_1, \dots, \gamma_k\} \subset \mathbb{R}$. The sum of the weights is always one. The numerical approximation of a probabilistic integral becomes a finite sum

$$\int_{\mathcal{D}} f(\boldsymbol{\mu})\rho(\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\mu} \approx \sum_{\ell=1}^{k} \gamma_\ell f(\boldsymbol{\mu}^{(\ell)}). \tag{10.9}$$

In the case of low numbers of random parameters (say, $p \leq 5$), we can use tensor product formulas of univariate quadrature rules. These methods become inefficient for higher dimensions due to the curse of dimensionality. Sparse grids or specific quadrature rules are available for large dimensions; see [17, 29, 43]. The curse of dimensionality is omitted in the sparse grid construction. A drawback is that often negative weights occur. However, also sparse grids become computationally infeasible for very high dimensions (say, $p > 30$). Consequently, we have to apply sampling schemes like Monte Carlo or quasi-Monte Carlo methods; see [28]. The weights become $\gamma_\ell = \frac{1}{k}$ for all ℓ in each sampling scheme. Pseudo-random numbers or sequences of low discrepancy yield the nodes in a Monte Carlo or quasi-Monte Carlo method, respectively. Any number k can be chosen in a sampling method. Yet high-dimensional problems require typically a large number k to achieve sufficiently accurate results.

Figure 10.1 illustrates examples of the nodes for different methods in the case of two independent uniformly distributed random variables $\mu_i \in [0, 1]$ for $i = 1, 2$.

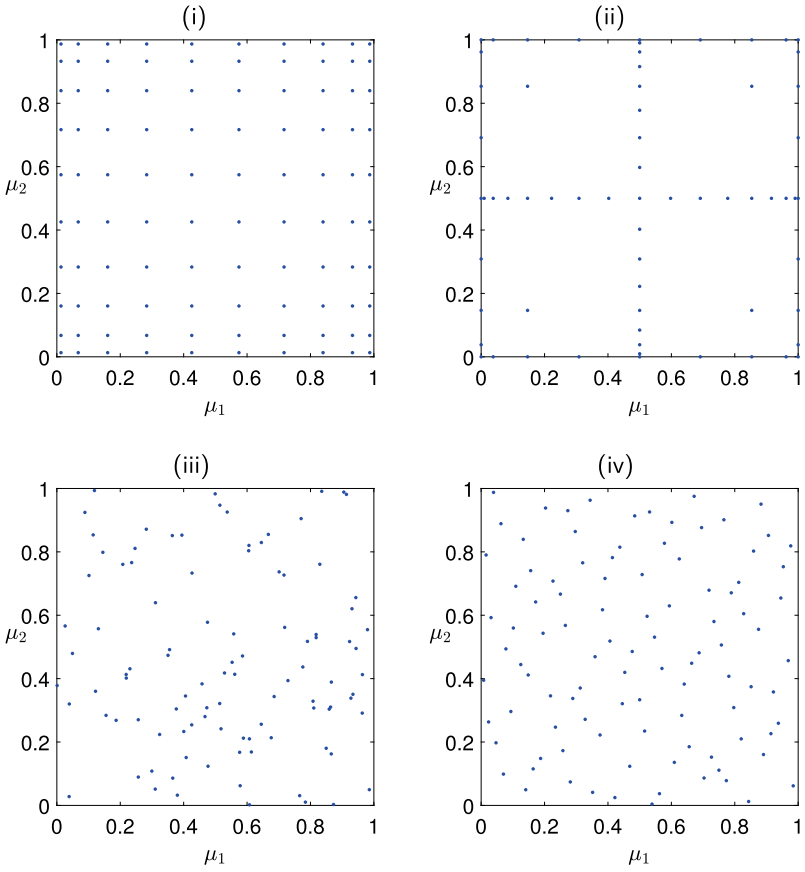


Figure 10.1: Nodes in quadrature rules or sampling methods for two uniformly distributed random variables: (i) tensor product Gauss–Legendre quadrature (100 points), (ii) sparse grid of level 4 based on the Clenshaw–Curtis rule (65 points), (iii) Monte Carlo with pseudo-random numbers (100 points), (iv) quasi-Monte Carlo with Halton sequence (100 points).

Since we consider dynamical systems (10.1) or (10.3) with a QoI, a function f depends on y in the integrand of (10.6), i. e., $f(\boldsymbol{\mu}) = \tilde{f}(y(t,\boldsymbol{\mu}))$ for fixed t. Hence the evaluation of an approximation (10.9) requires to solve k times an initial value problem of the dynamical system. This effort dominates the computation work in the stochastic model. The total effort is roughly proportional to k.

10.2.4 Polynomial expansions

The expected value (10.6) implies the inner product

$$\langle f, g \rangle = \mathbb{E}[fg] = \int_{\mathcal{D}} f(\boldsymbol{\mu}) g(\boldsymbol{\mu}) \rho(\boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\mu} \tag{10.10}$$

for two measurable functions depending on the random parameters. The associated Hilbert space is the set of square integrable functions

$$\mathcal{L}^2(\mathcal{D},\rho) = \{f : \mathcal{D} \to \mathbb{R} : f \text{ measurable and } \mathbb{E}[f^2] < \infty\}. \tag{10.11}$$

Its norm reads as $\|f\|_{\mathcal{L}^2} = \sqrt{\langle f, f\rangle}$.

We apply an expansion of the random process y into a set of orthogonal polynomials. Each traditional probability distribution exhibits a sequence of orthogonal polynomials (see [45]): Legendre polynomials for uniform distribution, Hermite polynomials for Gaussian distribution, Jacobi polynomials for beta distribution, etc. Let $(\phi_j^{(q)}(\mu_q))_{j\in\mathbb{N}_0}$ be the sequence of univariate orthonormal polynomials associated to the q-th random variable. The degree of the j-th polynomial is exactly j. We assume that the orthonormal basis is complete, which holds true for Gaussian, uniform, beta, and other distributions. However, there are exceptions; see [12]. The multivariate polynomials are just the products of the univariate polynomials. The set of all basis polynomials up to total degree d reads as

$$\{\Phi_i(\boldsymbol{\mu}) = \phi_{j_1}^{(1)}(\mu_1)\phi_{j_2}^{(2)}(\mu_2)\cdots\phi_{j_p}^{(p)}(\mu_p) : j_1 + j_2 + \cdots + j_p \le d\}. \tag{10.12}$$

There is a one-to-one mapping between the indices $i = 1, 2, 3, \ldots$ and the multiindices $(j_1, j_2, \ldots, j_p)$. The basis polynomials $(\Phi_i)_{i\in\mathbb{N}}$ satisfy the orthogonality property

$$\langle \Phi_i, \Phi_k\rangle = \begin{cases} 0 & \text{for } i \neq k, \\ 1 & \text{for } i = k, \end{cases}$$

with the inner product (10.10). The cardinality of the set (10.12) is (see [46])

$$\binom{p+d}{d} = \frac{(p+d)!}{p!d!}. \tag{10.13}$$

Hence the number of basis polynomials becomes large for large numbers p of random variables, even if the total degree is moderate, say, $2 \le d \le 5$. Figure 10.2 illustrates the growth of the number of multivariate basis polynomials.

Series including the orthogonal basis functions are called polynomial chaos (PC) expansions. The technique is analogous to Fourier series, where the trigonometric polynomials are just replaced by the orthonormal polynomials (10.12) with respect to the inner product (10.10). We expand the QoI satisfying the linear dynamical system (10.3) into

$$y(t,\boldsymbol{\mu}) = \sum_{i=1}^{\infty} w_i(t)\Phi_i(\boldsymbol{\mu}), \tag{10.14}$$

with a priori unknown coefficient functions $w_i : [t_0, T] \to \mathbb{R}$ satisfying

$$w_i(t) = \langle y(t,\boldsymbol{\mu}), \Phi_i(\boldsymbol{\mu})\rangle = \int_{\mathcal{D}} y(t,\boldsymbol{\mu})\Phi_i(\boldsymbol{\mu})\rho(\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\mu} \tag{10.15}$$

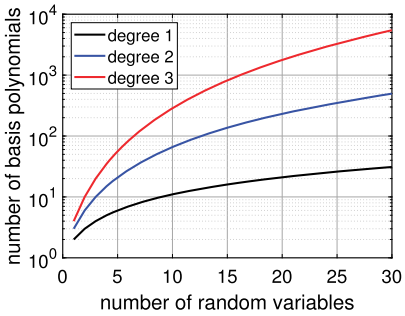


Figure 10.2: Number of basis polynomials in dependence on number of random variables for different total degrees (in semi-logarithmic scale).

for $i \in \mathbb{N}$. We obtain a finite approximation by a truncation of the series (10.14), i. e.,

$$y^{(m)}(t, \boldsymbol{\mu}) = \sum_{i=1}^{m} w_i(t)\Phi_i(\boldsymbol{\mu}). \tag{10.16}$$

The approximations (10.16) converge pointwise in time to the random process y for $m \to \infty$ in the $\mathcal{L}^2$-norm provided that the basis is complete.

We include all basis polynomials (10.12) up to some total degree d, where the number m is equal to (10.13). Let $\Phi_1 \equiv 1$ be the constant polynomial of degree zero. The approximation (10.16) also yields the expected value as well as an approximation of the variance (10.7) via

$$\mathbb{E}[y(t,\cdot)] = w_1(t) \quad \text{and} \quad \mathrm{Var}[y(t,\cdot)] \approx \sum_{i=2}^{m} w_i(t)^2. \tag{10.17}$$

The task is to compute numerically the coefficient functions of the truncated series (10.16) in this approach.

10.3 MOR for stochastic expansions

We demonstrate the potential to apply MOR methods for the numerical computation of the unknown coefficient functions in the PC expansions introduced in Section 10.2.4.

10.3.1 Stochastic Galerkin method

Let $\mathbf{v} = (\mathbf{v}_1^\top, \ldots, \mathbf{v}_m^\top)^\top \in \mathbb{R}^{mn}$ and $\mathbf{w} = (w_1, \ldots, w_m)^\top \in \mathbb{R}^m$. The random-dependent system (10.3) changes into a larger coupled linear dynamical system

$$\begin{aligned} \hat{E}\dot{\mathbf{v}}(t) &= \hat{A}\mathbf{v}(t) + \hat{B}\mathbf{u}(t), \\ \mathbf{w}(t) &= \hat{C}\mathbf{v}(t), \end{aligned} \tag{10.18}$$

with constant matrices $\hat{A}, \hat{E} \in \mathbb{R}^{mn \times mn}$, $\hat{B} \in \mathbb{R}^{mn \times n_{\text{in}}}$, and $\hat{C} \in \mathbb{R}^{m \times mn}$. Initial values $\mathbf{v}(t_0) = \mathbf{v}_0$ follow from a truncated PC expansion of the initial condition (10.2). The stochastic Galerkin system (10.18) always features multiple outputs even if the original system (10.3) has a single output. The number of inputs remains the same.

To define the matrices in the coupled system, we introduce an auxiliary matrix and a column vector by

$$S(\boldsymbol{\mu}) = \left(\Phi_i(\boldsymbol{\mu})\Phi_j(\boldsymbol{\mu})\right)_{i,j=1,\dots,m} \in \mathbb{R}^{m \times m} \quad \text{and} \quad \mathbf{s}(\boldsymbol{\mu}) = \left(\Phi_i(\boldsymbol{\mu})\right)_{i=1,\dots,m} \in \mathbb{R}^{m}.$$

The matrices follow from the original matrices in (10.3) by probabilistic integrals

$$\hat{A} = \mathbb{E}[S \otimes A], \quad \hat{B} = \mathbb{E}[\mathbf{s} \otimes B], \quad \hat{C} = \mathbb{E}[S \otimes C], \quad \hat{E} = \mathbb{E}[S \otimes E], \tag{10.19}$$

using the Kronecker product $\otimes$, where the expected values (10.6) are calculated componentwise. If the matrices of the system (10.3) represent polynomials of the random variables, then the expected values can often be calculated analytically. Otherwise, numerical quadrature schemes are required to calculate the matrices once.

The linear Galerkin system (10.18) may be unstable, even though the systems (10.3) are asymptotically stable for (strictly) all realizations of the random variables. However, such a loss of stability hardly occurs within problems from the applications. Examples of stability loss are just academic; cf. [37]. Thus we assume that the stochastic Galerkin system (10.18) is asymptotically stable. More details on the stochastic Galerkin method for linear dynamical systems can be found in [31, 32], for example.

The dimension of the stochastic Galerkin system (10.18) is mn. This dimensionality becomes huge for large numbers m given by (10.13). Thus the linear stochastic Galerkin system represents an excellent candidate for an MOR. Projection-based MOR operates on the constant matrices $\hat{A}, \hat{B}, \hat{C}, \hat{E}$ like in (10.5). Krylov subspace methods were successfully applied in [23, 47]. Balanced truncation was used in [15, 33, 35]. The reduction is often efficient such that reduced dimensions $r < m$ are still sufficiently accurate, i. e., the state space dimension is lower than the number of outputs.

10.3.2 Stochastic collocation techniques

If the stochastic model is solved approximately using the solutions of the dynamical system (10.3) for a finite number of realizations of the random parameters, then the approach is called a stochastic collocation method. In this context, we use a quadrature rule or a sampling scheme introduced in Section 10.2.3 to compute the unknown coefficient functions (10.15) of the PC expansion.

The original dynamical systems (10.3) may be small- or medium-sized. To make the stochastic collocation method accessible to MOR, we construct a large auxiliary system; see [33, 34]. A construction of this type was also applied to Itô differential

equations for another purpose in [27]. Given the nodes of a quadrature rule or sampling scheme, the initial value problems

$$\begin{aligned} E(\boldsymbol{\mu}^{(\ell)})\dot{\mathbf{x}}(t,\boldsymbol{\mu}^{(\ell)}) &= A(\boldsymbol{\mu}^{(\ell)})\mathbf{x}(t,\boldsymbol{\mu}^{(\ell)}) + B(\boldsymbol{\mu}^{(\ell)})\mathbf{u}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0(\boldsymbol{\mu}^{(\ell)}), \\ y(t,\boldsymbol{\mu}^{(\ell)}) &= C(\boldsymbol{\mu}^{(\ell)})\mathbf{x}(t,\boldsymbol{\mu}^{(\ell)}) \end{aligned} \tag{10.20}$$

are solved separately for $\ell = 1,\ldots,k$. The integrals in (10.15) change into the finite sums

$$w_i(t) = \sum_{\ell=1}^{k} \gamma_\ell \Phi_i(\boldsymbol{\mu}^{(\ell)}) y(t,\boldsymbol{\mu}^{(\ell)}) = \sum_{\ell=1}^{k} \gamma_\ell \Phi_i(\boldsymbol{\mu}^{(\ell)}) C(\boldsymbol{\mu}^{(\ell)})\mathbf{x}(t,\boldsymbol{\mu}^{(\ell)}) \tag{10.21}$$

for $i = 1,2,\ldots,m$.

The systems (10.20) are collected in a single system as done in [33]. Let

$$\begin{aligned} \hat{\mathbf{x}}(t) &:= \left(\mathbf{x}(t,\boldsymbol{\mu}^{(1)})^\top,\ldots,\mathbf{x}(t,\boldsymbol{\mu}^{(k)})^\top\right)^\top \in \mathbb{R}^{kn} \quad \text{and} \\ \mathbf{w}(t) &:= \left(w_1(t),\ldots,w_m(t)\right)^\top \in \mathbb{R}^m. \end{aligned}$$

The systems (10.20) for $\ell = 1,\ldots,k$ together with the outputs (10.21) for $i = 1,\ldots,m$ yield the larger weakly coupled system of the form (10.18). This system consists of k separate subsystems (10.20), which are coupled only by supplying of the same input and the definition of the outputs (10.21). Thus the matrices $\hat{A},\hat{E} \in \mathbb{R}^{kn\times kn}$ are block-diagonal. More precisely, we have

$$\hat{G} = \begin{pmatrix} G(\boldsymbol{\mu}^{(1)}) & & \\ & \ddots & \\ & & G(\boldsymbol{\mu}^{(k)}) \end{pmatrix} \quad \text{for } G \in \{A,E\} \text{ and } \quad \hat{B} = \begin{pmatrix} B(\boldsymbol{\mu}^{(1)}) \\ \vdots \\ B(\boldsymbol{\mu}^{(k)}) \end{pmatrix}.$$

Obviously, the weakly coupled system is asymptotically stable provided that the original systems (10.3) are asymptotically stable for all $\boldsymbol{\mu} \in \mathcal{D}$. Likewise, we define an auxiliary matrix

$$\widetilde{C} = \begin{pmatrix} C(\boldsymbol{\mu}^{(1)}) & & \\ & \ddots & \\ & & C(\boldsymbol{\mu}^{(k)}) \end{pmatrix} \in \mathbb{R}^{k\times kn}.$$

The quadrature rule (10.21) determines the output matrix $\hat{C} \in \mathbb{R}^{m\times kn}$ by

$$\hat{C} = F\widetilde{C} \quad \text{with } F = (f_{i\ell}) \in \mathbb{R}^{m\times k}, \quad f_{i\ell} = \gamma_\ell \Phi_i(\boldsymbol{\mu}^{(\ell)}).$$

Again the outputs $\mathbf{w}$ of (10.18) yield an approximation (10.16) of the QoI. For large numbers k of nodes or samples, the dimension kn becomes huge. Now we can apply methods of MOR to the weakly coupled system.

10.4 Parametric MOR for quadrature and sampling

We show a potential to compute statistics, where ROMs from pMOR are sampled instead of the full-order models (FOMs). Hence the ROM is used as a surrogate model. This approach is applicable to both linear dynamical systems (10.3) and nonlinear dynamical systems (10.1). The previous works [5, 8] represent surveys on pMOR. In [38], UQ and pMOR have already been combined in the case of linear dynamical systems. A specific pMOR method is presented for general dynamical systems in Chapter 7 of Volume 1 of *Model order reduction*.

10.4.1 Application of pMOR

Now the original parameter-dependent dynamical system is assumed to be high-dimensional. The aim of pMOR is to preserve the parameters in a reduction of the systems (10.1) or (10.3). Thus the ROMs are constructed in dependence on the parameters within an offline phase, where the computation work is significant. Whenever an ROM is required for a particular realization of the parameters, a cheap formula is available within an online phase.

In projection-based pMOR, there are mainly two possibilities to determine the projection matrices:

1. *Parameter-dependent projections*: A priori calculations yield formulas for the projection matrices $V(\boldsymbol{\mu}), W(\boldsymbol{\mu}) \in \mathbb{R}^{n\times r}$, which can be evaluated for any $\boldsymbol{\mu} \in \mathcal{D}$. The reduced matrices become

$$\begin{aligned} \bar{A}(\boldsymbol{\mu}) &= W(\boldsymbol{\mu})^\top A(\boldsymbol{\mu})V(\boldsymbol{\mu}), & \bar{B}(\boldsymbol{\mu}) &= W(\boldsymbol{\mu})^\top B(\boldsymbol{\mu}), \\ \bar{C}(\boldsymbol{\mu}) &= C(\boldsymbol{\mu})V(\boldsymbol{\mu}), & \bar{E}(\boldsymbol{\mu}) &= W(\boldsymbol{\mu})^\top E(\boldsymbol{\mu})V(\boldsymbol{\mu}), \end{aligned} \tag{10.22}$$

   for varying parameters $\boldsymbol{\mu} \in \mathcal{D}$. For example, a local reduced basis is computed for each element in a predetermined finite set of parameters. If the projection matrices are required for a particular $\boldsymbol{\mu} \in \mathcal{D}$, then a kind of interpolation produces $V(\boldsymbol{\mu})$ and $W(\boldsymbol{\mu})$ using neighboring local bases.
2. *Constant projections*: The information of the whole parameter domain $\mathcal{D}$ or a large finite set of samples is used to construct constant projection matrices V_0, W_0. Thus we have

$$V(\boldsymbol{\mu}) = V_0 \quad \text{and} \quad W(\boldsymbol{\mu}) = W_0 \quad \text{for all } \boldsymbol{\mu} \in \mathcal{D} \tag{10.23}$$

   in (10.22). This approach yields global projection matrices, which can be used for any realization of the parameters. However, a larger reduced dimension is often necessary for a sufficiently accurate MOR in comparison to a local construction.

In both cases, the two projection matrices are often identically selected ($W = V$) and thus just one projection matrix has to be identified.

In the two variants of projection-based pMOR, the crucial computation work takes place in an offline phase. Often this offline phase consists in evaluations of the FOM for a finite set of parameter values. Such evaluations could also be used in a quadrature method applied to the FOM without an MOR. Thus the critical issue is if the pMOR is able to identify a sufficiently accurate global ROM still with a low computational effort. Otherwise, the FOM could be sampled as well. This problem of certified accuracy in pMOR has also been recognized by [11]. If the error of the pMOR does not exceed the magnitude of the error in the time integration, then the described approach is reasonable.

A stochastic Galerkin method can be applied to a parameter-dependent ROM; see [38]. This approach features both advantages and disadvantages in comparison to the strategy from Section 10.3.1. Alternatively, we apply the approach of quadrature or sampling to the ROMs in this section.

10.4.2 Computation of global projection matrix

We demonstrate a technique to determine a global projection matrix from a set of local projection matrices. Alternative strategies can be found in [5, 14, 41]. Let parameter values $\{\boldsymbol{\mu}^{(1)}, \ldots, \boldsymbol{\mu}^{(s)}\} \subset \mathcal{D}$ be given, which should generate a good representation of all parameters within $\mathcal{D}$. We determine local projection matrices $V(\boldsymbol{\mu}^{(j)}) \in \mathbb{R}^{n \times r_j}$ for $j = 1, \ldots, s$ by some MOR technique applied to the dynamical system (10.1) or (10.3). These local projection matrices are not required to be orthogonal. We collect all local bases in a large matrix

$$\hat{V} = \left(V(\boldsymbol{\mu}^{(1)})\ V(\boldsymbol{\mu}^{(2)}) \cdots V(\boldsymbol{\mu}^{(s)})\right) \in \mathbb{R}^{n \times \hat{r}} \tag{10.24}$$

with $\hat{r} = r_1 + r_2 + \cdots + r_s$ columns, assuming $\hat{r} \ll n$. In [20], just an orthogonalization of a matrix like (10.24) is applied to define the global projection matrix.

We decrease the dimension $\hat{r}$ of the global basis further by an approach also used in [38]. Moreover, this technique removes a (numerical) rank deficiency in the matrix (10.24) if so. The singular value decomposition (SVD) of the matrix (10.24) reads as

$$\hat{V} = USQ^{\top}, \tag{10.25}$$

with orthogonal matrices $U \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{\hat{r} \times \hat{r}}$ and a diagonal matrix $S \in \mathbb{R}^{n \times \hat{r}}$ including the nonnegative singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\hat{r}}$. In our application, just the first $\hat{r}$ singular values and their singular vectors have to be computed, which makes the SVD cheap. The singular vectors are the columns $\mathbf{u}_1, \ldots, \mathbf{u}_{\hat{r}}$ of the matrix U. Depending on the decay of the singular values, the r dominant singular vectors are entered in the global basis ($r < \hat{r}$)

$$V_0 = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r) \in \mathbb{R}^{n \times r}. \tag{10.26}$$

Thus the global matrix V is already orthogonal ($V_0^\top V_0 = I_r$). Furthermore, we simply employ the Galerkin-type choice $W_0 = V_0$ to define the second projection matrix in (10.23).

If there were no parameter variations in the dynamical system, then the local projection matrices would be identical provided that the same MOR scheme is used. In (10.24), it follows that $V(\boldsymbol{\mu}^{(k)}) = V(\boldsymbol{\mu}^{(0)}) \in \mathbb{R}^{n\times r_0}$ for all $k = 1,\dots,s$ with any reference parameter $\boldsymbol{\mu}^{(0)} \in \mathcal{D}$. Now let this constant projection matrix be orthogonal. Consequently, the extended matrix (10.24) owns the singular values

$$\sigma_i = \begin{cases} \sqrt{s} & \text{for } i = 1,\dots,r_0, \\ 0 & \text{for } i = r_0+1,\dots,sr_0. \end{cases} \tag{10.27}$$

If a low amount of parameter variation is given in $\mathcal{D}$, then the singular values of (10.24) will be close to the trivial instance (10.27). Thus the deviation of the singular values from the case (10.27) provides a measure of the sensitivity of the problem with respect to the parameter variation.

10.4.3 Reduced basis methods

The class of reduced basis methods represents a type of pMOR. In particular, this approach is efficient in the case of stationary solutions of PDEs; cf. [18, 22]. For example, weak formulations of elliptic equations can be tackled.

We consider a general problem

$$L(\mathbf{x}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0 \tag{10.28}$$

defined by a (differential) operator $L : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$ on a function space $\mathcal{X}$ with norm $\|\cdot\|_{\mathcal{X}}$. Let a unique solution $\mathbf{x} \in \mathcal{X}$ exist for each $\boldsymbol{\mu} \in \mathcal{D}$. The operator L may be a weak formulation of a PDE. Alternatively, L can identify a time-dependent solution of ODEs like a periodic steady-state response, for example. Moreover, a (spatial) discretization of PDEs yields operators whose solutions represent high-dimensional approximations to the exact solutions of an underlying problem. Typical spatial discretizations are finite element methods and finite difference schemes.

Now we assume that a solution of (10.28) has to be computed many times for different realizations of the parameters. Thus we want to use a surrogate model that generates cheap approximations. In reduced basis methods, a relatively small set of linearly independent solutions is identified, which form the subspace

$$\mathcal{X}_r = \operatorname{span}\{\mathbf{x}(\boldsymbol{\mu}^{(1)}), \dots, \mathbf{x}(\boldsymbol{\mu}^{(r)})\}. \tag{10.29}$$

Given an arbitrary parameter value $\boldsymbol{\mu}$, the associated solution of (10.28) is approximated by a linear combination

$$\mathbf{x}(\boldsymbol{\mu}) \approx \sum_{j=1}^{r} \alpha_j(\boldsymbol{\mu}) \mathbf{x}(\boldsymbol{\mu}^{(j)}), \tag{10.30}$$

where the real coefficients $\alpha_1, \ldots, \alpha_r$ have to be determined in dependence on the parameter value. Consequently, the operator (10.28) is modified into an approximation

$$L_r(\mathbf{x}_r^*(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0 \tag{10.31}$$

with $L_r : \mathcal{X}_r \times \mathcal{D} \to \mathbb{R}$, which identifies an approximation in the subspace $\mathcal{X}_r$ and thus the required coefficients of (10.30).

Two tasks have to be accomplished in the reduced basis approach:

1. determination of the basis functions in (10.29),
2. construction of the reduced-order operator (10.31) and its efficient numerical solution.

The first task is typically achieved by a greedy algorithm. We approximate the parameter domain $\mathcal{D}$ by a large finite set of samples $\mathcal{D}_{\text{train}} \subset \mathcal{D}$. Let an initial solution $\mathbf{x}(\boldsymbol{\mu}^{(1)})$ be given. We compute the subspaces (10.29) recursively via $\mathcal{X}_{j+1} = \mathcal{X}_j \cup \operatorname{span}\{\mathbf{x}(\boldsymbol{\mu}^{j+1})\}$ with

$$\boldsymbol{\mu}^{(j+1)} = \operatorname*{arg\,max}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{train}}} \left\| \mathbf{x}(\boldsymbol{\mu}) - \mathbf{x}_j^*(\boldsymbol{\mu}) \right\|_{\mathcal{X}}, \tag{10.32}$$

including the solutions of the operators (10.28) and (10.31). However, the computation of the solutions and the norm of their difference is often too costly in (10.32). Hence we replace the norm by a residual-based estimator R. The bounds

$$c_1 R(\boldsymbol{\mu}) \leq \left\| \mathbf{x}(\boldsymbol{\mu}) - \mathbf{x}_j^*(\boldsymbol{\mu}) \right\|_{\mathcal{X}} \leq c_2 R(\boldsymbol{\mu}) \quad \text{for } \boldsymbol{\mu} \in \mathcal{D}$$

with constants $c_1, c_2 > 0$ are required for a certified error estimation. The computational effort of an evaluation of the residual-based criterion is low.

The second task consists in the derivation of the approximation (10.31) to the operator (10.28). This construction is problem-dependent. In weak formulations of PDEs, the original function space is just restricted to the low-dimensional subspace (10.29). In the case of linear operators, we compute the involved matrices a priori in the offline phase. The solution of (10.31) becomes cheap in the online phase now. In the case of nonlinear operators, a straightforward approximation still includes the complete nonlinear terms of (10.28). Thus we require cheap approximations of the nonlinearities. The (discrete) empirical interpolation method represents such an approximate construction; see [10] and the references therein.

The efficiency of reduced basis methods can be motivated by the manifold of the parametric solutions

$$\mathcal{M} = \{\mathbf{x}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{D}\} \subset \mathcal{X}. \tag{10.33}$$

If the dependence of the PDE solutions on the parameters is (sufficiently) smooth, then the Kolmogorov width of the manifold is small. Consequently, a sufficiently accurate approximation is possible by a low-dimensional subspace.

Stochastic reduced basis methods were already investigated in [26, 39]. The use of reduced basis methods for UQ of weak formulations of PDEs was presented in detail by [11]. The reduced basis approach was proven to be efficient for stationary problems in a spatial domain. Also parabolic equations which depend on time as well as space can be treated by these methods; see [11, 30]. However, the applicability to transient problems like our dynamical systems (10.1) and (10.3) still has to be examined. We think about dynamical systems which do not result from a spatial discretization of a PDE. In this case, the function space $\mathcal{X}$ may represent the periodic steady-state response in a time interval $[0, T]$, since periodic solutions satisfy a boundary value problem. The efficiency is still undecided for initial value problems. If just the solution's value in $\mathbb{R}^n$ at a final time $t = T$ represents the QoI, then reduced basis methods are unnecessarily complex for this task.

10.5 Numerical examples

We demonstrate the application of the approaches from Section 10.3 and Section 10.4 now.

10.5.1 MOR for stochastic expansions

Figure 10.3 depicts the diagram of a band pass filter. The mathematical modeling yields an explicit system of ODEs with dimension $n = 10$ for five node voltages and five branch currents. Physical parameters are included by five capacitances, five inductances, and four resistances ($p = 14$). A single input voltage is supplied, whereas a single output voltage drops at a load resistance. The Bode plot of the linear dynamical system is shown for a constant choice of the parameters by Figure 10.4. We recognize that there is just a small frequency interval around $\omega = 1$, where the magnitude of oscillations remains the same, while other frequencies are damped strongly.

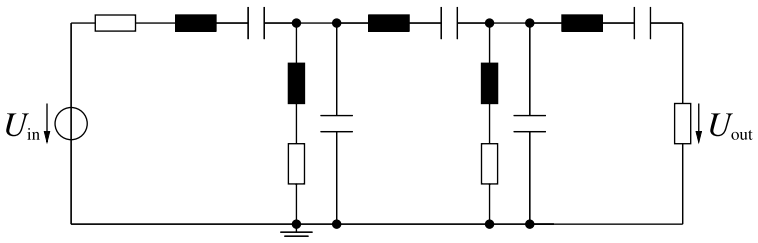


Figure 10.3: Electric circuit of a band pass filter.

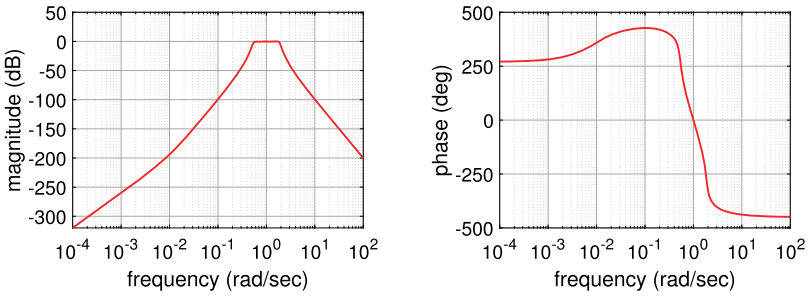


Figure 10.4: Bode plot of band pass filter for deterministic parameters.

We replace all physical parameters by independent random variables with uniform probability distributions, which vary 20 % around their mean values given by the constant choice of parameters from above. Hence the PC expansion (10.14) involves the multivariate Legendre polynomials. In the truncated PC expansion (10.16), we include all basis polynomials up to total degree $d = 3$, which implies $m = 680$ basis functions due to (10.13).

Now the two approaches from Section 10.3 are examined. On the one hand, we arrange the stochastic Galerkin system of dimension $mn = 6800$ as in Section 10.3.1. The system matrices (10.19) are computed by a sparse grid quadrature of level 3 based on the Clenshaw–Curtis rule, where $k = 4117$ nodes arise. On the other hand, we generate a stochastic collocation system as in Section 10.3.2, with a sparse grid quadrature of level 2 of the same type with $k = 421$ nodes. The dimension of this weakly coupled system is $kn = 4210$. Both systems feature a single input and m outputs. The outputs reproduce the expected value as well as the variance of the output voltage via (10.17).

We perform an MOR of both linear dynamical systems using the balanced truncation technique; see [2]. A direct linear algebra algorithm yields the Cholesky factors of the Gramian matrices. An SVD produces the Hankel singular values in each approach, which are shown in Figure 10.5. The singular values exhibit a similar rate of decay in both linear dynamical systems. We expect a high potential for an MOR due to the fast decay. The singular values and singular vectors allow for the construction of ROMs with any dimension.

We also perform a transient simulation to compare the accuracy of the stochastic expansion methods and their ROMs. The input voltage is chosen as the chirp signal

$$u(t) = \sin(2\pi t^2).$$

The output voltage represents the random QoI. The time interval $[0, 100]$ is considered with initial values (10.2) identical to zero. In the time integration, we use the trapezoidal rule with constant step size $\Delta t = 0.005$ in all cases. A reference solution of the expected value as well as the variance is computed using directly the sparse grid quadrature of level 3 with $k = 4117$ nodes (without projection to a PC expansion).

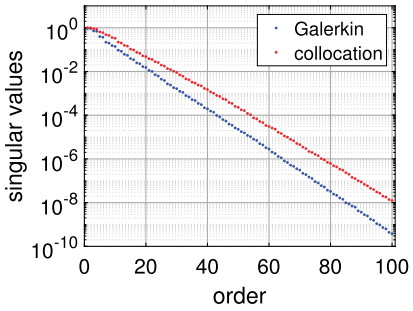


Figure 10.5: Dominating Hankel singular values of linear dynamical systems from the stochastic Galerkin method and the stochastic collocation technique in the band pass filter example.

Hence the error of the time integration becomes negligible in comparison to the error of the stochastic discretizations and the error of an MOR for moderate reduced dimensions. Figure 10.6 shows the approximations for the expected value as well as the standard deviation in the FOMs. The approximations of the expected value coincide in all techniques. The approximations of the standard deviation agree for the stochastic Galerkin method. The stochastic collocation yields a slightly different approximation, which still captures the main dynamics. Now we consider the ROMs in the stochastic Galerkin method and the stochastic collocation. Table 10.1 illustrates the differences of the expected value as well as the variance with respect to the FOM solution for varying reduced dimensions. Obviously, the differences diminish for increasing dimensions, which confirms the quality of the used MOR.

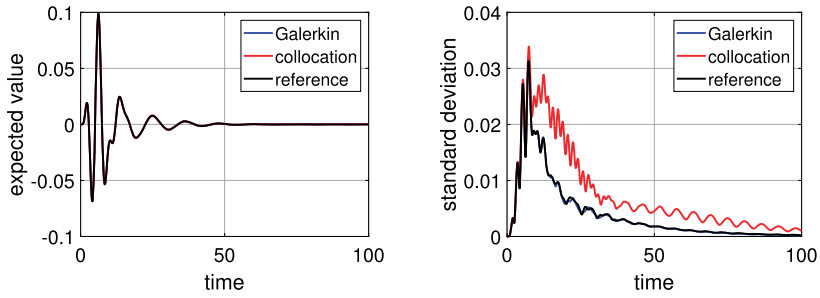


Figure 10.6: Transient simulation of expected value (left) and standard deviation (right) for random output voltage from the stochastic Galerkin approach, the stochastic collocation scheme, and reference solution.

In this example, we reproduced the expected value and the variance. Nevertheless, more sophisticated quantities can be derived from the PC approximation (10.16) using the coefficient functions.

Table 10.1: Maximum differences (rounded to one digit) in moments between FOM and ROM for the stochastic Galerkin method and the stochastic collocation technique.

Reduced dimension		10	20	30
Galerkin	Expected value	$8 \cdot 10^{-3}$	$7 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
	Variance	$1 \cdot 10^{-4}$	$2 \cdot 10^{-5}$	$1 \cdot 10^{-6}$
Collocation	Expected value	$2 \cdot 10^{-2}$	$3 \cdot 10^{-3}$	$4 \cdot 10^{-4}$
	Variance	$3 \cdot 10^{-4}$	$2 \cdot 10^{-5}$	$7 \cdot 10^{-6}$

10.5.2 PMOR for statistics

The anemometer system represents a benchmark in MOR; see [24, 25]. The convection-diffusion PDE

$$\rho_{\mathrm{fl}} c \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) - \rho_{\mathrm{fl}} c (\mathbf{v} \cdot \nabla T) + \dot{q}$$

models the time evolution of the temperature T with fluid density ρ_{fl}, thermal conductivity κ, specific heat c, and the velocity profile $\mathbf{v}$. The heat flow $\dot{q}$ becomes the input. The output is defined as the temperature difference between two sensors. We obtain a rough estimate of the flow velocity v (as a part of $\mathbf{v}$) by this difference. A finite element method performs a spatial discretization, which generates an implicit system of linear ODEs (10.3) with dimension $n = 29008$ and single-input-single-output. We arrange a constant fluid density $\rho_{\mathrm{fl}} = 1$. The system still depends on the three positive parameters $\mu_1 = c$, $\mu_2 = v$, $\mu_3 = \kappa$. Figure 10.7 depicts the Bode plot of the linear dynamical system in the case of deterministic parameters $\mu_i = 1$ for $i = 1, 2, 3$.

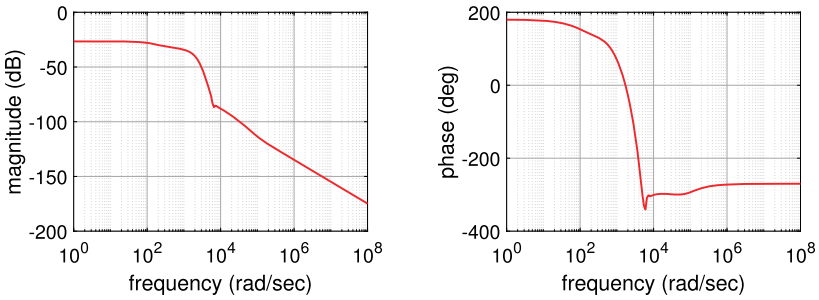


Figure 10.7: Bode plot of anemometer model for deterministic parameters.

In the stochastic modeling, we choose independent beta distributions for each parameter. Given a single beta-distributed random variable $\mu \in [-1, 1]$, the probability density function reads as

$$\rho(\mu) = C(\alpha, \beta)(1 - \mu)^{\alpha}(1 + \mu)^{\beta}, \tag{10.34}$$

with exponents $\alpha, \beta > -1$ and a constant $C(\alpha, \beta) > 0$ for normalization. A bijective linear transformation maps the interval $[-1, 1]$ to any interval $[\mu_{\min}, \mu_{\max}]$. We consider two choices of the parameter domain $\mathcal{D}$:

(i) small variation: $\mu_1 \in [0.9, 1.0]$, $\mu_2 \in [1.0, 1.1]$, $\mu_3 \in [1.0, 1.1]$,

(ii) large variation: $\mu_1 \in [0.5, 1.0]$, $\mu_2 \in [0.7, 1.2]$, $\mu_3 \in [1.0, 1.5]$.

Furthermore, we select the exponents $\alpha_i = 1$ and $\beta_i = 3$ for all $i = 1, 2, 3$.

In a pMOR, we use the technique from Section 10.4.2. We choose all vertices of the cube $\mathcal{D}$ for the computation of local reduced bases. Hence $s = 8$ parameter samples are involved. The one-sided Arnoldi method (see [2]) represents a moment-matching method, where a single expansion point $\theta \in \mathbb{C}$ is applied in the frequency domain. We employ the real expansion point $\theta = 10^4$ for each parameter sample, which causes real-valued results. A local orthonormal basis of dimension $r_j = 25$ is generated for each parameter sample $j = 1, \ldots, s$. The extended matrix (10.24) consists of $\hat{r} = 200$ columns. We compute its SVD (10.25) in the two cases (i) and (ii) of the parameter domains. The singular values are depicted in Figure 10.8. As expected, the singular values behave similar to the limit (10.27) of vanishing parameter dependence in the case (i) of small variations. The rate of decay becomes slower in the case (ii) of larger variations. Now a global reduced basis (10.26) can be arranged for any dimension $r \leq \hat{r}$, where the singular vectors associated to the dominant singular values are included.

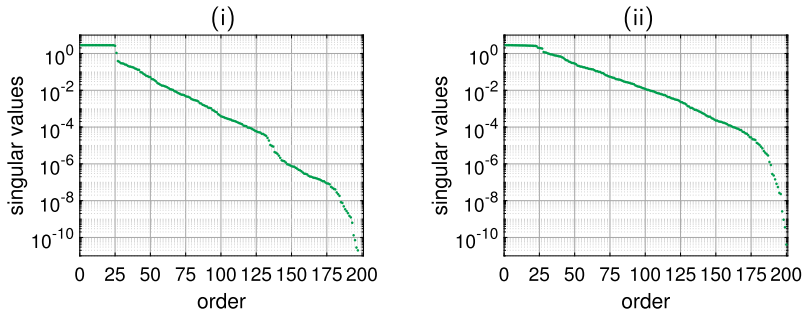


Figure 10.8: Singular values of extended matrix (10.24) in pMOR for the two choices (i) and (ii) of the parameter domain in the anemometer example.

We perform a transient simulation for a comparison of the FOM and the ROM in the case (ii). Using the time interval $[0, 0.01]$, the harmonic oscillation

$$u(t) = A \sin\left(\frac{2\pi}{T} t\right)$$

is supplied as input with period $T = 10^{-4}$ and amplitude $A = 10^4$. Initial values are identical to zero. The trapezoidal rule performs a time integration with constant step size $\Delta t = \frac{T}{20}$. Our aim is to compute statistics of the random process induced by

the single output. We use the Gauss–Jacobi quadrature on a tensor product grid with $k = 4^3 = 64$ nodes. Further tests indicate that this quadrature scheme is sufficiently accurate. In each node, the initial value problem is solved numerically for both the FOM and an ROM.

Firstly, we select the dimension $r = 75$ of the global basis in the ROM. Figure 10.9 illustrates the expected value as well as the standard deviation for both FOM and ROM. The approximation of the expected value is appropriate, whereas the standard deviation includes incorrect oscillations in the ROM. Secondly, we arrange the reduced dimension $r = 125$. Figure 10.10 shows the expected value and the standard deviation again. Now the solution from the ROM represents a good approximation to both statistical quantities. Furthermore, the skewness and the kurtosis (see (10.8)), which relate to the third moment and the fourth moment, respectively, are displayed in Figure 10.11. Although the approximations of the higher moments are less accurate in the MOR, the dynamics as well as the magnitude of these statistics are captured correctly.

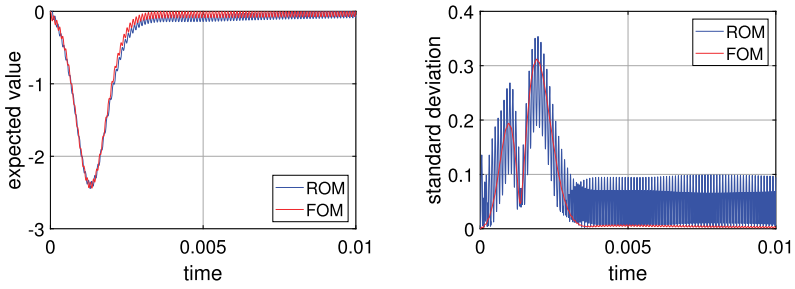


Figure 10.9: Expected value (left) and standard deviation (right) of random output in anemometer system obtained by ROM of dimension $r = 75$.

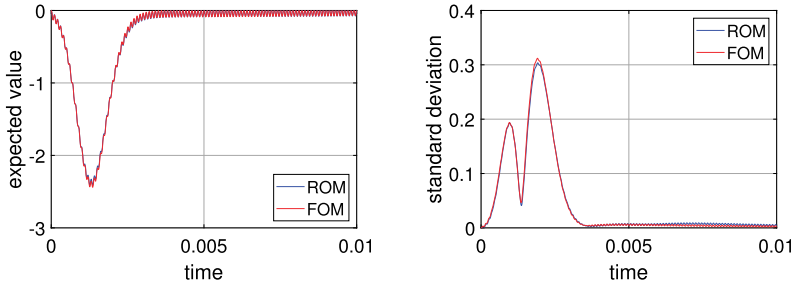


Figure 10.10: Expected value (left) and standard deviation (right) of random output in the anemometer system obtained by the ROM of dimension $r = 125$.

We note that a global projection matrix computed for a parameter domain $\mathcal{D}$ can be reused for any probability distribution in $\mathcal{D}$. For example, different exponents may be chosen in the beta distributions (10.34).

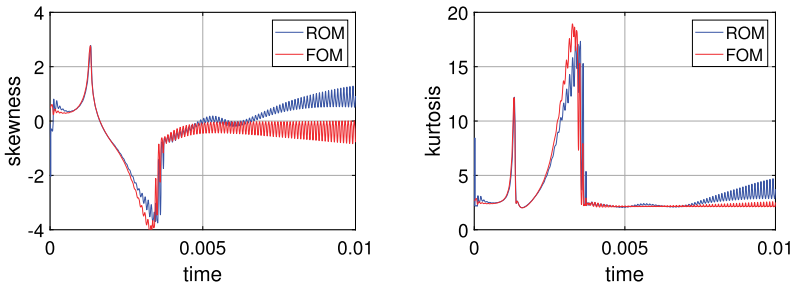


Figure 10.11: Skewness (left) and kurtosis (right) of random output in the anemometer system obtained by the ROM of dimension $r = 125$.

10.6 Conclusions and outlook

We discussed two approaches for MOR of dynamical systems including random parameters to model uncertainties. On the one hand, the stochastic Galerkin method and the stochastic collocation technique produce high-dimensional deterministic dynamical systems, which can be reduced by traditional MOR algorithms. On the other hand, pMOR is applied, where an ROM is sampled instead of the FOM within a quadrature scheme or (quasi-)Monte Carlo method.

In the second approach, the computational effort for the construction of the parametric ROM is critical. If this effort becomes too large, then a quadrature scheme applied to the FOM may yield results of the same quality with lower computation work. Hence we require efficient pMOR methods, where error bounds or error estimates are available to decide the quality of an ROM. More precisely, this error should be in the magnitude of the error in a time integration to accept the results.

As usual, MOR of nonlinear dynamical systems is challenging also in the field of UQ. However, the stochastic Galerkin method is often less efficient in the case of nonlinear dynamical systems, since some probabilistic integrals cannot be evaluated analytically. Thus sampling methods and collocation schemes are preferred. One should check if a given nonlinear dynamical system can be converted into a quadratic-bilinear system, either equivalently or approximately. Consequently, efficient MOR methods are available for QB systems. Research on MOR and parametric MOR still continues for nonlinear dynamical systems.

Bibliography

[1] M. I. Ahmad, P. Benner, and L. Feng, Interpolatory model reduction for quadratic-bilinear systems using error estimators, *Eng. Comput.*, **36** (2019), 25–44.

[2] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, 2005.

[3] F. Augustin, A. Gilg, M. Paffrath, P. Rentrop, and U. Wever, Polynomial chaos for the approximation of uncertainties: chances and limits, *Eur. J. Appl. Math.*, **19** (2008), 149–190.

[4] P. Benner, P. K. Goyal, and S. Gugercin, H_2-quasi-optimal model order reduction for quadratic-bilinear control systems, *SIAM J. Matrix Anal. Appl.*, **39** (2018), 983–1032.

[5] P. Benner, S. Gugercin, and K. Willcox, A survey of projection-based model order reduction methods for parametric dynamical systems, *SIAM Rev.*, **57** (2015), 483–531.

[6] P. Benner, M. Hinze, and E. J. W. ter Maten (eds.), *Model Reduction for Circuit Simulation*, Lect. Notes in Electr. Engng., vol. 74, Springer, 2011.

[7] P. Benner, V. Mehrmann, and D. C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems*, Lect. Notes Comput. Sci. Engin., vol. 45, Springer, 2005.

[8] P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban (eds.), *Model Reduction of Parameterized Systems*, MS&A, vol. 17, Springer, 2017.

[9] P. Benner and T. Stykel, Model order reduction for differential-algebraic equations: a survey, in A. Ilchmann, T. Reis (eds.), *Surveys in Differential-Algebraic Equations IV*, pp. 107–160, Springer, 2017.

[10] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (2010), 2737–2764.

[11] P. Chen and Ch. Schwab, Model order reduction methods in computational uncertainty quantification, in R. Ghanem, D. Higdon, H. Owhadi (eds.), *Handbook of Uncertainty Quantification*, pp. 936–990, Springer, 2017.

[12] O. G. Ernst, A. Mugler, H. J. Starkloff, and E. Ullmann, On the convergence of generalized polynomial chaos expansions, *ESAIM: Math. Model. Numer. Anal.*, **46** (2012), 317–339.

[13] L. Feng, Y. Yue, N. Banagaaya, P. Meuris, W. Schoenmaker, and P. Benner, Parametric modeling and model order reduction for (electro-)thermal analysis of nanoelectronic structures, *J. Math. Ind.*, **6** (2016), 10.

[14] J. Fernandez Villena and L. M. Silveira, Multi-dimensional automatic sampling schemes for multi-point modeling methodologies, *IEEE Trans. CAD Integr. Circuits Syst.*, **30** (2011), 1141–1151.

[15] F. D. Freitas, R. Pulch, and J. Rommes, Fast and accurate model order reduction for spectral methods in uncertainty quantification, *Int. J. Uncertain. Quantificat.*, **6** (2016), 271–286.

[16] R. Freund, Model reduction methods based on Krylov subspaces, *Acta Numer.*, **12** (2003), 267–319.

[17] T. Gerstner and M. Griebel, Numerical integration using sparse grids, *Numer. Algorithms*, **18** (1998), 209–232.

[18] J. S. Hesthaven, B. Stamm, and S. Zhang, Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods, ESAIM, *Math. Model. Numer. Anal.*, **48** (2011), 259–283.

[19] J. Li and D. Xiu, Evaluation of failure probability via surrogate models, *J. Comput. Phys.*, **229** (2010), 8966–8980.

[20] P. Li, F. Liu, X. Li, L. Pileggi, and S. Nassif, Modeling interconnect variability using efficient parametric model order reduction, in *Proc. of Design Automation and Test in Europe Conference (DATE)*, pp. 958–963, 2005.

[21] P. Manfredi, D. Vande Ginste, D. De Zutter, and F. G. Canavero, Stochastic modelling of nonlinear circuits via SPICE-compatible spectral equivalents, *IEEE Trans. Circuits Syst. I, Regul. Pap.*, **61** (2014), 2057–2065.

[22] I. Martini, B. Haasdonk, and G. Rozza, Certified reduced basis approximation for the coupling of viscous and inviscid parameterized flow models, *J. Sci. Comput.*, **74** (2018), 197–219.

[23] N. Mi, S. X.-D. Tan, P. Liu, J. Cui, Y. Cai, and X. Hong, Stochastic extended Krylov subspace method for variational analysis of on-chip power grid networks, in *Proc. ICCAD*, pp. 48–53, 2007.

[24] Ch. Moosmann and A. Greiner, Convective thermal flow problems, in P. Benner, V. Mehrmann, D. C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems*, Lect. Notes Comput. Sci. Engin., vol. 45, pp. 341–343, Springer, 2005.

[25] "MOR Wiki," online document, https://morwiki.mpi-magdeburg.mpg.de/morwiki, cited Sep 2, 2019.

[26] P. B. Nair and A. J. Keane, Stochastic reduced basis methods, *AIAA J.*, **40** (2002), 1653–1664.

[27] M. Navarro Jimenez, O. P. Le Maitre, and O. M. Knio, Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations, *SIAM/ASA J. Uncertain. Quantificat.*, **5** (2017), 378–402.

[28] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, New York, 1992.

[29] F. Nobile, R. Tempone, and C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, **46** (2008), 2309–2345.

[30] M. Ohlberger and S. Rave, Reduced basis methods: success, limitations and future challenges, in *Proceedings of the Conference Algoritmy*, pp. 1–12, 2016.

[31] R. Pulch, Polynomial chaos for linear differential algebraic equations with random parameters, *Int. J. Uncertain. Quantificat.*, **1** (2011), 223–240.

[32] R. Pulch, Stochastic collocation and stochastic Galerkin methods for linear differential algebraic equations, *J. Comput. Appl. Math.*, **262** (2014), 281–291.

[33] R. Pulch, Model order reduction for stochastic expansions of electric circuits, in A. Bartel, M. Clemens, M. Günther, E. J. W. ter Maten (eds.), *Scientific Computing in Electrical Engineering SCEE 2014*, Mathematics in Industry, vol. 23, pp. 223–232, Springer, 2016.

[34] R. Pulch, A Hankel norm for quadrature rules solving random linear dynamical systems, *J. Comput. Appl. Math.*, **316** (2017), 307–318.

[35] R. Pulch, Model order reduction and low-dimensional representations for random linear dynamical systems, *Math. Comput. Simul.*, **144** (2018), 1–20.

[36] R. Pulch, Model order reduction for random nonlinear dynamical systems and low-dimensional representations for their quantities of interest, *Math. Comput. Simul.*, **166** (2019), 76–92.

[37] R. Pulch and F. Augustin, Stability preservation in stochastic Galerkin projections of dynamical systems, *SIAM/ASA J. Uncertain. Quantificat.*, **7** (2019), 634–651.

[38] R. Pulch and E. J. W. ter Maten, Stochastic Galerkin methods and model order reduction for linear dynamical systems, *Int. J. Uncertain. Quantificat.*, **5** (2015), 255–273.

[39] S. K. Sachdeva, P. B. Nair, and A. J. Keane, Hybridization of stochastic reduced basis methods with polynomial chaos expansions, *Probab. Eng. Mech.*, **21** (2006), 182–192.

[40] W. H. A. Schilders, M. A. van der Vorst, and J. Rommes (eds.), *Model Order Reduction: Theory, Research Aspects and Applications*, Mathematics in Industry, vol. 13, Springer, 2008.

[41] T. Soll and R. Pulch, Sample selection based on sensitivity analysis in parameterized model order reduction, *J. Comput. Appl. Math.*, **316** (2017), 271–286.

[42] B. Sonday, R. Berry, B. Debusschere, and H. Najm, Eigenvalues of the Jacobian of a Galerkin-projected uncertain ODE system, *SIAM J. Sci. Comput.*, **33** (2011), 1212–1233.

[43] A. Stroud, *Approximative Calculation of Multiple Integrals*, Prentice-Hall, Inc., 1971.

[44] T. J. Sullivan, *Introduction to Uncertainty Quantification*, Springer, 2015.

[45] D. Xiu and G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, **24** (2002), 619–644.

[46] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.

[47] Y. Zou, Y. Cai, Q. Zhou, X. Hong, S. X.-D. Tan, and L. Kang, Practical implementation of the stochastic parameterized model order reduction via Hermite polynomial chaos, in *Proc. ASP-DAC*, pp. 367–372, 2007.

Xiaodong Cheng, Jacquelien M. A. Scherpen, and
Harry L. Trentelman

11 Reduced-order modeling of large-scale network systems

Abstract: Large-scale network systems describe a wide class of complex dynamical systems composed of many interacting subsystems. A large number of subsystems and their high-dimensional dynamics often result in highly complex topology and dynamics, which pose challenges to network management and operation. This chapter provides an overview of reduced-order modeling techniques that are developed recently for simplifying complex dynamical networks. In the first part, clustering-based approaches are reviewed, which aim to reduce the network scale, i. e., find a simplified network with a fewer number of nodes. The second part presents structure-preserving methods based on generalized balanced truncation, which can reduce the dynamics of each subsystem.

Keywords: Reduced-order modeling, graph clustering, balanced truncation, semistable systems, Laplacian matrix

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

11.1 Introduction

Network systems, or multiagent systems, are a class of structured systems composed of multiple interacting subsystems. In real life, systems taking the form of networks are ubiquitous, and the study of network systems has received compelling attention from many disciplines; see, e. g., [61, 60, 52] for an overview. Coupled chemical oscillators, cellular and metabolic networks, interconnected physical systems, and electrical power grids are only a few examples of such systems. To capture the behaviors and properties of network systems, graph theory is often useful [37]. The interconnection structure among the subsystems can be represented by a *graph*, in which *vertices* and *edges* represent the subsystems and the interactions among them, respectively. However, when network systems are becoming more large-scale, we have to deal with graphs of complex topology and nodal dynamics, which can cause great difficulty in transient analysis, failure detection, distributed controller design, and system simulation. From a practical point of view, it is always desirable to construct a reduced-order model to capture the essential behavior of the original system e. g., stability and passivity, frequency response, and inoput/output properties, while avoiding too ex-

Xiaodong Cheng, Eindhoven University of Technology, Eindhoven, Netherlands
Jacquelien M. A. Scherpen, Harry L. Trentelman, University of Groningen, Groningen, Netherlands

https://doi.org/10.1515/9783110499001-011

pensive computation. In the reduction of network systems, reduced-order models are designed not only to capture the main input–output feature of original complex network models but also to preserve the network structure such that they are usable for some potential applications, e. g., distributed controller design and sensor allocation in dynamic networks.

In the past few decades, a variety of theories and techniques of model reduction have been intensively investigated for generic dynamical systems. Techniques, including Krylov subspace methods (also known as moment-matching), balanced truncation, and Hankel norm approximation [4, 3, 59, 36], provide us systematic procedures to generate reduced-order models that well approximate the input–output mapping of a high-dimensional system; see [2, 5, 6] for an overview. However, when addressing the reduction of dynamical networks, the direct application of these methods may be not advisable, since they potentially destroy the network structure such that obtained reduced-order models could not have the network feature any more. Structure-preserving model reduction is crucial for the application of network systems. Taking into account the two aspects of the complexity of network systems, namely, large-scale interconnection (i. e., a large number of subsystems) and high-dimensional subsystems, two types of problems are studied in the literature towards the approximation of network systems in a structure-preserving manner.

The first problem aims to simplify the underlying network topology by reducing the number of nodes. The mainstream methods for this problem are based on *graph clustering*, which is an unsupervised learning technique widely used in data science and computer graphics [45, 71]. For approximating dynamical networks, clustering-based methods basically follow a two-step process: The first step is to partition the nodes into several nonoverlapping subsets (clusters), and then all the nodes in each cluster are aggregated into a single node. The aggregation step can be interpreted as a Petrov–Galerkin approximation using a clustering-based projection; see [74, 42, 58]. However, how to find the "best" clustering such that the approximation error is minimized still remains an open question. In [58, 46], a particular clustering, called *almost equitable partition*, is considered, which leads to an analytic $\mathcal{H}_2$ expression for the reduction error. However, finding almost equitable partitions itself is rather difficult and computationally expensive for general graphs. Clustering can also be found using the QR decomposition with column pivoting on the projection matrix obtained by the Krylov subspace method [54]. For undirected networks with tree topology, an asymptotically stable *edge system* can be considered, which has a pair of diagonal generalized Gramian matrices for characterizing the importance of edges. Then, vertices linked by the less important edges are iteratively clustered [8]. The notion of *reducibility* is introduced in [42, 41, 43] to characterize the uncontrollability of clusters. Using this notion, an upper bound for the network reduction error is established, which can determine the clustering. The works in [11, 13] extend the notion of dissimilarity for dynamical systems, where nodal behaviors are represented by the transfer functions mapping from external inputs to node states, and dissimilarity between two

nodes are quantified by the norm of their behavior deviation. Then clustering algorithms, e. g., hierarchical clustering and K-means clustering, can be adapted to group nodes in such a way that nodes in the same cluster are more similar to each other than to those in other clusters [12, 63]. Subsequent research in [12, 22, 17, 19] shows that the dissimilarity-based clustering method can also be extended to second-order networks, controlled power networks, and directed networks. In [25, 24], a framework is presented on how to build a reduced-order model from a given clustering. The edge weights in the reduced network are parameterized so that an optimization problem is formulated to minimize the reduction error.

An alternative methodology to simplify the complexity of the network structure is based on time scale analysis, and in particular, *singular perturbation approximation*; see some of earlier works in [73, 64, 10]. Recently, this approach has also been extensively applied to biochemical systems and electric networks [65, 1, 39, 44, 27, 32, 56, 67]. This class of approaches relies on the fact that the nodal states of network systems evolve over different time scales. Removing the vertices with fast states and reconnecting the remaining vertices with slow states will generate a reduced-order model that retains the low frequency behavior of the original network system. This methodology is closely related to the so-called *Kron reduction* in electric networks [27, 32, 56], where the Schur complement of a graph Laplacian is taken that is again a Laplacian of a smaller-scale network. The singular perturbation approximation is capable of preserving the physical meaning of a network system. However, how to identify and separate fast/slow states is a crucial step in this approach, and its application is highly dependent on specific systems.

A network system can be simplified if the dimension of individual subsystems is reduced, which leads to the second research direction in reduced-order modeling of network systems; see, e. g., [69, 57, 23]. In this framework, the approximation is applied to each subsystem in a way that certain properties of the overall network, such as synchronization and stability, are preserved. Relevant methods are developed using generalized Gramian matrices [34] that allow for more freedom to preserve some desired structures than the standard Gramians. Networked nonlinear robustly synchronized Lur'e-type systems are reduced in [23], which shows that performing model reduction on the linear component of each nonlinear subsystem allows for preserving the robust synchronization property of a Lur'e network. Techniques in [43, 21] can reduce the complexity of network structures and subsystem dynamics *simultaneously*. In [43], the graph structure is simplified using clustering, while subsystems are reduced via some orthogonal projection. In contrast, [21] reduces graph structure and subsystem dynamics in a unified framework of generalized balanced truncation. Although a reduced-order system that is obtained by balanced truncation does not necessarily preserve the network structure, a set of coordinates can be found in which the reduced-order model has a network interpretation.

In this chapter, we will focus on the two problems of model reduction for linear network systems with diffusive couplings. In the aspect of simplifying network

topology, we only review several clustering-based methods for space reasons. For the reduction of subsystems, we present the generalized balanced truncation as the main approach to perform a synchronization-preserving model reduction. The rest of this chapter is organized as follows. In Section 11.2, we provide preliminaries on balanced truncation, semi-stable systems, and necessary concepts in graph theory. The model of diffusively coupled networks is also introduced. In Section 11.3, we present clustering-based model reduction methods for simplifying network topology, and in Section 11.4, the generalized balanced truncation approach is reviewed to reduce the dimension of subsystems. In Section 11.5, we glance at open problems and make some concluding remarks.

Notation

The symbols $\mathbb{R}$ and $\mathbb{R}_+$ denote the set of real numbers and real positive numbers, respectively; I_n is the identity matrix of size n, and $\mathbb{1}_n$ represents the vector in $\mathbb{R}^n$ of all ones; $\mathbf{e}_i$ is the i-th column of I_n; the cardinality of a finite set $\mathcal{S}$ is denoted by $|\mathcal{S}|$; $\mathrm{Tr}(A)$, $\mathrm{im}(A)$, $\ker(A)$ denote the trace, image, and kernel of a matrix A, respectively; and $\|G(s)\|_{\mathcal{H}_\infty}$ and $\|G(s)\|_{\mathcal{H}_2}$ represent the $\mathcal{H}_\infty$-norm and $\mathcal{H}_2$-norm of a transfer matrix $G(s)$.

11.2 Preliminaries

In this section, we first briefly recapitulate the theory of balancing as a basis for the model reduction of linear control systems. New results for semi-stable systems and pseudo-Gramians are also introduced. Moreover, we review some basic concepts from graph theory, which are then used for the modeling of network systems.

11.2.1 Generalized balanced truncation

From [34, 2], we review some basic facts on model reduction by using *generalized balanced truncation*. Consider a linear time-invariant system

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{11.1}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{q\times n}$, whose transfer function is given by $G(s) := C(sI_n - A)^{-1}B$. Let the system (11.1) be asymptotically stable and minimal, i. e., A is Hurwitz, the pair (A, B) is controllable, and the pair (C, A) is observable. Note that if a system (11.1) is not minimal, we can always use the Kalman decomposition to remove the uncontrollable or unobservable states from the model (11.1). Thus, a minimal state-space realization can be obtained, of which the transfer function also is equal to $G(s)$.

For such a system (11.1), there always exist positive definite matrices P and Q satisfying the following Lyapunov inequalities:

$$AP + PA^\top + BB^\top \leq 0, \tag{11.2a}$$
$$A^\top Q + QA + C^\top C \leq 0. \tag{11.2b}$$

Any P and Q as the solutions of (11.2) are called *generalized controllability and observability Gramians* of the system (11.1) [34]. When the equalities are achieved in (11.2), we obtain the standard controllability and observability Gramians, which become unique solutions of the Lyapunov equations [2].

Similar to the standard balancing, generalized balancing of the system (11.1) amounts to finding a nonsingular matrix $T \in \mathbb{R}^{n\times n}$ such that P and Q are simultaneously diagonalized in the following way:

$$TPT^\top = T^{-\top}QT^\top = \Sigma := \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n), \tag{11.3}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ are called *generalized Hankel singular values* (GHSVs) of system (11.1). Using T as a coordinate transformation, we obtain a balanced realization of system (11.1), in which the state components corresponding to the smaller GHSVs are relatively difficult to reach and observer and thus have less influence on the input–output behavior. Let the triplet $(\hat{A}, \hat{B}, \hat{C})$ be the r-dimensional reduced-order model (with $r \ll n$) obtained by truncating the states with the smallest GHSVs in the balanced system. Then, the reduced-order model $\hat{G}(s) := \hat{C}(sI_r - \hat{A})^{-1}\hat{B}$ preserves stability and moreover, an a priori upper bound for the approximation error can be expressed in terms of the neglected GHSVs, i. e.,

$$\left\|G(s) - \hat{G}(s)\right\|_{\mathcal{H}_\infty} \leq 2\sum_{i=r+1}^{n} \sigma_i. \tag{11.4}$$

11.2.2 Semi-stable systems and pseudo-Gramians

Semi-stability is a more general concept than asymptotic stability as it allows for multiple zero poles in a system [9, 40]. A linear system $\dot{x} = Ax$ is *semi-stable* if $\lim_{t\to\infty} e^{At}$ exists for all initial conditions $x(0)$. The following lemma provides an equivalent condition for *semi-stability*.

Lemma 1. [7] *A system $\dot{x} = Ax$ is semi-stable if and only if the zero eigenvalues of A are semi-simple (i. e., the geometric multiplicity of the zero eigenvalues coincides with the algebraic multiplicity), and all the other eigenvalues have negative real parts.*

Let the triplet (A, B, C) be a linear semi-stable system. The definition of semi-stability implies that the transfer $G(s) = C(sI - A)^{-1}B$ is not necessarily in the $\mathcal{H}_2$-space,

and the standard controllability and observability Gramians in [2] are not well-defined in this case. Instead, we can define a pair of *pseudo-Gramians* as follows [20]:

$$\mathcal{P} = \int_0^\infty (e^{At} - \mathcal{J})BB^\top (e^{A^\top t} - \mathcal{J}^\top)\,\mathrm{d}t, \quad \mathcal{Q} = \int_0^\infty (e^{A^\top t} - \mathcal{J}^\top)C^\top C(e^{At} - \mathcal{J})\,\mathrm{d}t, \tag{11.5}$$

where $\mathcal{J} := \lim_{t\to\infty} e^{At}$ is a constant matrix. The pseudo-Gramians $\mathcal{P}$ and $\mathcal{Q}$ in (11.5) are well-defined for semi-stable systems and can be viewed as a generalization of standard Gramian matrices for asymptotically stable systems. Furthermore, the pseudo-Gramians can be computed as

$$\mathcal{P} = \tilde{\mathcal{P}} - \mathcal{J}\tilde{\mathcal{P}}\mathcal{J}^\top, \quad \mathcal{Q} = \tilde{\mathcal{Q}} - \mathcal{J}^\top\tilde{\mathcal{Q}}\mathcal{J}, \tag{11.6}$$

where $\tilde{\mathcal{P}}$ and $\tilde{\mathcal{Q}}$ are arbitrary symmetric solution of the Lyapunov equations

$$\begin{aligned} A\tilde{\mathcal{P}} + \tilde{\mathcal{P}}A^\top + (I - \mathcal{J})BB^\top(I - \mathcal{J}^\top) &= 0, \\ A^\top\tilde{\mathcal{Q}} + \tilde{\mathcal{Q}}A + (I - \mathcal{J}^\top)C^\top C(I - \mathcal{J}) &= 0, \end{aligned}$$

respectively. The pseudo-Gramians lead to a characterization of the $\mathcal{H}_2$-norm of a semi-stable system.

Theorem 1. [20] *Consider a semi-stable system with the triplet* (A, B, C). *Then,* $G(s) := C(sI - A)^{-1}B \in \mathcal{H}_2$ *if and only if* $C\mathcal{J}B = 0$. *Furthermore, if* $\|G(s)\|_{\mathcal{H}_2}$ *is well-defined, then*

$$\|G(s)\|_{\mathcal{H}_2}^2 = \mathrm{Tr}(C\mathcal{P}C^\top) = \mathrm{Tr}(B^\top\mathcal{Q}B). \tag{11.7}$$

11.2.3 Graph theory

The concepts in graph theory are instrumental in analyzing network systems [52]. The interconnection structure of a network is often characterized by a graph $\mathcal{G}$ that consists of a finite and nonempty node set $\mathcal{V} := \{1, 2, \ldots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each element in $\mathcal{E}$ is an ordered pair of elements of $\mathcal{V}$, and we say that the edge is directed from vertex i to vertex j if $(i, j) \in \mathcal{E}$. This leads to the definition of the *incidence matrix* $R \in \mathbb{R}^{n\times|\mathcal{E}|}$:

$$[R]_{ij} = \begin{cases} +1 & \text{if edge } (i,j) \in \mathcal{E}, \\ -1 & \text{if edge } (j,i) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{11.8}$$

If each edge is assigned a positive value (weight), the graph $\mathcal{G}$ is *weighted*, and a *weighted adjacency matrix* $\mathcal{W}$ can be defined such that $w_{ij} = [\mathcal{W}]_{ij}$ is positive if there exists a directed edge from node j to node i, i. e., $(j, i) \in \mathcal{E}$, and $w_{ij} = 0$ otherwise.

A (directed) graph $\mathcal{G}$ is called *undirected* if $\mathcal{W}$ is symmetric. An undirected graph $\mathcal{G}$ is called *simple* if $\mathcal{G}$ does not contain self-loops (i. e., $\mathcal{E}$ does not contain edges of the form (i, i), $\forall\, i$), and there exists only one undirected edge between any two distinct nodes. Two distinct vertices i and j are said to be *neighbors* if there exists an edge between i and j, and the set $\mathcal{N}_i$ denotes all the neighbors of node i.

The *Laplacian matrix* $L \in \mathbb{R}^{n\times n}$ of a weighted graph $\mathcal{G}$ is defined as

$$[L]_{ij} = \begin{cases} \sum_{j\in\mathcal{N}_i} w_{ij}, & i = j, \\ -w_{ij}, & \text{otherwise.} \end{cases} \tag{11.9}$$

Furthermore, we can define an undirected graph Laplacian using an alternative formula:

$$L = RWR^{\top}, \tag{11.10}$$

where R is an incidence matrix obtained by assigning an arbitrary orientation to each edge of $\mathcal{G}$ and $W := \operatorname{diag}(w_1, w_2, \ldots, w_{|\mathcal{E}|})$, with w_k being the weight associated with the edge k, for each $k = 1, 2, \ldots, |\mathcal{E}|$.

Remark 1. If $\mathcal{G}$ is a simple undirected connected graph, the associated Laplacian matrix L has the following structural properties:

1. $L^{\top} = L \geq 0$;
2. $\ker(L) = \operatorname{im}(\mathbb{1})$;
3. $L_{ij} \leq 0$ if $i \neq j$, and $L_{ij} > 0$ otherwise.

Conversely, any real square matrix satisfying the above conditions can be interpreted as a Laplacian matrix that uniquely represents a simple undirected connected graph.

11.2.4 Network systems

In this chapter, we mainly focus on an important class of networks, namely, *consensus networks*, where subsystems are interconnected via *diffusive couplings*. Various applications, including formation control of mobile vehicles, synchronization in power networks, and balancing in chemical kinetics, involve the concept of consensus networks [66, 50, 35, 33, 53, 72, 75].

Here, we consider a network system in which the interconnection structure is represented by a simple weighted undirected graph with the node set $\mathcal{V} = \{1, 2, \ldots, n\}$. The dynamics of each vertex (agent) are described by

$$\boldsymbol{\Sigma}_i : \begin{cases} \dot{x}_i = Ax_i + Bv_i, \\ \eta_i = Cx_i, \end{cases} \tag{11.11}$$

where $x_i \in \mathbb{R}^\ell$, $v_i \in \mathbb{R}^m$, and $\eta_i \in \mathbb{R}^m$ are the state, control input, and output of node i, respectively. The n subsystems are interconnected such that

$$m_i v_i = -\sum_{j\in\mathcal{N}_i} w_{ij}(\eta_i - \eta_j) + \sum_{j=1}^{p} f_{ij}u_j, \tag{11.12}$$

where $m_i \in \mathbb{R}_+$ denotes the weight of node i. In (11.12), the first term on the left is referred to as *diffusive coupling*, where $w_{ij} \in \mathbb{R}_+$ is the entry of the adjacency matrix $[\mathcal{W}]_{ij}$ standing for the intensity of the coupling between nodes i and j. The second term indicates the influence from external input u_j, where the value of $f_{ij} \in \mathbb{R}$ represents the amplification of u_j acting on vertex i. Let $F \in \mathbb{R}^{n\times p}$ be a matrix with $[F]_{ij} = f_{ij}$, and we introduce the external outputs as $y_i = \sum_{j=1}^{n}[H]_{ij}\eta_j$, with $y_i \in \mathbb{R}^m$ as the i-th external output of the network. We then represent the network system in compact form as

$$\boldsymbol{\Sigma}: \begin{cases} (M\otimes I)\dot{x} = (M\otimes A - L\otimes BC)x + (F\otimes B)u, \\ y = (H\otimes C)x, \end{cases} \tag{11.13}$$

with joint state vector $x^\top := [x_1^\top\ x_2^\top\ \dots\ x_n^\top] \in \mathbb{R}^{n\ell}$, external control input $u^\top := [u_1^\top\ u_2^\top\ \dots\ u_p^\top] \in \mathbb{R}^{pm}$, and external output $y = [y_1^\top\ y_2^\top\ \dots\ y_q^\top] \in \mathbb{R}^{qm}$; $M := \mathrm{diag}(m_1, m_2, \dots, m_n) > 0$, and $L \in \mathbb{R}^{n\times n}$ is the graph Laplacian matrix that characterizes the coupling structure among the subsystems. In many studies of undirected networks, the matrix $M = I_n$ is considered.

The simplest scenario in network systems is that all the vertices are just single-integrators, i. e., $m_i\dot{x}_i = v_i$ with $x_i \in \mathbb{R}$. Then, the model of a networked single-integrator system can be formed by taking $A = 0$ and $B = C = 1$ in (11.13), which leads to

$$\begin{cases} M\dot{x} = -Lx + Fu, \\ y = Hx. \end{cases} \tag{11.14}$$

A variety of physical systems are of this form, such as mass–damper systems and single-species reaction networks [74]. Note that the system (11.14) is call *semi-stable* [9], since L has a simple zero eigenvalue.

An important issue in the context of diffusively coupled networks is *synchronization*. The system $\boldsymbol{\Sigma}$ in (11.13) achieves synchronization if, for any initial conditions, the zero input response of (11.13) satisfies

$$\lim_{t\to\infty}[x_i(t) - x_j(t)] = 0, \quad \text{for all } i, j \in \mathcal{V}. \tag{11.15}$$

Using the property of L in Remark 1, it is clear that the single-integrator network in (11.14) can reach synchronization. However, for the general form of (11.13), we need to take into account the subsystems as well. Denote by $0 = \lambda_1 < \lambda_2 \le \cdots \le \lambda_n$ the eigenvalues of the matrix $M^{-1}L$ in ascending order. A sufficient and necessary condition for the synchronization of a network consisting of agents as in (11.11) is found in, e. g., [50].

Lemma 2. *The multiagent system* $\mathbf{\Sigma}$ *in* (11.13) *achieves synchronization if and only if* $A - \lambda_k BC$ *is Hurwitz, for all* $k \in \{2, 3, \dots, n\}$.

11.3 Clustering-based model reduction

In this section, we introduce clustering-based methods that combine the Petrov–Galerkin projection with graph clustering. A reduced-order network model can be constructed by using the characteristic matrix of a graph clustering. Moreover, we will also briefly recap some other clustering-based methods.

Graph clustering is an important notion in graph theory [37]. Consider a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A *graph clustering* of $\mathcal{G}$ is to divide its vertex set $\mathcal{V}$ (with $|\mathcal{V}| = n$) into r nonempty and disjoint subsets, denoted by $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r$, where $\mathcal{C}_i$ is called a *cluster* (or a *cell* of $\mathcal{G}$).

Definition 1. The characteristic matrix of the clustering $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r\}$ is characterized by the binary matrix $\Pi \in \mathbb{R}^{n\times r}$ as

$$[\Pi]_{ij} := \begin{cases} 1, & \text{if vertex } i \in \mathcal{C}_j, \\ 0, & \text{otherwise.} \end{cases} \tag{11.16}$$

Note that each vertex is assigned to a unique cluster. Therefore, each row of the characteristic matrix Π has exactly one nonzero element, and the number of nonzero elements in each column of Π is the number of vertices in the corresponding cluster. Specifically, we have

$$\Pi \mathbb{1}_r = \mathbb{1}_n \quad \text{and} \quad \mathbb{1}_n^\top \Pi = [|\mathcal{C}_1|, |\mathcal{C}_2|, \dots, |\mathcal{C}_r|]. \tag{11.17}$$

It is worth noting that for any given undirected graph Laplacian L, the reduced matrix $\Pi^\top L \Pi$ is also a Laplacian matrix, representing an undirected graph of smaller size. This important property allows for a structure-preserving model reduction of network systems using Π for the Petrov–Galerkin projection.

Let $\mathbf{\Sigma}$ in (11.13) be a network system with underlying graph $\mathcal{G}$ of n vertices. To formulate a reduced-order network model of r dimensions, we first find a graph clustering that partitions the vertices of $\mathcal{G}$ into r clusters. Then we use the characteristic matrix of the clustering as a basis that projects the state space of $\mathbf{\Sigma}$ to a reduced subspace. Specifically, a reduced-order model of $\mathbf{\Sigma}$ can be constructed via the Petrov–Galerkin projection framework as

$$\hat{\mathbf{\Sigma}} : \begin{cases} (\hat{M} \otimes I)\dot{z} = (\hat{M} \otimes A - \hat{L} \otimes B)z + (\hat{F} \otimes B)u, \\ \hat{y} = (\hat{H} \otimes C)z, \end{cases} \tag{11.18}$$

where $\hat{M} := \Pi^\top M \Pi \in \mathbb{R}^{r\times r}$, $\hat{L} := \Pi^\top L \Pi \in \mathbb{R}^{r\times r}$, $\hat{F} = \Pi^\top F$, and $\hat{H} = H\Pi$. The new state vector $z^\top := [z_1^\top \; z_2^\top \; \dots \; z_r^\top] \in \mathbb{R}^{r\ell}$, where each component $z_i \in \mathbb{R}^\ell$ represents an estimated dynamics of all the vertices in the i-th cluster, and $\hat{x} = (\Pi \otimes I)z \in \mathbb{R}^{n\ell}$ can be an

approximation of the original state x. For the single-integrator network system (11.14), clustering-based projection yields the reduced-order model as

$$\begin{cases} \hat{M}\dot{z} = -\hat{L}z + \hat{F}u, \\ \hat{y} = \hat{H}z. \end{cases} \tag{11.19}$$

In the reduced-order models in (11.18) and (11.19), $\hat{M}$ is a positive diagonal matrix, and $\hat{L} \in \mathbb{R}^{r\times r}$ is a Laplacian matrix representing a graph of a lower number of vertices. More preciously, $\hat{M}$ and $\hat{L}$ can be computed as

$$[\hat{M}]_{kk} = \sum_{i\in\mathcal{C}_k} m_i, \quad [\hat{L}]_{kl} = \begin{cases} \sum_{i\in\mathcal{C}_k, j\in\mathcal{C}_l}[L]_{ij}, & k \neq l, \\ \sum_{i\in\mathcal{C}_k}[L]_{ij}, & k = l. \end{cases} \tag{11.20}$$

Clearly, the reduced-order models in (11.18) and (11.19) preserve the network structure and thus can be interpreted as simplified dynamical networks with diffusive couplings. The following example then illustrates the physical meaning of a projected reduced-order network model.

Example 1. Consider a mass–damper system in Figure 11.1 (left inset), where u_1, u_2 are external forces acting on the first and fourth mass blocks. Suppose that all the masses are identical. Then we model the network system in the form of (11.14) with

$$M = I_5, \quad L = \begin{bmatrix} 6 & -3 & 0 & -2 & -1 \\ -3 & 4 & -1 & 0 & 0 \\ 0 & -1 & 6 & -2 & -3 \\ -2 & 0 & -2 & 5 & -1 \\ -1 & 0 & -3 & -1 & 5 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The off-diagonal entry $-[L]_{ij}$ represents the damping coefficient of the edge (i, j). Let $\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\} = \{\{1, 2\}, \{3, 5\}, \{4\}\}$ be the clustering of the network, which leads to

$$\Pi = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^\top.$$

A reduced-order network model is obtained as in (11.19) with

$$\hat{M} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{L} = \begin{bmatrix} 4 & -2 & -2 \\ -2 & 5 & -3 \\ -2 & -3 & 5 \end{bmatrix}, \quad \hat{F} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

It is clear that each mass in the reduced network is equal to the sums of the masses in the corresponding cluster. Moreover, the structure of a Laplacian matrix is retained, which allows for a physical interpretation of the reduced model, as shown in Figure 11.1 (right inset).

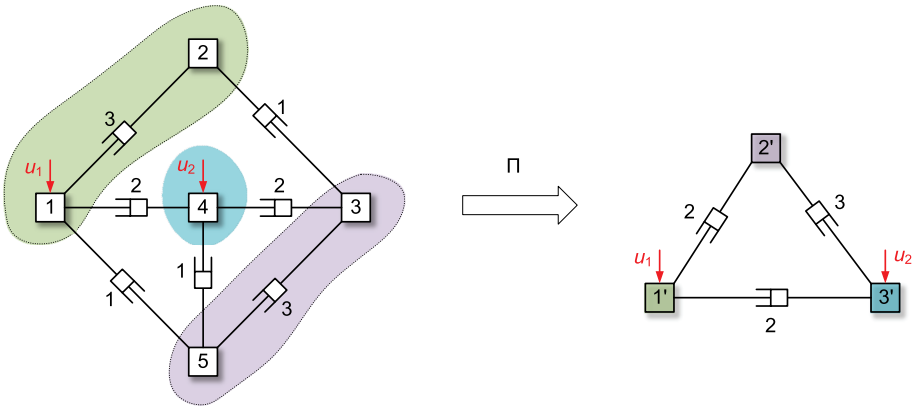


Figure 11.1: An illustrative example of clustering-based model reduction for a mass–damper network system.

Next, the properties of the reduced-order models in (11.18) and (11.19) are discussed. First, it is clear that system (11.19) preserves the synchronization property. Moreover, the following result holds.

Lemma 3. [11, 13] *Consider the single-integrator networks in* (11.14) *and* (11.19). *The impulse responses of the two systems satisfy*

$$\lim_{t\to\infty} y(t) = \lim_{t\to\infty} \hat{y}(t) = \frac{H\mathbb{1}_n\mathbb{1}_n^\top F}{\mathbb{1}_n^\top M\mathbb{1}_n}. \tag{11.21}$$

Denote

$$S := H(sI_n + L)^{-1}F, \quad \hat{S} := \hat{H}(sI_r + \hat{L})^{-1}\hat{F}. \tag{11.22}$$

This lemma implies the reduction error $\|S - \hat{S}\|_{\mathcal{H}_2}$ is well-defined, for any clustering Π. For the reduced-order network system (11.18), the analysis of synchronization and reduction error is more complicated, since the subsystem dynamics will also be involved. Denote

$$G(s) := (H \otimes C)[M \otimes (sI_\ell - A) + L \otimes BC](F \otimes B), \tag{11.23a}$$

$$\hat{G}(s) := (\hat{H} \otimes C)[\hat{M} \otimes (sI_\ell - A) + \hat{L} \otimes BC](\hat{F} \otimes B) \tag{11.23b}$$

as the transfer matrices of the systems (11.13) and (11.18), respectively. Generally, $G(s) - \hat{G}(s)$ is not guaranteed to be stable. However, a theoretical guarantee can be obtained if the subsystem (A, B, C) in (11.11) is *passive* [38], namely, there exists a positive definite K such that

$$A^\top K + KA \le 0, \quad \text{and} \quad C^\top = BK. \tag{11.24}$$

In this case, we have the synchronization property and bounded reduction error for the system (11.18).

Theorem 2. *Consider the subsystem* (A, B, C) *in* (11.11)*, which is passive and minimal. Then the following statements hold.*

1. *The original network system* (11.13) *achieves synchronization for any L representing an undirected connected graph* [70, 26].
2. *The reduced-order network system* (11.18) *achieves synchronization for any clustering* Π [8, 13].
3. $G(s) - \hat{G}(s) \in \mathcal{H}_2$*, for any clustering* Π [8, 13].

In the framework of clustering-based projection, the approximation error $\|G(s) - \hat{G}(s)\|_{\mathcal{H}_2}$ only depends on the choice of graph clustering. Thus, it is a crucial problem in this framework to select a suitable clustering such that the obtained reduced-order model (11.18) can well approximate the behavior of the original network system (11.13). In the following subsections, we review several cluster selection methods.

11.3.1 Almost equitable partitions

It is suggested in [58] to place those vertices that connect to the rest of the network in a similar fashion into the same cluster. This idea leads to a special class of graph clusterings, namely, *almost equitable partitions*.

Definition 2. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a weighted undirected graph. A graph clustering $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ is called an almost equitable partition if for each $\mu, \nu \in \{1, 2, \ldots, r\}$ with $\mu \neq \nu$, we have $\sum_{k \in \mathcal{C}_\nu} w_{ik} = \sum_{k \in \mathcal{C}_\nu} w_{jk}$, $\forall\, i, j \in \mathcal{C}_\mu$, where w_{ij} denotes the (i, j)-th entry of the adjacency matrix of $\mathcal{G}$.

If $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ is an almost equitable partition, its characteristic matrix Π has the key property that $\mathrm{im}(\Pi)$ is L-invariant [58], i. e., $L\,\mathrm{im}(\Pi) \subseteq \mathrm{im}(\Pi)$.

Consider the single-integrator network in (11.14) with $\mathcal{V}$ being the vertex set. In the context of leader–follower networks, a subset of vertices $\mathcal{V}_L \subseteq \mathcal{V}$ are the leaders, with $|\mathcal{V}_L| = p$, which are controlled by external inputs. Moreover, $F \in \mathbb{R}^{n \times p}$ in (11.14) is the binary matrix such that $[F]_{ij} = 1$ if vertex i is the j-th leader, and $[F]_{ij} = 0$ otherwise. Assume that the output of (11.14) is given as

$$y = Hx = W^{\frac{1}{2}} R^\top x, \tag{11.25}$$

where R is the incidence matrix of $\mathcal{G}$ and W is the edge weight matrix defined in (11.10). Then, the output of the reduced network model (11.19) is obtained as $\hat{y} = \hat{H}x = W^{\frac{1}{2}} R^\top \Pi x$ with Π being the characteristic matrix of the given almost equitable partition. Using the property of the output matrices that $H^\top H = L$ and $\hat{H}^\top \hat{H} = \hat{L}$,

an explicit $\mathcal{H}_2$-error can be derived, which is characterized by the cardinalities of the clusters containing leaders [58, 46].

Theorem 3. *Consider the network system* (11.14) *with output defined in* (11.25). *Let* Π *be the characteristic matrix of an almost equitable partition of the underlying graph:* $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$. *Denote* S *and* $\hat{S}$ *as the transfer matrices of* (11.14) *and* (11.19), *respectively. Then, we have*

$$\frac{\|S - \hat{S}\|_{\mathcal{H}_2}^2}{\|S\|_{\mathcal{H}_2}^2} = \frac{\|S\|_{\mathcal{H}_2}^2 - \|\hat{S}\|_{\mathcal{H}_2}^2}{\|S\|_{\mathcal{H}_2}^2} = \frac{\sum_{i=1}^{p}(1 - \frac{1}{|\mathcal{C}_{k_i}|})}{p(1 - \frac{1}{n})}, \tag{11.26}$$

where $n = |\mathcal{V}|$, $p = |\mathcal{V}_L|$, *and* k_i *is the integer index such that the* i*-th leader is within* $\mathcal{C}_{k_i}$.

In [46], a formula for the $\mathcal{H}_\infty$-error is also derived by assuming a specific output $y = Lx$ in (11.14). If the network (11.14) is clustered according to an almost equitable partition $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$, then we have

$$\|S - \hat{S}\|_{\mathcal{H}_\infty}^2 = \begin{cases} \max_{1 \le i \le q}(1 - \frac{1}{|\mathcal{C}_{k_i}|}) & \text{if the leaders are in different clusters,} \\ 1 & \text{otherwise,} \end{cases}$$

with k_i being the integer such that the i-th leader is within $\mathcal{C}_{k_i}$.

More results on model reduction methods based on almost equitable partitions can be found in [46], where network systems of the form (11.13) with symmetric subsystems are also discussed.

11.3.2 Clustering of tree networks

If the underlying graph of the considered network model (11.13) is a tree, we can resort to the model reduction procedure proposed in [8]. Consider the network model $\mathbf{\Sigma}$ in (11.13), where the subsystems are passive and minimal and the Laplacian matrix L represents an undirected tree graph $\mathcal{T}$. Note that if $\mathcal{T}$ contains n vertices, then it has $n - 1$ edges. Relevant to (11.10), an *edge Laplacian* is defined:

$$L_{\mathrm{e}} = R^{\top} R W, \tag{11.27}$$

where $R \in \mathbb{R}^{n \times (n-1)}$ is the oriented incidence matrix of $\mathcal{T}$ and $W \in \mathbb{R}^{(n-1)\times(n-1)}$ is the edge weight matrix. It is not hard to see that L_{e} has all eigenvalues real and positive, and these eigenvalues coincide to the nonzero eigenvalues of L.

Let $M = I_n$ in (11.13), and an edge system can be defined as

$$\mathbf{\Sigma}_{\mathrm{e}} : \begin{cases} \dot{x}_{\mathrm{e}} = (I_{n-1} \otimes A - L_{\mathrm{e}} \otimes BC)x_{\mathrm{e}} + (F_{\mathrm{e}} \otimes B)u, \\ y_{\mathrm{e}} = (H_{\mathrm{e}} \otimes C)x_{\mathrm{e}}, \end{cases} \tag{11.28}$$

where $x_e = (R^\top \otimes I)x \in \mathbb{R}^{(n-1)\ell}$, $F_e = R^\top F$, and $H_e = HRWL_e^{-1}$. A dual edge system is also introduced with a different realization as

$$\mathbf{\Sigma}_f : \begin{cases} \dot{x}_f = (I_{n-1} \otimes A - L_e \otimes BC)x_f + (F_f \otimes B)u, \\ y_e = (H_f \otimes C)x_f, \end{cases} \tag{11.29}$$

with $x_f = (L_e^{-1} \otimes I)x_e$, $F_f = L_e^{-1}F_e$, and $H_f = HRW$.

Assuming that (A, B, C) is passive and minimal, the system (11.13) achieves synchronization from Theorem 2, which means that $A - \lambda_k BC$ is Hurwitz for all nonzero eigenvalues λ_k of graph Laplacian matrix L. This further implies that both $\mathbf{\Sigma}_e$ and $\mathbf{\Sigma}_f$ are asymptotically stable. As a result, generalized controllability and observability Gramians of the edge systems (11.28) and (11.29) can be analyzed.

Lemma 4. [8] *Consider the edge systems* (11.28) *and* (11.29) *of a tree network. There exist matrices* $X > 0$ *and* $Y > 0$ *such that the following inequalities hold:*

$$-L_e X - XL_e^\top + R^\top FF^\top R \le 0, \tag{11.30}$$

$$-L_e^\top Y - YL_e + WR^\top H^\top HRW \le 0. \tag{11.31}$$

Moreover, $P_e := X \otimes K^{-1}$ *and* $Q_f := Y \otimes K$ *are a generalized controllability Gramian of* $\mathbf{\Sigma}_e$ *in* (11.28) *and a generalized observability Gramian of* $\mathbf{\Sigma}_f$ *in* (11.29)*, respectively, where* K *satisfies* (11.24) *for the passive subsystems.*

According to [8], the matrices X and Y can be chosen to admit a diagonal structure:

$$X = \operatorname{diag}(\xi_1, \xi_2, \ldots, \xi_{n-1}), \quad Y = \operatorname{diag}(\eta_1, \eta_2, \ldots, \eta_{n-1}), \tag{11.32}$$

where the ordering $\xi_i\eta_i \ge \xi_{i+1}\eta_{i+1}$ is imposed. Note that X and Y imply the controllability and observability properties of the edges, respectively, and the value of $\xi_i\eta_i$ can be viewed as an indication for the importance of the i-th edge. Following a similar reasoning as balanced truncation in Section 11.2.1, removing the edges according to the value of $\xi_i\eta_i$ is meaningful. In [8], a graph clustering procedure is presented to recursively aggregate the two vertices connected by the least important edge. Furthermore, an a priori upper bound on the approximation error in terms of the $\mathcal{H}_\infty$-norm can be derived.

Theorem 4. *Consider the networked system in* (11.14) *with* $M = I_n$*. Assume each subsystem is minimal and passive, and the underlying graph is a tree. Let* (11.18) *be the* r*-th-order reduced network system obtained by aggregating the vertices connecting by the least important edges of the original network. Then, the following error bound holds:*

$$\|G(s) - \hat{G}(s)\|_{\mathcal{H}_\infty} \le 2\left(\sum_{i=r}^{n-1} [L_e^{-1}]_{ii} \sqrt{\xi_i\eta_i}\right), \tag{11.33}$$

where $G(s)$ *and* $\hat{G}(s)$ *are transfer matrices in* (11.23)*,* $[L_e^{-1}]_{ii}$ *denotes the* i*-th diagonal entry of the matrix* L_e^{-1}*, and* ξ_i *and* η_i *are the diagonal entries of* X *and* Y *in* (11.32)*, respectively.*

Note that the proposed method in [8] heavily relies on the assumption of tree topology. For networks with general topology, applying this method would be challenging, since there may not exist edge systems as in (11.28) and (11.29), which admit diagonal Gramians as in (11.32).

11.3.3 Dissimilarity-based clustering

The methods in Section 11.3.1 and Section 11.3.2 rely either on a special graph clustering or on a specific topology. In this section, we review a dissimilarity-based method, which can be performed to reduce more general network systems. Clustering of data points in data science is usually based on some similarity measure in terms of vector norms. To cluster a dynamical network, we can extend the concept of *dissimilarity* using the function norms, which serves as a metric for quantifying how differently two distinct vertices (subsystems) behave [11, 13].

Definition 3. Consider a network system in (11.13) or (11.14). The dissimilarity between vertices i and j is defined as

$$\mathcal{D}_{ij} := \|\eta_i(s) - \eta_j(s)\|_{\mathcal{H}_2}, \tag{11.34}$$

where $\eta_i(s) := (\mathbf{e}_i^\top \otimes C)[M \otimes (sI_\ell - A) + L \otimes BC](F \otimes B)$ if (11.13) is considered and $\eta_i(s) := \mathbf{e}_i^\top (sM + L)^{-1}F$ if (11.14) is considered.

The transfer matrix $\eta_i(s)$ is the mapping from the external control signal u to the output of the i-th subsystem, y_i, and thus $\eta_i(s)$ is interpreted as the behavior of the i-th vertex with respect to the external inputs. The concept of dissimilarity indicates how different two vertices are in terms of their behaviors. It is verified in [13] that if the network system (11.13) is synchronized, $\mathcal{D}_{ij}$ in (11.34) is well-defined, and a dissimilarity matrix $\mathcal{D} \in \mathbb{R}^{n\times n}$ with $[\mathcal{D}]_{ij} = \mathcal{D}_{ij}$ is symmetric and with zero diagonal elements and nonnegative off-diagonal entries. However, it could be a formidable task to compute the dissimilarity between each pair of vertices in a large-scale network based on its definition. Next, we discuss efficient methods for computing dissimilarity $\mathcal{D}_{ij}$.

First, we consider the single-integrator network in (11.14), which is a semi-stable system. Following Section 11.2.2, the pseudo-controllability Gramian of (11.14) is computed as $\mathcal{P} = \mathcal{J}\tilde{\mathcal{P}}\mathcal{J}^\top$, where $\tilde{\mathcal{P}}$ is an arbitrary solution of

$$-M^{-1}L\tilde{\mathcal{P}} - \tilde{\mathcal{P}}LM^{-1} + (I - \mathcal{J})M^{-1}FF^\top M^{-1}(I - \mathcal{J}) = 0, \quad \mathcal{J} := \frac{\mathbb{1}\mathbb{1}^\top M}{\mathbb{1}^\top M \mathbb{1}}. \tag{11.35}$$

We refer to [15, 20] for more details. Note that Theorem 1 implies that the transfer function error $\eta_i(s) - \eta_j(s)$ is in the $\mathcal{H}_2$-space for any nodes i and j in the network, and an efficient method for computing $\mathcal{D}$ is presented based on the pseudo-controllability Gramian:

$$\mathcal{D}_{ij} = \sqrt{(\mathbf{e}_i - \mathbf{e}_j)^\top \mathcal{P}(\mathbf{e}_i - \mathbf{e}_j)}. \tag{11.36}$$

Next, we consider the network system (11.13) which achieves synchronization. If the overall system (11.13) is semi-stable, we can still apply pseudo-Gramians to compute dissimilarity. However, the subsystems in the network may be unstable. In this case, we present another computation approach [13]. Denote

$$\mathcal{S} := \begin{bmatrix} -I_{n-1} \\ \mathbb{1}_{n-1}^\top \end{bmatrix} \in \mathbb{R}^{n\times(n-1)}, \quad \mathcal{S}^\dagger = \left(\mathcal{S}^\top M^{-1}\mathcal{S}\right)^{-1}\mathcal{S}^\top M^{-1}, \tag{11.37}$$

which satisfy $\mathcal{S}\mathbb{1} = 0$, $\mathcal{S}^\dagger M\mathbb{1} = 0$, and $\mathcal{S}^\dagger\mathcal{S} = I_{n-1}$. Let

$$\mathcal{A} := I_{n-1}\otimes A - \mathcal{S}^\dagger LM^{-1}\mathcal{S}\otimes BC, \quad \mathcal{B} = \mathcal{S}^\dagger F\otimes B,$$

where $\mathcal{A}$ is Hurwitz if and only if the system (11.13) achieves synchronization.

Theorem 5. *Let the network system* (11.13) *achieve synchronization. Then, there exists a symmetric matrix $\bar{\mathcal{P}} \in \mathbb{R}^{(n-1)\ell\times(n-1)\ell}$, which is the unique solution of the Lyapunov equation $\bar{A}\bar{\mathcal{P}} + \bar{\mathcal{P}}\bar{A} + \bar{\mathcal{B}}\bar{\mathcal{B}}^\top = 0$. Moreover,*

$$\mathcal{D}_{ij} = \sqrt{\mathrm{Tr}\left(\Psi_{ij}\bar{\mathcal{P}}\Psi_{ij}^\top\right)}, \tag{11.38}$$

where $\Psi_{ij} := (\mathbf{e}_i - \mathbf{e}_j)^\top M\mathcal{S}\otimes C$.

The definition of pairwise dissimilarity in (11.34) measures how close two subsystems behave, and aggregating vertices with similar behaviors potentially leads to a small approximation error. Having dissimilarity as a metric, clustering algorithms for static graphs in, e. g., [45, 71] can be also adopted to solve the model reduction problem for dynamical networks. For instant, a *hierarchical clustering* algorithm is applied in [12] as in Algorithm 11.1.

An iterative approach for single-integrator networks can be found in [11], and an alternative clustering method is presented in [63], which takes into account the connectedness of vertices such that the vertices in each cluster form a connected graph.

In Algorithm 11.1, the proximity of two clusters $\mathcal{C}_\mu$ and $\mathcal{C}_\nu$ is evaluated by (11.39), which means the average dissimilarity of the vertices in the two clusters. Other metrics of cluster proximity can be used as well. For instance, we can take the smallest dissimilarity of the vertices from two clusters, or the largest dissimilarity of the nodes from two clusters. The proximity of two clusters allows us to link pairs of clusters with smaller proximity and place them into binary clusters. Then, the newly formed clusters can be grouped into larger ones according to the cluster proximity. In each loop, two clusters with the lowest proximity are merged together, and finally a binary hierarchical tree, called *dendrogram*, that visualizes this process can be generated; see Figure 11.2 in the following example.

Algorithm 11.1 Hierarchical clustering algorithm.

1: Compute the dissimilarity matrix $\mathcal{D}$.
2: Place each node into a singleton cluster, i. e., $\mathcal{C}_i \leftarrow \{i\},\ \forall\, 1 \le i \le n$.
3: Find two clusters $\mathcal{C}_k$ and $\mathcal{C}_l$ such that

$$(k, l) := \arg\min\left(\frac{1}{|\mathcal{C}_k| \cdot |\mathcal{C}_l|} \sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_l} \mathcal{D}_{ij}\right). \tag{11.39}$$

4: Merge clusters $\mathcal{C}_k$ and $\mathcal{C}_l$ into a single cluster.
5: Repeat steps 3 and 4 until r clusters are obtained.
6: Compute the characteristic matrix $\Pi \in \mathbb{R}^{n \times r}$ and return

$$\hat{M} \leftarrow \Pi^\top M \Pi, \hat{L} \leftarrow \Pi^\top L \Pi, \hat{F} \leftarrow \Pi^\top F.$$

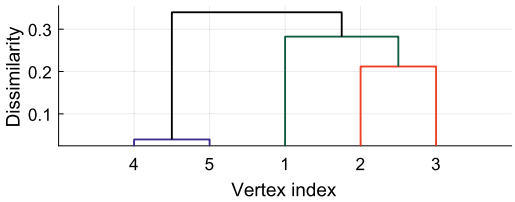


Figure 11.2: Dendrogram illustrating the hierarchical clustering of the networked mass–damper system. The horizontal axis is labeled by vertex numberings, while the vertical axis represents the dissimilarity of clusters. The dissimilarity is measured in the $\mathcal{H}_2$-norm, and the level at which branches merge indicates the dissimilarity between two clusters.

Example 2. Consider the networked mass–damper system in Example 1. The dissimilarity matrix can be computed using either (11.36) or (11.38), which yields

$$\mathcal{D} = \begin{bmatrix} 0 & 0.2494 & 0.3154 & 0.3919 & 0.4142 \\ 0.2494 & 0 & 0.2119 & 0.3688 & 0.3842 \\ 0.3154 & 0.2119 & 0 & 0.2410 & 0.2394 \\ 0.3919 & 0.3688 & 0.2410 & 0 & 0.0396 \\ 0.4142 & 0.3842 & 0.2394 & 0.0396 & 0 \end{bmatrix}.$$

The minimal value is 0.0396, indicating that vertices 4 and 5 have the most similar behavior compared to the other pairs of vertices. Thus, vertices 4 and 5 are first aggregated, which leads to clusters: $\{\{1\}, \{2\}, \{3\}, \{4, 5\}\}$. In the hierarchical clustering, we check the proximities of the clusters by (11.39) and then obtain a coarser clustering $\{\{1\}, \{2, 3\}, \{4, 5\}\}$. This process can be continued until we have generated a dendrogram as depicted in Figure 11.2.

Algorithm 11.1 is based on pairwise dissimilarities of the vertices and minimizes within-cluster variances. The variance within a cluster can be characterized by the largest dissimilarity between all pairs of vertices within the cluster, which leads to an upper bound on the $\mathcal{H}_2$-approximation error [13].

Theorem 6. *Consider the network system* (11.13) *with the output matrix* $H = I_n$. *Let* $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ *be the graph clustering of the network, and let* $G(s)$ *and* $\hat{G}(s)$ *denote the transfer matrices defined in* (11.23). *If* A *in* (11.11) *satisfies* $A + A^\top < 0$, *then we have*

$$\left\|G(s) - \hat{G}(s)\right\|_{\mathcal{H}_2} < \gamma \cdot \sum_{k=1}^{r} |\mathcal{C}_k| \cdot \max_{i,j \in \mathcal{C}_k} \mathcal{D}_{ij}, \tag{11.40}$$

where $\gamma \in \mathbb{R}_+$ *only depends on the original system* (11.13) *and satisfies*

$$\begin{bmatrix} I \otimes (A^\top + A) - L \otimes (C^\top B^\top + BC) & L \otimes BC & -I \otimes C^\top \\ L \otimes C^\top B^\top & -\gamma I & I \\ -I \otimes C & I & -\gamma I \end{bmatrix} < 0. \tag{11.41}$$

If the considered network system is in the form of (11.14), we further obtain an error bound based on the pseudo-controllability Gramian.

Proposition 1. [20] *Let* S *and* $\hat{S}$ *in* (11.22) *be the transfer matrices of* (11.14) *and* (11.19), *respectively. We have*

$$\|S - \hat{S}\|_{\mathcal{H}_2} \le \gamma_s \sqrt{\mathrm{Tr}(I - \Pi\Pi^\dagger)\mathcal{P}(I - \Pi\Pi^\dagger)^\top}, \tag{11.42}$$

where $\Pi^\dagger = (\Pi^\top M \Pi)^{-1}\Pi^\top M$ *and* $\mathcal{P}$ *is the pseudo-controllability Gramian of* (11.14). *The constant* $\gamma_s \in \mathbb{R}_+$ *is a solution of*

$$\begin{bmatrix} MLM^{-1} + M^{-1}LM & M^{-1}L & (I - \mathcal{J}^\top)H^\top \\ LM^{-1} & -\gamma_s I & H^\top \\ H(I - \mathcal{J}) & H & -\gamma_s I \end{bmatrix} \le 0, \tag{11.43}$$

with $\mathcal{J}$ *defined in* (11.35).

The core step in dissimilarity-based clustering is to properly define the dissimilarity of dynamical vertices. For linear time-variant networks, nodal dissimilarity can be always defined as the transfer from the external inputs to the vertex states. This mechanism of dissimilarity-based clustering is applicable to different types of dynamical networks; see, e. g., [12, 19, 19, 17] for more results on second-order networks, directed networks, and controlled power networks. For nonlinear networks, DC gain, a function of input amplitude, can be considered [48], in which model reduction aggregates state variables having similar DC gains.

11.3.4 Edge weighting approach

Generally, all the existing clustering-based reduction methods fall into the framework of Petrov–Galerkin projections. In [25, 24], an $\mathcal{H}_2$-optimal approach is presented, which does not aim to find a suitable graph clustering. Instead, this approach focuses on how to construct a "good" reduced-order model for a given clustering. To formulate this problem, the topology of a reduced network can be obtained from the given clustering, while all the edge weights are free parameters to be determined via optimization algorithms.

Consider the original network system in (11.14) with graph $\mathcal{G}$. Let $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ be a given graph clustering of $\mathcal{G}$. Then, a *quotient graph* $\hat{\mathcal{G}}$ is an r-vertex directed graph obtained by aggregating all the vertices in each cluster as a single vertex, while retaining connections between clusters and ignoring the edges within each cluster. If there is an edge $(i, j) \in \mathcal{G}$ with vertices i, j in the same cluster, then this edge will be ignored in $\hat{\mathcal{G}}$. However, if the edge (i, j) satisfies $i \in \mathcal{C}_k$ and $j \in \mathcal{C}_l$, then there will be an edge (k, l) in $\hat{\mathcal{G}}$. The incidence matrix $\hat{R}$ of the quotient graph $\hat{\mathcal{G}}$ can be obtained by removing all the zero columns of $\Pi^\top R$, where R is the incidence matrix of $\mathcal{G}$, and Π is the characteristic matrix of the clustering. Denote

$$\hat{W} = \operatorname{diag}(\hat{w}), \quad \text{with } \hat{w} = \begin{bmatrix} \hat{w}_1 & \hat{w}_2 & \cdots & \hat{w}_\kappa \end{bmatrix}^\top, \tag{11.44}$$

as the edge weight matrix of $\hat{\mathcal{G}}$, where $\hat{w}_k \in \mathbb{R}_+$ and κ denotes the number of edges in $\hat{\mathcal{G}}$. Then, a parameterized model of a reduced-order network is obtained:

$$\begin{cases} \hat{M}\dot{z} = -\hat{R}\hat{W}\hat{R}^\top z + \hat{F}u, \\ \hat{y} = \hat{H}z, \end{cases} \tag{11.45}$$

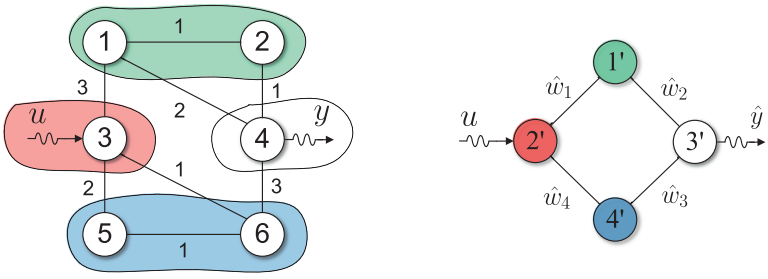
where $\hat{M} = \Pi^\top M \Pi$, $\hat{F} = \Pi^\top F$, and $\hat{H} = H\Pi$. The edge weight matrix $\hat{W}$ is the only unknown to be determined. Let

$$S_p := \hat{H}\left(s\hat{M} + \hat{R}\hat{W}\hat{R}^\top\right)^{-1}\hat{F}. \tag{11.46}$$

Then, an optimization problem can be formulated to minimize the approximation error $\|S - S_p\|_{\mathcal{H}_2}$ by tuning the edge weights. Here, an example is used to demonstrate the parameterized modeling of a reduced network system.

Example 3. Consider an undirected graph composed of six vertices in Figure 11.3a. An external force u is acting on vertex 3, and the state of vertex 4 is measured as the output y. Given a clustering with $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \{3\}$, $\mathcal{C}_3 = \{4\}$, $\mathcal{C}_4 = \{5, 6\}$, the quotient graph is obtained in Figure 11.3b with the incidence matrix

$$\hat{R} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}.$$



(a) (b)

Figure 11.3: (a) An undirected network consisting of six vertices, in which vertex 3 is the leader and vertex 4 is measured. Four clusters are indicated by different colors. (b) A quotient graph obtained by clustering.

Let $\hat{W} = \mathrm{diag}(\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4)$ be the weights of the corresponding edges. The Laplacian matrix of the reduced network is constructed as

$$\hat{R}\hat{W}\hat{R}^\top = \begin{bmatrix} \hat{w}_1 + \hat{w}_2 & -\hat{w}_1 & -\hat{w}_2 & 0 \\ -\hat{w}_1 & \hat{w}_1 + \hat{w}_4 & 0 & -\hat{w}_4 \\ -\hat{w}_2 & 0 & \hat{w}_2 + \hat{w}_3 & -\hat{w}_3 \\ 0 & -\hat{w}_4 & -\hat{w}_3 & \hat{w}_3 + \hat{w}_4 \end{bmatrix},$$

and moreover, we have $\hat{F} = \Pi^\top F = [0\,1\,0\,0]^\top$ and $\hat{H} = H\Pi = [0\,0\,1\,0]$. If in the original network, $M = I_6$, in the reduced-order model (11.45), $\hat{M} = \Pi^\top M \Pi = \mathrm{diag}(2, 1, 1, 2)$.

An optimization technique based on the *convex-concave decomposition* can be applied to search for a set of optimal weights iteratively. Before proceeding, a necessary and sufficient condition for characterizing $\|G_e(s)\|_{\mathcal{H}_2}$ is shown.

Theorem 7. *Given the network system* (11.14). *A reduced-order model in* (11.45) *satisfies* $\|S - S_p\|_{\mathcal{H}_2}^2 < \gamma_p$ *if and only if there exist matrices* $\hat{Q} = \hat{Q}^\top > 0$, $\hat{Z} = \hat{Z}^\top > 0$, *and* $\hat{\delta} \in \mathbb{R}_+$ *such that* $\mathrm{Tr}(\hat{Z}) < \gamma_p$, *and*

$$\begin{bmatrix} \hat{Q}\bar{A} + \bar{A}^\top \hat{Q} & \hat{Q}B_e & \hat{Q}E \\ B_e^\top \hat{Q} & -\hat{\delta}I & 0 \\ E^\top \hat{Q} & 0 & 0 \end{bmatrix} + \begin{bmatrix} -\bar{A}_r^\top \bar{A}_r & 0 & \bar{A}_r^\top \\ 0 & 0 & 0 \\ \bar{A}_r & 0 & -I \end{bmatrix} < 0, \tag{11.47}$$

$$\begin{bmatrix} \hat{Q} & \hat{\delta}C_e^\top \\ \hat{\delta}C_e & \hat{Z} \end{bmatrix} > 0, \tag{11.48}$$

where

$$\bar{A} = \begin{bmatrix} -\mathcal{S}_n^+ L M^{-1} \mathcal{S}_n & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{A}_r = \begin{bmatrix} 0 & -\mathcal{S}_r^+ \hat{R}\hat{W}\hat{R}^\top \hat{M}^{-1} \mathcal{S}_r \\ 0 & 0 \end{bmatrix}, \quad B_e = \begin{bmatrix} \mathcal{S}_n^+ F \\ \mathcal{S}_r^+ \hat{F} \end{bmatrix},$$

$$C_e = [HM^{-1}\mathcal{S}_n \quad -\hat{H}\hat{M}^{-1}\mathcal{S}_r], \quad E = \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix}, \quad \mathcal{S}_n = \begin{bmatrix} -I_{n-1} \\ \mathbb{1}_{n-1}^\top \end{bmatrix}, \quad \mathcal{S}_r = \begin{bmatrix} -I_{r-1} \\ \mathbb{1}_{r-1}^\top \end{bmatrix}.$$

Based on Theorem 7, the edge weighting problem is formulated as a minimization problem:

$$\min_{\hat{Q}>0,\ \hat{W}} \operatorname{Tr}(Z), \quad \text{s.t.} \quad (11.47) \text{ and } (11.48) \text{ hold}, \tag{11.49}$$

where $\hat{Z} = \hat{\delta}Z$. Consider the matrix-valued mapping

$$\Phi(\hat{Q}, \hat{\delta}, \hat{W}) = \psi(\hat{Q}, \hat{\delta}) + \varphi(\hat{W}), \tag{11.50}$$

where

$$\psi(\hat{Q}, \hat{\delta}) = \begin{bmatrix} \hat{Q}\bar{A} + \bar{A}^\top\hat{Q} & \hat{Q}B_e & \hat{Q}E \\ B_e^\top\hat{Q} & -\hat{\delta}I & 0 \\ E^\top\hat{Q} & 0 & 0 \end{bmatrix}, \quad \varphi(\hat{W}) = \begin{bmatrix} -\bar{A}_r^\top\bar{A}_r & 0 & \bar{A}_r^\top \\ 0 & 0 & 0 \\ \bar{A}_r & 0 & -I \end{bmatrix}.$$

Then, the pair $(\psi, -\varphi)$ is a psd-convex-concave decomposition of Φ [24]. The bilinear matrix inequality (11.47) with the nonlinearity term $\bar{A}_r^\top\bar{A}_r$ can be handled using such a decomposition, which can linearize the optimization problem (11.49) at a stationary point $\hat{W}$ [31]. Rewrite $\varphi(\hat{W})$ in (11.50) as $\phi(\hat{w}) = \varphi(\hat{W})$, with $\hat{w} \in \mathbb{R}_+^\kappa$ defined in (11.44). Given a point $\hat{w}^{(k)}$, the linearized formulation of the problem (11.49) at $\hat{w}^{(k)}$ is formulated as a *convex* problem:

$$\begin{aligned} \min_{\hat{Q}>0, \hat{w}\in\mathbb{R}_+^\kappa} \ & f(\hat{w}) = \operatorname{Tr}(Z) \\ \text{s.t.} \ & \begin{bmatrix} \hat{Q} & \hat{\delta}C_e^\top \\ \hat{\delta}C_e & \hat{Z} \end{bmatrix} > 0, \ \hat{\delta} \in \mathbb{R}_+, \ \hat{Z} = \hat{\delta}Z > 0, \\ & \psi(\hat{Q}, \hat{\delta}) + \varphi(\hat{W}^{(k)}) + D\phi(\hat{w}^{(k)})[\hat{w} - \hat{w}^{(k)}] < 0, \end{aligned} \tag{11.51}$$

where the derivative of $\phi(\hat{w}^{(k)})$ is defined as

$$D\phi(\hat{w}^{(k)})[\hat{w} - \hat{w}^{(k)}] := \sum_{i=1}^{\kappa} (\hat{w}_i - \hat{w}_i^{(k)}) \frac{\partial \phi}{\partial \hat{w}_i^{(k)}}(\hat{w}^{(k)}).$$

Then, an algorithmic approach is presented in Algorithm 11.2 for solving the minimization problem in (11.49) in an iterative fashion.

If $\hat{w}$ is initialized as the outcome of clustering-based projection methods, the approximation accuracy obtained by the edge weighting approach will be better than the ones obtained by clustering-based projection after iteration. Furthermore, to solve the optimization problem in (11.49), we can also use a cross-iteration algorithm presented in [25].

Algorithm 11.2 Iterative edge weighting.

1: Choose an initial vector $\hat{w}^{(0)} \in \mathbb{R}_+^{\kappa}$.
2: Set iteration step: $k \leftarrow 0$.
3: **repeat**
4: Solve (11.51) to obtain the optimal solution $\hat{w}^*$.
5: $k \leftarrow k+1$, and $\hat{w}^{(k)} \leftarrow \hat{w}^*$.
6: **until** $|f(\mu^{(k+1)}) - f(\mu^{(k)})| \leq \varepsilon$, with ε a prefixed error tolerance.
7: Return $\hat{W}^* \leftarrow \mathrm{diag}(\hat{w}^*)$.

11.3.5 Other clustering-based methods

In this section, several other model reduction schemes based on graph clustering are reviewed. The method in [14] formulates the clustering-based model reduction as a nonconvex optimization problem with mixed-binary variables Π and the objective function to minimize the $\mathcal{H}_2$-norm of the approximation error. The error system between (11.14) and (11.19) is defined, of which the controllability and the observability Gramians are used to derive an explicit expression for the gradient of the objective function. Then a projected gradient algorithm can be employed to solve this optimization problem with mathematical guarantees on its convergence. Related to the work in [58], a combination of the Krylov subspace method with graph clustering is proposed in [54], where a reduced basis is firstly found by the iterative rational Krylov algorithm, and then a graph partition is obtained by the QR decomposition with column pivoting on the projection matrix. An alternative graph-based model reduction method is proposed in [49], which finds a graph clustering based on the edge agreement protocol of a network (see the definition in [79]) and provides a greedy contraction algorithm as a suboptimal solution of graph clustering. The clustering and aggregation approach in [30, 29] is based on spectral analysis of Markov chains. The Kullback–Leibler divergence rate is employed as a metric to measure the difference between the original network model and its approximation.

Clustering-based model reduction approaches are also found in the applications of other types of networks, i. e., network systems that do not reach consensus. Instead, other network properties are emphasized. For instance, [51] proposes a reduction method for scale-free networks, which are networks whose degree distribution follows a power law. They are roughly characterized by the presence of few vertices with a large degree (number of connections) and a large number of verities with small degree. The method in [51] preserves the eigenvector centrality of the adjacency matrix of the original network such that the obtained reduced network remains scale-free.

Positive networks are considered in [42, 41]. A single-input bidirectional positive network is given in [42] as

$$\dot{x} = Ax + bu, \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}, \tag{11.52}$$

where $b \in \mathbb{R}^p$ and $A := -D - L$, with $D \geq 0$ being a diagonal matrix (i. e., at least one diagonal entry of D should be positive and the rest of the diagonal entries are zero) and $L \geq 0$ being a Laplacian matrix representing an undirected connected graph. It is verified that A is negative definite, and thus the system (11.52) is asymptotically stable. The structure of A can be interpreted as a network containing self-loops. In [42], a set of clusters is constructed based on the notion of cluster reducibility, which characterizes the uncontrollability of local state variables. By aggregating the reducible clusters, a reduced-order model is obtained that preserves the stability and positivity. The work in [41] extends this method to the directed case, where A in (11.52) is now assumed to be irreducible, Metzler, and semi-stable. In this case, the Frobenius eigenvector of A is used for constructing the projections such that both semi-stability and positivity are preserved in the resulting reduced-order network model. In both [42] and [41], an upper bound on the approximation error is established using the cluster reducibility, and then a clustering scheme is proposed to select suitable clusters, aiming at minimizing the a posteriori bound on the reduction error.

11.4 Balanced truncation of network systems

Reducing the dimension of each subsystem also results in a simplification of overall networks. To reduce the dynamics of vertices, balanced truncation based on generalized Gramian matrices is commonly used (see, e. g., [57, 23, 21]), in which preserving the synchronization property of the overall network is of particular interest. In this section, we review some recent results in the synchronization-preserving model reduction of large-scale network systems using the classic generalized balanced truncation. For simplicity, we assume $M = I_n$ in (11.13) throughout this section.

11.4.1 Model reduction of subsystems in networks

Starting from a synchronized network system in (11.13), the aim of this subsection is to derive a network model with reduced-order subsystems such that synchronization is preserved in the reduced-order network in (11.18).

If each subsystem in (11.11) is asymptotically stable, we might apply standard balanced truncation to reduce the dimension of the subsystem regardless of their interconnection structure. However, this reduction is possible to destroy the property of the overall network system (11.13), e. g., stability and synchronization. To achieve synchronization preservation, [57] adopts a sufficient small gain type of condition to guarantee synchronization of (11.13).

Lemma 5. *Denote by* $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ *the eigenvalues of the Laplacian matrix L. The network system* (11.13) *achieves synchronization if there exists a nonzero eigenvalue*

$\lambda \in \{\lambda_2, \ldots, \lambda_n\}$ such that $A - \lambda BC$ is Hurwitz and there exists a positive definite matrix K satisfying the Riccati inequality

$$(A - \lambda BC)^\top K + K(A - \lambda BC) + C^\top C + \left(\frac{\delta}{\gamma}\right)^2 KBB^\top K < 0, \tag{11.53}$$

where $\delta := \max\{\lambda - \lambda_2, \lambda_n - \lambda\}$.

It is worth noting that (11.53) is equivalent to the small gain condition

$$\left\| C(sI_\ell - A + \lambda BC)^{-1} B \right\|_{\mathcal{H}_\infty} < \frac{\delta}{\gamma}.$$

Let K_m and K_M be the minimal and maximal real symmetric solutions of (11.53). Then K_M^{-1} and K_m can be regarded as a pair of generalized Gramians of the system $(A + \lambda BC, \frac{\delta}{\gamma}B, C)$. Applying the generalized balanced truncation introduced in Section 11.2.1, a reduced-order model $(\hat{A} + \lambda\hat{B}\hat{C}, \frac{\delta}{\gamma}\hat{B}, \hat{C})$ with $\hat{A} \in \mathbb{R}^{k\times k}$ is obtained such that the small gain condition $\|\hat{C}(sI_\ell - \hat{A} + \lambda\hat{B}\hat{C})^{-1}\hat{B}\|_{\mathcal{H}_\infty} < \frac{\delta}{\gamma}$ is retained. Therefore, the following reduced-order network model preserves the synchronization property:

$$\begin{cases} \dot{\xi} = (I_n \otimes \hat{A} - L \otimes \hat{B}\hat{C})\xi + (F \otimes \hat{B})u, \\ \eta = (H \otimes \hat{C})\xi. \end{cases} \tag{11.54}$$

Theorem 8. *Consider a network system* (11.13) *that satisfies the synchronization condition in Lemma* 5. *Then, the reduced-order network model in* (11.54) *obtained by generalized balanced truncation using K_M^{-1} and K_m achieves synchronization.*

Moreover, similar to (11.25), we assume a particular output $y = (W^{\frac{1}{2}}R^\top \otimes C)x$. Then the error system between (11.13) and (11.54) is stable. We denote $\tilde{G}(s)$ as the transfer matrix of system (11.54), and the model reduction error is upper-bounded as

$$\left\| G(s) - \tilde{G}(s) \right\|_{\mathcal{H}_\infty} \le \frac{2\gamma\sqrt{\lambda_n}}{\delta(1-\gamma^2)} \sum_{i=k+1}^{\ell} \sigma_i, \tag{11.55}$$

where σ_i are the GHSVs computed using K_M^{-1} and K_m [57].

Inspired by the work [57] for linear networks, [18, 23] consider dynamical networks of diffusively interconnected nonlinear Lur'e subsystems. The robust synchronization of the Lur'e network can be characterized by a linear matrix inequality (LMI). Different from [57, 18], where the minimum and maximum solutions of the LMI are used as generalized Gramians, [23] suggests to only use the solution of the LMI with the minimal trace as a generalized controllability Gramian, while the observability counterpart is taken by the standard observability Gramian as the solution of the Lyapunov equation, which is less conservative than the LMI. Using such a pair, generalized balanced truncation is performed on the linear component of each Lur'e subsystem, and the resulting reduced-order network system is still guaranteed to have the robust synchronization property.

11.4.2 Simultaneously reduction of network structure and subsystems

In the line of works [42, 41], a reduction method for network systems composed of higher-dimensional dissipative subsystems is presented in [43], where the subsystems are reduced via block-diagonal orthogonal projection, while the network structure is simplified using clustering. In [16], the balancing method, for the first time, is applied for reducing the interconnection structure of networks with diffusively coupled vertices, and more extensions are found in [78, 77] based on eigenvalue assignment and moment-matching. In [21], the idea in [16] is further developed and applied to general networks of the form (11.13). The proposed approach can reduce the complexity of network structures and individual agent dynamics simultaneously via a unified framework.

Consider the network system $\mathbf{\Sigma}$ in (11.13), where each subsystem $\mathbf{\Sigma}_i$ as in (11.11) is passive, namely, there exists a positive definite K such that (11.24) holds. Note that L is singular and A in (11.11) is not necessarily Hurwitz, implying that the overall system $\mathbf{\Sigma}$ may be not asymptotically stable, and thus a direct application of balanced truncation to $\mathbf{\Sigma}$ is not feasible. The method in [21] starts off with a decomposition of $\mathbf{\Sigma}$ using a *spectral decomposition* of the graph Laplacian:

$$L = T\Lambda T^\top = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} \bar{\Lambda} & \\ & 0 \end{bmatrix} \begin{bmatrix} T_1^\top \\ T_2^\top \end{bmatrix}, \tag{11.56}$$

where $T_2 = \mathbb{1}_n/\sqrt{n}$ and $\bar{\Lambda} := \operatorname{diag}(\lambda_2, \lambda_3, \ldots, \lambda_n)$, with λ_i denoting the nonzero eigenvalues of L. Then, the system $\mathbf{\Sigma}$ can be split into two components, namely, an *average module*

$$\mathbf{\Sigma_a} : \begin{cases} \dot{z}_a = Az_a + \frac{1}{\sqrt{n}}(\mathbb{1}_n^\top F \otimes B)u, \\ y_a = \frac{1}{\sqrt{n}}(H\mathbb{1}_n \otimes C)z_a, \end{cases} \tag{11.57}$$

with $z_a \in \mathbb{R}^\ell$, and an asymptotically stable system

$$\mathbf{\Sigma_s} : \begin{cases} \dot{z}_s = (I_{n-1} \otimes A - \bar{\Lambda} \otimes BC)z_s + (\bar{F} \otimes B)u, \\ y_s = (\bar{H} \otimes C)z_s, \end{cases} \tag{11.58}$$

where $z_s \in \mathbb{R}^{(n-1)\times\ell}$, $\bar{F} = T_1^T F$, and $\bar{H} = HT_1$. The stability is guaranteed as the system $\mathbf{\Sigma}$ achieves synchronization (Theorem 2).

The model reduction procedure is as follows. First, we can apply balanced truncation to $\mathbf{\Sigma_s}$ to generate a lower-order approximation $\hat{\mathbf{\Sigma}}_\mathbf{s}$. It meanwhile gives a reduced subsystem $(\hat{A}, \hat{B}, \hat{C})$ resulting in a reduced-order average module $\hat{\mathbf{\Sigma}}_\mathbf{a}$. Combining $\hat{\mathbf{\Sigma}}_\mathbf{s}$ with $\hat{\mathbf{\Sigma}}_\mathbf{a}$ then formulates a reduced-order model $\tilde{\mathbf{\Sigma}}$ whose input–output behavior approximates that of the original system $\mathbf{\Sigma}$. However, at this stage, the network structure is

not necessarily preserved by $\tilde{\mathbf{\Sigma}}$. Then, it is desired to use a coordinate transformation to convert $\tilde{\mathbf{\Sigma}}$ to $\hat{\mathbf{\Sigma}}$, which restores the Laplacian structure. The whole procedure is summarized in Figure 11.4. There are two key problems here:

1. How can one retain the subsystem structure in $\hat{\mathbf{\Sigma}}_\mathbf{s}$ such that subsystem dynamics do not mix with the topological information?
2. How can one recover a network interpretation in the reduced-order model $\tilde{\mathbf{\Sigma}}$ via a coordinate transformation?

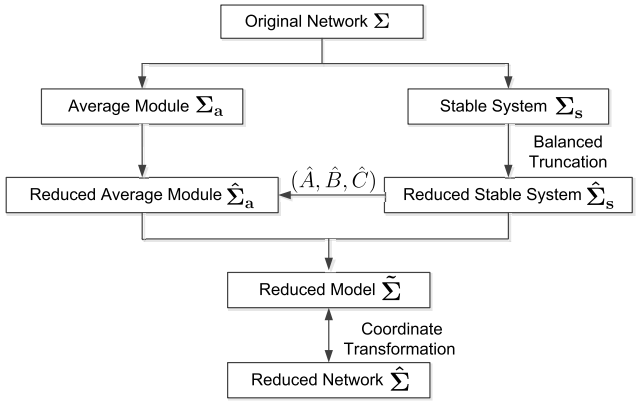


Figure 11.4: The model reduction scheme for networked passive systems, where the simplification of network structure and the reduction of subsystems are performed simultaneously.

To resolve the first problem, we resort to the balanced truncation approach based on generalized Gramians. Suppose $\bar{\Lambda}$ in (11.56) has s distinct diagonal entries ordered as $\bar{\lambda}_1 > \bar{\lambda}_2 > \cdots > \bar{\lambda}_s$. We rewrite $\bar{\Lambda}$ as $\bar{\Lambda} = \mathrm{blkdiag}(\bar{\lambda}_1 I_{m_1}, \bar{\lambda}_2 I_{m_2}, \ldots, \bar{\lambda}_s I_{m_s})$, where m_i is the multiplicity of $\bar{\lambda}_i$, and $\sum_{i=1}^{s} m_i = n - 1$. Then, the following Lyapunov equation and inequality have solutions X and Y:

$$-\bar{\Lambda}X - X\bar{\Lambda} + \bar{F}\bar{F}^\top = 0, \tag{11.59a}$$

$$-\bar{\Lambda}Y - Y\bar{\Lambda} + \bar{H}^\top\bar{H} \le 0, \tag{11.59b}$$

where $X = X^\top > 0$ and $Y := \mathrm{blkdiag}(Y_1, Y_2, \ldots, Y_s)$, with $Y_i = Y_i^\top > 0$ and $Y_i \in \mathbb{R}^{m_i \times m_i}$, for $i = 1, 2, \ldots, s$. The generalized controllability and observability Gramians of the stable system $\mathbf{\Sigma}_\mathbf{s}$ are characterized by the following theorem.

Theorem 9. *Let $X > 0$ be the unique solution of* (11.59a)*, and let $Y > 0$ be a solution of* (11.59b)*. Let $K_m > 0$ and $K_M > 0$ be the minimum and maximum solutions of* (11.24)*, respectively. Then the matrices*

$$\mathcal{X} := X \otimes K_M^{-1} \quad \text{and} \quad \mathcal{Y} := Y \otimes K_m \tag{11.60}$$

are a pair of generalized Gramians of the asymptotically stable system $\mathbf{\Sigma_s}$*. Moreover, there exist two nonsingular matrices* $T_{\mathcal{G}}$ *and* $T_{\mathcal{D}}$ *such that* $\mathcal{T} = T_{\mathcal{G}} \otimes T_{\mathcal{D}}$ *satisfies*

$$\mathcal{T}\mathcal{X}\mathcal{T}^{\top} = \mathcal{T}^{-T}\mathcal{Y}\mathcal{T}^{-1} = \Sigma_{\mathcal{G}} \otimes \Sigma_{\mathcal{D}}. \tag{11.61}$$

Here, $\Sigma_{\mathcal{G}} := \operatorname{diag}\{\sigma_1, \sigma_2, \dots, \sigma_{n-1}\}$ *and* $\Sigma_{\mathcal{D}} := \operatorname{diag}\{\tau_1, \tau_2, \dots, \tau_\ell\}$*, where* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n-1}$ *and* $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_\ell$ *are equal to the square roots of the eigenvalues of* XY *and* $K_M^{-1}K_m$*, respectively.*

The block-diagonal structure of Y will be crucial to guarantee that the reduced-order model, obtained by preforming balanced truncation on the basis of X and Y, can be interpreted as a network system again, as will be shown in Theorem 10 below.

Remark 2. By the duality between controllability and observability, we can also use $-\bar{\Lambda}X - X\bar{\Lambda} + \bar{F}\bar{F}^{\top} \leq 0$ and $-\bar{\Lambda}Y - Y\bar{\Lambda} + \bar{H}^{\top}\bar{H} = 0$ to characterize the pair X and Y for the balanced truncation, where now X is constrained to have a block-diagonal structure.

Selecting the pair of Gramians in (11.60) with the Kronecker product structure is meaningful, since they can be simultaneously diagonalized, (i. e., balanced) using transformations of the form $\mathcal{T} = T_{\mathcal{G}} \otimes T_{\mathcal{D}}$. Note that $T_{\mathcal{G}}$ and $T_{\mathcal{D}}$ are independently generated from (11.59) and (11.24). More precisely, $T_{\mathcal{G}}$ only balances the network structure, or the triplet $(\bar{\Lambda}, \bar{F}, \bar{H})$, while $T_{\mathcal{D}}$ only balances the agent dynamics, i. e., the triplet (A, B, C). Thus, the Laplacian dynamics and each subsystem (11.11) can be reduced independently, allowing the resulting reduced-order model to preserve a network interpretation as well as the passivity of subsystems.

Denote by $(\hat{\Lambda}_1, \hat{F}_1, \hat{H}_1)$ and $(\hat{A}, \hat{B}, \hat{C}) := \hat{\mathbf{\Sigma}}_i$ the reduced-order models of $(\bar{\Lambda}, \bar{F}, \bar{H})$ and (A, B, C), respectively, where $\hat{\Lambda}_1 \in \mathbb{R}^{(r-1)\times(r-1)}$, $\hat{F}_1 \in \mathbb{R}^{(r-1)\times p}$, $\hat{H}_1 \in \mathbb{R}^{q\times(r-1)}$, $\hat{A} \in \mathbb{R}^{k\times k}$, $\hat{B} \in \mathbb{R}^{k\times m}$, and $\hat{C} \in \mathbb{R}^{m\times k}$. Here, $k < \ell$ and $r < n$. Consequently, the reduced-order models of the average module (11.57) and the stable system (11.58) are constructed:

$$\hat{\mathbf{\Sigma}}_{\mathbf{a}} : \begin{cases} \dot{\hat{z}}_a = \hat{A}\hat{z}_a + \frac{1}{\sqrt{n}}(\mathbb{1}_n^{\top}F \otimes \hat{B})u, \\ \hat{y}_a = \frac{1}{\sqrt{n}}(H\mathbb{1}_n \otimes \hat{C})\hat{z}_a, \end{cases} \tag{11.62a}$$

$$\hat{\mathbf{\Sigma}}_{\mathbf{s}} : \begin{cases} \dot{\hat{z}}_s = (I_{r-1} \otimes \hat{A} - \hat{\Lambda}_1 \otimes \hat{B}\hat{C})\hat{z}_s + (\hat{F}_1 \otimes \hat{B})u, \\ \hat{y}_s = (\hat{H}_1 \otimes \hat{C})\hat{z}_s. \end{cases} \tag{11.62b}$$

Combining the reduced-order models $\hat{\mathbf{\Sigma}}_{\mathbf{a}}$ and $\hat{\mathbf{\Sigma}}_{\mathbf{s}}$, a lower-dimensional approximation of the overall system $\mathbf{\Sigma}$ is formulated as

$$\tilde{\mathbf{\Sigma}} : \begin{cases} \dot{\hat{z}} = (I_r \otimes \hat{A} - \Gamma \otimes \hat{B}\hat{C})\hat{z} + (\mathcal{F} \otimes \hat{B})u, \\ \hat{y} = (\mathcal{H} \otimes \hat{C})\hat{z}, \end{cases} \tag{11.63}$$

where

$$\Gamma = \begin{bmatrix} \hat{\Lambda}_1 & \\ & 0 \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} \hat{F}_1 \\ \frac{1}{\sqrt{n}}\mathbb{1}_n^{\top}F \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} \hat{H}_1 & \frac{1}{\sqrt{n}}H\mathbb{1}_n \end{bmatrix}.$$

Here, Γ is not yet a Laplacian matrix, but it has only one zero eigenvalue at the origin and all the other eigenvalues are positive real. To restore a network interpretation in the reduced-order model $\tilde{\mathbf{\Sigma}}$, the following theorem is provided in [21], which states that there exists a similarity transformation between Γ and an undirected graph Laplacian matrix.

Theorem 10. *A real square matrix* Γ *is similar to the Laplacian matrix* $\mathcal{L}$ *associated with an weighted undirected connected graph if and only if* Γ *is diagonalizable and has an eigenvalue at* 0 *with multiplicity* 1 *while all the other eigenvalues are real and positive.*

By Theorem 10, we find a reduced Laplacian matrix $\hat{L}$ which has the same spectrum as Γ, namely, there exists a nonsingular matrix $\mathcal{T}_n$ such that $\hat{L} = \mathcal{T}_n^{-1}\Gamma\mathcal{T}_n$. The matrix $\hat{L}$ characterizes a reduced connected undirected graph $\hat{\mathcal{G}}$, which contains r vertices. Applying a coordinate transform $\hat{z} = (\mathcal{T}_n \otimes I_r)\hat{x}$ to the system $\tilde{\mathbf{\Sigma}}$ in (11.63) yields a reduced-order network model

$$\hat{\mathbf{\Sigma}} : \begin{cases} \dot{\hat{x}} = (I_r \otimes \hat{A} - \hat{L} \otimes \hat{B}\hat{C})\hat{x} + (\hat{F} \otimes \hat{B})u, \\ \hat{y} = (\hat{H} \otimes \hat{C})\hat{x}, \end{cases} \tag{11.64}$$

with $\hat{F} = \mathcal{T}_n^{-1}\mathcal{F}$ and $\hat{H} = \mathcal{H}\mathcal{T}_n$. It can be verified that the reduced-order network system $\hat{\mathbf{\Sigma}}$ in (11.64) preserves synchronization. Moreover, denote the transfer matrices of $\mathbf{\Sigma}$, $\hat{\mathbf{\Sigma}}$, $\mathbf{\Sigma_s}$, $\hat{\mathbf{\Sigma}}_\mathbf{s}$, $\mathbf{\Sigma_a}$, and $\hat{\mathbf{\Sigma}}_\mathbf{a}$ by G, $\hat{G}$, T_s, $\hat{T}_s$, T_a, and $\hat{T}_a$, respectively. The approximation error can be analyzed as follows:

$$\begin{aligned} \|G - \hat{G}\|_{\mathcal{H}_\infty} &= \|(T_s + T_a) - (\hat{T}_s + \hat{T}_a)\|_{\mathcal{H}_\infty} \\ &\le \|T_s - \hat{T}_s\|_{\mathcal{H}_\infty} + \|T_a - \hat{T}_a\|_{\mathcal{H}_\infty}, \end{aligned} \tag{11.65}$$

in which an a priori upper bound on the reduction error of the stable system $\mathbf{\Sigma_s}$ is given as

$$\|T_s - \hat{T}_s\|_{\mathcal{H}_\infty} \le 2\sum_{i=r}^{n-1}\sum_{j=1}^{\ell}\sigma_i\tau_j + 2\sum_{i=1}^{r-1}\sum_{j=k+1}^{\ell}\sigma_i\tau_j, \tag{11.66}$$

with σ_i and τ_i being the diagonal entries of $\Sigma_\mathcal{G}$ and $\Sigma_\mathcal{D}$ in (11.61), respectively. Denote by S_i and $\hat{S}_i$ the transfer matrices of $\mathbf{\Sigma}_i$ and $\hat{\mathbf{\Sigma}}_i$, respectively. If $S_i - \hat{S}_i \in \mathcal{H}_\infty$, we obtain

$$\|T_a - \hat{T}_a\|_{\mathcal{H}_\infty} \le \frac{1}{n}\left\|H\mathbb{1}_n\mathbb{1}_n^\top F\right\|_2 \cdot \|S_i - \hat{S}_i\|_{\mathcal{H}_\infty}. \tag{11.67}$$

In several special cases, the a priori error bounds on $\|G - \hat{G}\|_{\mathcal{H}_\infty}$ in (11.65) can be obtained. The first case is when we only reduce the dimension of the network while the agent dynamics are untouched as in [8]. In this case, we obtain $\|T_a - \hat{T}_a\|_{\mathcal{H}_\infty} = 0$, which yields

$$\|G - \hat{G}\|_{\mathcal{H}_\infty} = \|T_s - \hat{T}_s\|_{\mathcal{H}_\infty} \le 2\sum_{i=r}^{n-1}\sum_{j=1}^{\ell}\sigma_i\tau_j. \tag{11.68}$$

The second case is when the average module is not observable from the outputs of the overall system $\mathbf{\Sigma}$ or not controllable by the external inputs. Consider

$$H\mathbb{1}_n = 0, \quad \text{or} \quad \mathbb{1}_n^\top F = 0. \tag{11.69}$$

Then, the approximation between $\mathbf{\Sigma}$ and $\hat{\mathbf{\Sigma}}$ is bounded by $\|G - \hat{G}\|_{\mathcal{H}_\infty} = \|T_s - \hat{T}_s\|_{\mathcal{H}_\infty}$, whose upper bound is given in (11.66). A special example of (11.69) can be found in [57, 58, 49], where the output matrix H in (11.13) is taken as in (11.25).

11.5 Conclusions

In this chapter, model reduction techniques for linear dynamical networks with diffusive couplings have been reviewed. There exists a vast amount of literature on this topic, and the reference list in this chapter is certainly not complete. For example, in [65, 28, 27, 32, 56], the approximation approaches are developed based on singular perturbation approximation and applied to reduce the complexity of chemical reaction networks and power networks. In [47], the interconnection topology is simplified by removing cycles in the network, and in, e. g., [10, 55, 48], preliminary results for reducing nonlinear dynamical networks are developed. Recently, a lot of interest is taken in the combination of network reduction and controller and observer designs. For example, [76] presents a linear quadratic Gaussian controller for large-scale dynamical networks using the clustering-based reduction, and [68, 62] proposes the average state observer based on reduced-order network models.

Generally speaking, order reduction methods for linear network systems have been extensively investigated. However, the approximation of complex network containing nonlinear couplings or subsystems is still challenging, and the existing results on nonlinear networks are far from satisfactory. Another challenge in this area is the order reduction of heterogeneous networks, i. e., network systems composed of nonidentical subsystems.

Bibliography

[1] J. Anderson, Y.-C. Chang, and A. Papachristodoulou, Model decomposition and reduction tools for large-scale networks in systems biology, *Automatica*, **47** (6) (2011), 1165–1174.
[2] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, vol. 6, SIAM, 2005.
[3] A. Astolfi, Model reduction by moment matching for linear and nonlinear systems, *IEEE Trans. Autom. Control*, **55** (10) (2010), 2321–2336.
[4] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, *Appl. Numer. Math.*, **43** (1–2) (2002), 9–44.

[5] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.

[6] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Silveira (eds.), *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, 2020.

[7] D. S. Bernstein and S. P. Bhat, Lyapunov stability, semistability, and asymptotic stability of matrix second-order systems, *J. Mech. Des.*, **117** (B) (1995), 145–153.

[8] B. Besselink, H. Sandberg, and K. H. Johansson, Clustering-based model reduction of networked passive systems, *IEEE Trans. Autom. Control*, **61** (10) (2016), 2958–2973.

[9] S. P. Bhat and D. S. Bernstein, Lyapunov analysis of semistability, in *Proceedings of the 1999 American Control Conference*, vol. 3, pp. 1608–1612, IEEE, 1999.

[10] E. Bıyık and M. Arcak, Area aggregation and time-scale modeling for sparse nonlinear networks, *Syst. Control Lett.*, **57** (2) (2008), 142–149.

[11] X. Cheng, Y. Kawano, and J. M. A. Scherpen, Graph structure-preserving model reduction of linear network systems, in *Proceedings of the 15th European Control Conference, Aalborg, Denmark, June 2016*, pp. 1970–1975, 2016.

[12] X. Cheng, Y. Kawano, and J. M. A. Scherpen, Reduction of second-order network systems with structure preservation, *IEEE Trans. Autom. Control*, **62** (2017), 5026–5038.

[13] X. Cheng, Y. Kawano, and J. M. A. Scherpen, Model reduction of multi-agent systems using dissimilarity-based clustering, *IEEE Trans. Autom. Control*, **64** (4) (2019), 1663–1670.

[14] X. Cheng and I. Necoara, A suboptimal H_2 clustering-based model reduction approach for linear network systems, in *Proceedings of the 18th European Control Conference*, pp. 1961–1966, 2020.

[15] X. Cheng and J. M. A. Scherpen, Introducing network Gramians to undirected network systems for structure-preserving model reduction, in *Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas, the USA*, pp. 5756–5761, 2016.

[16] X. Cheng and J. M. A. Scherpen, Balanced truncation approach to linear network system model order reduction, *IFAC-PapersOnLine*, **50** (1) (2017), 2451–2456.

[17] X. Cheng and J. M. A. Scherpen, Clustering approach to model order reduction of power networks with distributed controllers, *Adv. Comput. Math.*, **44** (6) (2018), 1917–1939.

[18] X. Cheng and J. M. A. Scherpen, Robust synchronization preserving model reduction of Lur'e networks, in *Proceedings of the 16th European Control Conference, Limassol, Cyprus*, pp. 2254–2259, 2018.

[19] X. Cheng and J. M. A. Scherpen, Clustering-based model reduction of Laplacian dynamics with weakly connected topology, *IEEE Trans. Autom. Control* (2019).

[20] X. Cheng and J. M. A. Scherpen, Novel Gramians for linear semistable systems, *Automatica*, **115** (2020), 108911.

[21] X. Cheng, J. M. A. Scherpen, and B. Besselink, Balanced truncation of networked linear passive systems, *Automatica*, **104** (2019), 17–25.

[22] X. Cheng, J. M. A. Scherpen, and Y. Kawano, Model reduction of second-order network systems using graph clustering, in *Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas, the USA*, pp. 7471–7476, 2016.

[23] X. Cheng, J. M. A. Scherpen, and F. Zhang, Model reduction of synchronized homogeneous Lur'e networks with incrementally sector-bounded nonlinearities, *Eur. J. Control*, **50** (2019), 11–19.

[24] X. Cheng, L. Yu, D. Ren, and J. M. A. Scherpen, Reduced order modeling of diffusively coupled network systems: an optimal edge weighting approach, arXiv preprint arXiv:2003.03559, (2020).

[25] X. Cheng, L. Yu, and J. M. A. Scherpen, Reduced order modeling of linear consensus networks using weight assignments, in *Proceedings of the 17th European Control Conference, Napoli, Italy, June 2019*, pp. 2005–2010 2019.
[26] N. Chopra, Output synchronization on strongly connected graphs, *IEEE Trans. Autom. Control*, **57** (11) (2012), 2896–2901.
[27] J. H. Chow, *Power System Coherency and Model Reduction*, Springer, 2013.
[28] B. Chu, S. Duncan, and A. Papachristodoulou, A structured model reduction method for large scale networks, in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7782–7787, IEEE, 2011.
[29] K. Deng, S. Goyal, P. Barooah, and P. G. Mehta, Structure-preserving model reduction of nonlinear building thermal models, *Automatica*, **50** (4) (2014), 1188–1195.
[30] K. Deng, P. G. Mehta, and S. P. Meyn, Optimal Kullback-Leibler aggregation via spectral theory of Markov chains, *IEEE Trans. Autom. Control*, **56** (12) (2011), 2793–2808.
[31] Q. T. Dinh, S. Gumussoy, W. Michiels, and M. Diehl, Combining convex-concave decompositions and linearization approaches for solving BMIs, with application to static output feedback, *IEEE Trans. Autom. Control*, **57** (6) (2011), 1377–1390.
[32] F. Dörfler and F. Bullo, Kron reduction of graphs with applications to electrical networks, *IEEE Trans. Circuits Syst. I, Regul. Pap.*, **60** (1) (2013), 150–163.
[33] F. Dörfler and F. Bullo, Synchronization in complex networks of phase oscillators: a survey, *Automatica*, **50** (6) (2014), 1539–1564.
[34] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*, Springer, New York, the USA, 2013.
[35] J. A. Fax and R. M. Murray, *Graph Laplacians and Stabilization of Vehicle Formations*, 2001.
[36] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their L_∞-error bounds, *Int. J. Control*, **39** (6) (1984), 1115–1193.
[37] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, vol. 207, Springer Science & Business Media, 2013.
[38] J. G. Willems, Realization of systems with internal passivity and symmetry constraints, *J. Franklin Inst.*, **301** (6) (1976), 605–621.
[39] E. J. Hancock, G.-B. Stan, J. A. Arpino, and A. Papachristodoulou, Simplified mechanistic models of gene regulation for analysis and design, *J. R. Soc. Interface*, **12** (108) (2015), 20150312.
[40] Q. Hui, W. M. Haddad, and S. P. Bhat, Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria, *IEEE Trans. Autom. Control*, **54** (10) (2009), 2465–2470.
[41] T. Ishizaki, K. Kashima, A. Girard, J. -i. Imura, L. Chen, and K. Aihara, Clustered model reduction of positive directed networks, *Automatica*, **59** (2015), 238–247.
[42] T. Ishizaki, K. Kashima, J. I. Imura, and K. Aihara, Model reduction and clusterization of large-scale bidirectional networks, *IEEE Trans. Autom. Control*, **59** (2014), 48–63.
[43] T. Ishizaki, R. Ku, and J.-i. Imura, Clustered model reduction of networked dissipative systems, in *Proceedings of the 2016 American Control Conference*, pp. 3662–3667, IEEE, 2016.
[44] T. Ishizaki, H. Sandberg, K. H. Johansson, K. Kashima, J.-i. Imura, and K. Aihara, Structured model reduction of interconnected linear systems based on singular perturbation, in *2013 American Control Conference*, pp. 5524–5529, IEEE, 2013.
[45] A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: a review, *ACM Comput. Surv.*, **31** (3) (1999), 264–323.
[46] H.-J. Jongsma, P. Mlinarić, S. Grundel, P. Benner, and H. L. Trentelman, Model reduction of linear multi-agent systems by clustering with H_2 and H_∞ error bounds, *Math. Control Signals Syst.*, **30** (2018), 1–38.

[47] H.-J. Jongsma, H. L. Trentelman, and K. M. Camlibel, Model reduction of networked multiagent systems by cycle removal, *IEEE Trans. Autom. Control*, **63** (3) (2018), 657–671.
[48] Y. Kawano, B. Besselink, J. M. Scherpen, and M. Cao, Data-driven model reduction of monotone systems by nonlinear dc gains, *IEEE Trans. Autom. Control*, **65** (5) (2020), 2094–2106.
[49] N. Leiter and D. Zelazo, Graph-based model reduction of the controlled consensus protocol, *IFAC-PapersOnLine*, **50** (1) (2017), 9456–9461.
[50] Z. Li, Z. Duan, G. Chen, and L. Huang, Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint, *IEEE Trans. Circuits Syst. I, Regul. Pap.*, **57** (1) (2010), 213–224.
[51] N. Martin, P. Frasca, and C. Canudas-De-Wit, Large-scale network reduction towards scale-free structure, in *IEEE Transactions on Network Science and Engineering*, 2018.
[52] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.
[53] C. Michele, S. Trip, C. D. Persis, X. Cheng, A. Ferrara, and A. van der Schaft, A robust consensus algorithm for current sharing and voltage regulation in DC microgrids, *IEEE Trans. Control Syst. Technol.*, **27** (4) (2018), 1583–1595.
[54] P. Mlinarić, S. Grundel, and P. Benner, Efficient model order reduction for multi-agent systems using QR decomposition-based clustering, in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 4794–4799, 2015.
[55] P. Mlinarić, T. Ishizaki, A. Chakrabortty, S. Grundel, P. Benner, and J. -i. Imura, Synchronization and aggregation of nonlinear power systems with consideration of bus network structures, in *Proceedings of the 2018 European Control Conference*, pp. 2266–2271, 2018.
[56] N. Monshizadeh, C. De Persis, A. J. van der Schaft, and J. M. A. Scherpen, A novel reduced model for electrical networks with constant power loads, *IEEE Trans. Autom. Control*, **63** (5) (2017), 1288–1299.
[57] N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel, Stability and synchronization preserving model reduction of multi-agent systems, *Syst. Control Lett.*, **62** (1) (2013), 1–10.
[58] N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel, Projection-based model reduction of multi-agent systems using graph partitions, *IEEE Trans. Control Netw. Syst.*, **1** (2014), 145–154.
[59] B. C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, *IEEE Trans. Autom. Control*, **26** (1) (1981), 17–32.
[60] M. Newman, *Networks: An Introduction*, Oxford university press, 2010.
[61] M. E. Newman, The structure and function of complex networks, *SIAM Rev.*, **45** (2) (2003), 167–256.
[62] M. U. B. Niazi, C. Canudas-de Wit, and A. Y. Kibangou, Average observability of large-scale network systems, in *Proceedings of the 18th European Control Conference*, pp. 1506–1511, IEEE, 2019.
[63] M. U. B. Niazi, X. Cheng, C. Canudas de Wit, and J. M. A. Scherpen, Structure-based clustering for model reduction of large-scale networks, in *Proceedings of the 58th IEEE Conference on Decision and Control, Nice, France*, pp. 5038–5043, 2019.
[64] R. Phillips and P. Kokotovic, A singular perturbation approach to modeling and control of Markov chains, *IEEE Trans. Autom. Control*, **26** (5) (1981), 1087–1094.
[65] S. Rao, A. J. van der Schaft, and B. Jayawardhana, A graph-theoretical approach for the analysis and model reduction of complex-balanced chemical reaction networks, *J. Math. Chem.*, **51** (9) (2013), 2401–2422.
[66] W. Ren, R. W. Beard, and E. M. Atkins, A survey of consensus problems in multi-agent coordination, in *Proceedings of the 2005 American Control Conference*, pp. 1859–1864, IEEE, 2005.

[67] D. Romeres, F. Dörfler, and F. Bullo, Novel results on slow coherency in consensus and power networks, in *Proceedings of the 2013 European Control Conference*, pp. 742–747, IEEE, 2013.
[68] T. Sadamoto, T. Ishizaki, and J. -i. Imura, Average state observers for large-scale network systems, *IEEE Trans. Control Netw. Syst.*, **4** (4) (2017), 761–769.
[69] H. Sandberg and R. M. Murray, Model reduction of interconnected linear systems, *Optim. Control Appl. Methods*, **30** (3) (2009), 225–245.
[70] L. Scardovi and R. Sepulchre, Synchronization in networks of identical linear systems, in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 546–551, IEEE, 2008.
[71] S. E. Schaeffer, Graph clustering, *Comput. Sci. Rev.*, **1** (1) (2007), 27–64.
[72] S. Trip, M. Cucuzzella, X. Cheng, and J. Scherpen, Distributed averaging control for voltage regulation and current sharing in DC microgrids, *IEEE Control Syst. Lett.*, **3** (1) (2018), 174–179.
[73] V. Van Breusegem and G. Bastin, Reduced order dynamical modelling of reaction systems: a singular perturbation approach, in *Proceedings of the 30th IEEE Conference on Decision and Control, Volume 2*, pp. 1049–1054, 1991.
[74] A. J. van der Schaft, On model reduction of physical network systems, in *Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems, Groningen, The Netherlands*, pp. 1419–1425, 2014.
[75] A. J. van der Schaft, S. Rao, and B. Jayawardhana, Complex and detailed balancing of chemical reaction networks revisited, *J. Math. Chem.*, **53** (6) (2015), 1445–1458.
[76] N. Xue and A. Chakrabortty, LQG control of large networks: a clustering-based approach, in *Proceedings of the 2017 American Control Conference*, pp. 2333–2338, IEEE, 2017.
[77] L. Yu, X. Cheng, and J. M. A. Scherpen, h_2 sub-optimal model reduction for second-order network systems, in *Proceedings of the 58th IEEE Conference on Decision and Control, Nice, France*, pp. 5062–5067, 2019.
[78] L. Yu, X. Cheng, and J. M. A. Scherpen, Synchronization preserving model reduction of multi-agent network systems by eigenvalue assignments, in *Proceedings of the 58th IEEE Conference on Decision and Control, Nice, France*, pp. 7794–7799, 2019.
[79] D. Zelazo, S. Schuler, and F. Allgöwer, Performance and design of cycles in consensus networks, *Syst. Control Lett.*, **62** (1) (2013), 85–96.

Dirk Hartmann, Matthias Herz, Meinhard Paffrath, Joost Rommes,
Tommaso Tamarozzi, Herman Van der Auweraer, and Utz Wever

12 Model order reduction and digital twins

Abstract: We are currently facing a substantial transformation of our industrial world and the way our economics are organized. This transformation, known as digitalization, is driven by the systemic integration of information technology in all kinds of devices, machines, and factories such that new smart networks are formed and new smart products have the ability to monitor, to forecast, and to control their behavior. One of the fundamental pillars of digitalization is simulation technology, since it enables the new intelligence layer in the form of digital twins which mirror the physical systems into the digital world – also named by Gartner Inc. as a top technology trend for 2017 and 2018. Creating such intelligence layers over several domains and life cycle phases requires, among other challenges, technologies for transforming and reducing complex simulation models. Exactly for this task a key technology is model order reduction (MOR). However, MOR is not only a key technology within emerging digital twins but also helps to reduce simulation times in the existing everyday business of simulation engineers. This is especially important when for a simulation model a large number of evaluations are needed. Within this chapter we present use cases where MOR is a key enabler for the realization of digital services and the reduction of simulation times. Furthermore we outline the potential of MOR in the context of realizing the digital twin vision.

Keywords: digital twin, virtual sensors, control, predictive maintenance, circuit simulation

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

12.1 Introduction

This chapter provides an overview of several projects which we worked on throughout the last years. These projects were initiated from different directions and perspectives since the authors of this chapter work in multiple departments across Siemens. Nevertheless, all of our projects were either part of concrete business opportunities

Acknowledgement: The digital twin was implemented by Christoph Ludwig; see again [13], [64], and [63].

Dirk Hartmann, Meinhard Paffrath, Utz Wever, Siemens AG, Munich, Germany
Matthias Herz, Siemens AG, Erlangen, Germany
Joost Rommes, Mentor, a Siemens Business, Wilsonville, USA
Tommaso Tamarozzi, Herman Van der Auweraer, Siemens Industry Software NV, Leuven, Belgium

https://doi.org/10.1515/9783110499001-012

or of predevelopment activities to evaluate new business opportunities. This means that the goal was always to improve products, to develop new products, or to evaluate the potential lying in innovative business ideas. In this environment the application of model order reduction (MOR) was not a goal in its own. Instead the application of MOR was always triggered by the requirements coming from the project goals. In particular for predevelopment projects such a goal is typically to evaluate the commercial benefit lying in new technologies, which in our case was MOR.

In this chapter we start with outlining the underlying business visions of digitalization and digital twins and the role of MOR within this vision. This part is followed by a report of our experience with productizing MOR algorithms. Finally, we report the content, the challenges, and the results of some of our projects.

Throughout this chapter we try to give an insight into our work between the poles of business models and technological challenges, which is sometimes even the greatest challenge.

12.2 Digitalization and digital twins

Complexity in today's industry is exploding. New production methods, miniaturization of electronics, novel sensor technologies, and last but not least the Internet of things have led to many disruptive developments implying more and more complex products. On the one hand, this offers unique opportunities, e. g., in terms of efficiency or autonomy of components, products, and complex systems. On the other hand, it challenges today's design, engineering, operation, and service paradigms mostly focusing on manual expert interaction, which can hardly, if at all, handle this enormous complexity.

Digitalization changes everything everywhere. With the rise of new technology trends, such as AI foundations, intelligent things, cloud to edge, or immersive experiences [76], many of today's paradigms can be expected to be disrupted. Not only in the consumer market, as we can clearly observe today, but also in the industrial and medical sectors we see disruptions as proven by first early adopters.

Digital twins will be one key answer to these challenges; see, e. g., [24, 35, 81] for a broad overview from an engineering perspective. They are the next wave in simulation technologies (Figure 12.1). Digital twins integrate all (electronic) information and knowledge generated during the lifetime of a product, from the product definition and ideation to the end of its life. Examples of these data range from the initial requirements which have led to the design of the product, the design and engineering data, which have been generated during virtual design, to operation data such as sensor values collected during operation. The data themselves are only a central asset, if it can be used to make relevant predictions providing the right level of information at

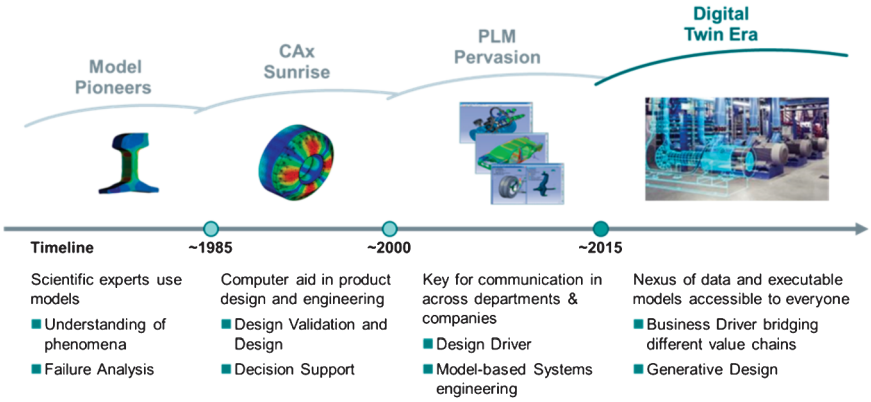


Figure 12.1: Simulation is evolving from a troubleshooting tool to a key business driver in the form of digital twins.

the right time. Ultimately, digital twins mirror products and systems from the real into the digital world and vice versa.

From a high-level point of view, information included in digital twins can be split in to two categories: (i) pure data values with only little additional structure and knowledge associated, such as data gathered from sensors, and (ii) structured executable model-based data, in particular simulation models. Thus from this point of view digital twins bring together classical data-based schemes with model-based approaches such as simulation and optimization (Figure 12.2).

Today, most model-based approaches, and in particular simulation, are domain-specific and mostly used during design and engineering. The core concept of the dig-

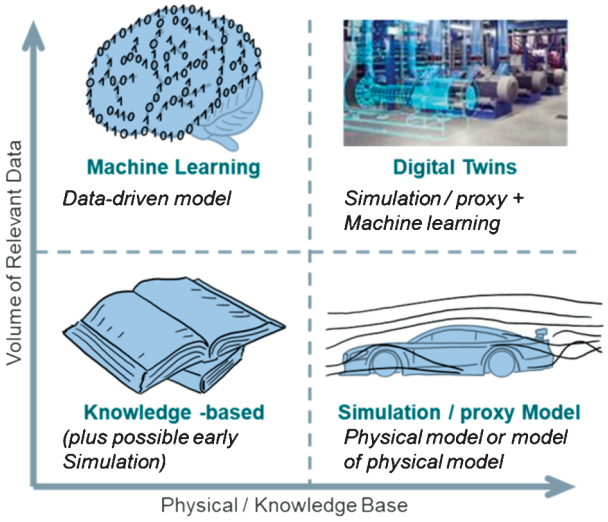


Figure 12.2: Digital twins integrate model- and simulation-based approaches with data-based approaches such as artificial intelligence.

ital twin is to extend their usage along the complete life cycle and to deliver new services providing the right information at the right place in an efficient way, for example, digital twins supporting early system configuration during the sales process or optimization of operation and service concepts. This broad usage implies a number of requirements to modeling and simulation which diverge from its classical use in design and engineering:

- **Interactivity** – Speed and accuracy define the value of simulation and digital twins. Being very accurate, today's model and simulation approaches are extremely time-consuming. Speeding them up, while retaining the right level of accuracy, is crucial for extending the use of digital twins.
- **Reliability** – Users of digital twins cannot be expected to be sophisticated experts, like it can be expected during the use in design and engineering. Thus any prediction by the digital twin must be fail-safe and/or provided along with confidence intervals such that no expertise is required to interpret the results or can be used autonomously, e. g., by controls.
- **Usability** – Model-based and simulation tools are expert-centric today. Their resources are limited and thus the use of corresponding tools today is limited by the availability. Therefore, any digital twin solution must be accessible also for nonexperts from a usability perspective.
- **Security** – Many business models based on the digital twin will require to exchange digital twins between different parties. Reverse engineering must be prevented, such that no intellectual property is lost.
- **Deployability** – Digital twins will be used differently from the place where they have been created, e. g., on customer premises, in the cloud, on controls. Thus deployment must be easy to reduce barriers and efforts.

The digital twin concept has been originally introduced in 2003 by Michael Grieves [41] and first put to public by NASA in 2012 [38]. Digital twins are considered so important to business, that they were named one of Gartner's Top 10 Strategic Technology Trends for 2017 [76]. They are becoming a business imperative, covering the entire life cycle of an asset or process and forming the foundation for connected products and services. Companies that fail to respond will be left behind. For example, it is predicted that companies who invest in digital twin technology will see a 30 % improvement in cycle times of critical processes [77]. A potential market of 90 billion US dollar per year associated to corresponding offerings is predicted [28].

To realize the vision of digital twins, MOR is a key technology. Other key technologies cover novel user interaction paradigms and devices (such as virtual, augmented, or mixed reality), technologies for merging data and model-based approaches, or semantic technologies to easier built-up systems of digital twins.

12.3 Model order reduction in the context of digitalization and digital twins

The digital twin vision extends the expert-centric focus of modeling, simulation, and optimization technologies towards a digital assistance for everyone in day-to-day decisions. This is supported by a double exponential growth of capability in simulation technology. On the one hand, computational hardware is developing exponentially according to Moore's law [94]. On the other hand, efficiency of simulation algorithms is subject to exponential growth as well [91]. With this growing capability, computer-aided paradigms have become so powerful that they can provide novel simulation-based assistance in many fields, for example, digital twins providing new services for predicting failures, increasing operational efficiency, or for service planning [76].

However, compared to computer-aided tools in engineering, computer-aided assistance by means of digital twins is a niche application. The manual setup of corresponding models is a tedious task requiring simulation experts. This limits the use of model-corresponding concepts since corresponding efforts and costs are major roadblockers for increased use [60]. Furthermore, the lack of rigorous concepts for quantifying errors often implies very conservative safety margins, so that the full potential often cannot be exploited. Missing protection of intellectual property of models and the lack of standards (the functional mock-up interface [FMI] is only adopted slowly [10]) are hindering further. Thus today, digital twin-based approaches are only adopted in applications of high value, e. g., heavy-duty vehicles [44]. MOR [5, 4, 95] is a key technology to solve these challenges in the context of digital twins. By splitting computations in an offline and an online phase, computational effort is shifted to an offline phase allowing interactive simulation during the online phase. However, not only does this imply a speedup of calculations, but due to their reduced information set, reduced-order models (ROMs) protect intellectual property efficiently. While geometries can be recovered from the meshes of three-dimensional simulations, this is not the case for ROMs, in particular since the output generally focuses on the quantity of interest, i. e., a temperature at a single location rather than a complete temperature field. This furthermore increases the usability, since only relevant information is accessible. In addition, ROMs can be efficiently containerized using available standards such as FMI [10], thus increasing usability. In particular in view of the challenges laid down in Section 12.2, MOR is a key technology for digital twins.

A variety of concepts and approaches have been introduced in the last decades mostly using projection-based approaches such as proper orthogonal decomposition (e. g., [111]), balanced truncation (e. g., [42]), the reduced basis method (e. g., [80]), or Krylov subspace methods (e. g., [7]). The key idea of most approaches is to reduce the space of considered functions by means of an appropriate low-dimensional basis. For (close-to-)linear models, MOR is state-of-the-art in computational engineering and science. For nonlinear models it is a highly active field of research (e. g., [8]).

In addition to classical MOR methods, machine learning offers an alternative approach. Many successful applications, such as the efficient operation of wind parks [62], have been realized during the last years. Compared to model-based approaches, machine learning concepts require comparably little manual efforts to be set up. However, being data-centric, machine learning is not applicable where only few data are available. This is often the case in industrial applications, where relevant data cannot be measured, cannot be shared (e. g., due to IP concerns), or is simply not available (e. g., failure data for small lot products). On the one hand machine learning could be used to speed up simulation models by means of learning the underlying simulation data (e. g., [43]), but a combined approach with the ROM as the foundation and machine learning closing the accuracy gap seems to be a more promising approach [58]. However, such combined approaches have rarely been considered in the past and we believe that it has a strong future potential.

Within the following sections we review the application of MOR in projects which tackled concrete aspects of the digital twin vision described above. However, before describing these projects we generally review the process of productizing algorithms since the overall goal of every industrial R&D activity is to improve or deliver new products or services.

12.4 Model order reduction – from algorithms to products

12.4.1 Introduction

The process of making an algorithm suitable for use in commercial (CAE) software, also referred to as productizing, can be long and difficult to plan. Even if algorithms are known in the literature to be generally robust, the applicability to commercial software implementation is not always straightforward. In particular, it is challenging to foresee the user's needs and desired application of a method so that the method's assumptions do not lose validity. Moreover methods and software developers often face strict boundary conditions regarding implementation variants that are dictated by, e. g., the structure of the underlying physics engine or solver in which novel methods are implemented. Furthermore, while algorithms are usually designed by experts, the actual end-users are typically *not* experts in using those algorithms – they are experts in their own domain. Hence successful *productizing* requires not only that algorithms are robust with respect to applications, but also that their parameters can be (re)set in an *automatic* and *dynamic* way: *automatic* to reduce the need for users to set parameters and *dynamic* because parameters may need to be adjusted not only at the start of but also during the simulation. In this way the numerical methods become *transparent* to the user while the freedom of the user to interact with the algorithm is

somehow restricted. A good balance between *transparency* and *user freedom* has to be found. The situation becomes even more complicated if a working algorithm is not available or if the problem at hand is not yet fully understood and analyzed.

In this section we describe the various phases from algorithms to products. We assume that the problem to be solved is sufficiently well-defined (and constrained) and end-user requirements are known, and hence we focus on the process of solving the problem. As a concrete example, one could consider the typical MOR problem: given a dynamical system, find a reduced dynamical system that approximates the original system with a controllable trade-off between error and speed, and preservation of key properties like stability. We identify the following phases that will be discussed in more detail in the next subsections:

- research: literature study and investigation of novel approaches;
- prototyping: implementation of stand-alone or integrated software to allow feasibility studies;
- productizing: implementation in, or as, a product;
- customer feedback: closing the loop with new results and new requirements from end-users.

These phases may overlap in practice and moreover the process might become iterative: After customer feedback, but also during productizing, often new insights are obtained which require further research and prototyping.

12.4.2 Research

During the research phase, traditionally two activities are dominant: literature study and design of novel approaches. Depending on the complexity and confidentiality of the problem, these activities are carried out by one or more researchers, e. g., a technical leader, a (team of) researcher(s), and a MSc/PhD student, or even outsourced to an external party. For literature study, it is not only important to have the problem at hand well-defined, one must also know *which* literature to study. In some cases the right sources are naturally available because the researcher has experience on the topic. In other cases the topic may be less or even not covered in existing literature, or not in the context of the application at hand. Communication with colleagues (potentially in different divisions) and external parties like universities is then required to at least find a starting point. In several cases such contacts, for instance made during conferences or European networks like EU-MORNET [30], may develop to long-lasting collaborations with rewards such as scientific and commercial breakthroughs and staffing opportunities. The circuit simulation-related MOR work described in Section 12.10, for example, has been performed in collaboration with the TU Eindhoven, in the European project ASIVA14 [21], while the drivetrain dynamics simulation tools described

in Section 12.8 have been developed in cooperation with the KU Leuven and the University of Calabria within several years of research interactions and projects such as the Marie Curie H2020 project DEMETRA [57].

Often the problem is not sufficiently covered in the literature: The context or application may be different, the boundary conditions imposed by the main CAE solvers could be a limiting factor to the implementation of original algorithms, or the problem itself may simply be new. Even if the problem is well covered, one usually has to adapt and tune the proposed methods to the problem at hand. This stage, which may vary from simple changes of existing strategies to the design of novel approaches, typically involves prototyping, which we discuss in more detail in the next subsection.

12.4.3 Prototyping

When a set of methods is defined to achieve a specific target it is time to develop the first prototype code in order to test if the assumptions made during the research stage are valid and if the knowledge gained has application potential. In the prototyping phase, usually, one or more method developers and/or software engineers start to define preliminary software architectures and begin the implementation of a prototype code. Common choices for development environments are MATLAB [66] and Python [79]. As a good practice, the developed mock-up code should be easy to extend, it should be tested in a similar environment as compared to the target solver in which the final implementation is foreseen, and it should be flexible enough to be tested in multiple scenarios and maintain a satisfactory level of user-friendliness. In this way new extensions of the methods can be easily tested on multiple scenarios, the code can be shared with colleague researchers and consultants for usage in bilateral projects, and the risk of failure during the prototyping-to-product transition is reduced. Once the set of algorithms is mature enough, it is important to perform stress tests in the largest possible range of applications. Automatic testing is not mandatory but is surely an added value.

Using the specific case of MOR-related algorithms, it can happen that a large amount of parameters must be set by the user and that these are of difficult physical and mathematical interpretation to nonexpert users. Moreover, automatic parameter tuning algorithms are rarely available in the literature for the specific application foreseen for the implemented method. For this reason a big effort during the prototyping phase is generally spent in making the numerical methods *robust* and the automatic parameter setting *transparent* while still allowing advanced users to retain the desired level of control on the numerical method. During the prototyping phase of the method described in Section 12.8 the original number of parameters linked to the underlying MOR strategy was drastically reduced thanks to automatic parameter setting and the remaining parameters have been readapted to represent physical quantities that are easy to understand from a user point of view. Similarly, for the MOR approach

described in Section 12.10, most of the low-level parameters have been combined into macro-options that give the user (and developer) easy control over performance and accuracy.

If this target is achieved, the prototype should be tested on real engineering cases during, e. g., bilateral services projects and/or funded research projects. This step is useful to confirm the potential of the method, find out unforeseen usages, and detect potential limitations.

Often, at the end of the prototyping stage, a preliminary user interface is created to explore the usability of complex numerical solutions.

12.4.4 Productizing

Once the set of algorithms has reached a satisfactory level of robustness and usability the prototyping phase can be sided by the productizing phase. First the developed methods should be assessed for their market value, general applicability, and strategic importance. This stage is fundamental in order to assign a well-balanced amount of development resources. After this assessment the correct number of resources – generally one or more developers and/or software engineers – is assigned the task of implementation into the target commercial CAE solver. The goal is to translate customer specifications, design requirements, and prototype code into a professional and consistent implementation. Especially during the implementation of novel methods, it is of paramount importance that researchers and developers communicate on a regular basis. In practice, the specific research knowledge and the application-oriented character of many methods makes it hard to make consistent and complete code design specifications. In this case, developers may face the challenge of interpreting prototype code and might implement nonintended behavior. It is advisable to initially allow researchers and developers to spend time together and even promote pair-coding activities. The more the algorithms are complex and have a dual theoretical-applied character, the more this practice should be promoted. During this period and in parallel with the method implementation into commercial solvers, a team of developers might also start to implement a user-friendly user interface. The more the numerical method has been refined and made robust, the less the user interface creation process is challenging. During the creation of the MOR method applied to drivetrains described in Section 12.8 a prototype user interface was also created in parallel with the research and method prototyping. This and the strong cooperation between the research and development units of Siemens allowed for a smooth transition of the prototype code and prototype user interface into a commercially available solution for MOR applied to drivetrain problems. One of the main challenges in the productizing of the methods in Section 12.10 was the choice on which parameters to make available to the user. This has been an iterative process itself, where researchers, developers, and application engineers were involved.

12.4.5 Customer feedback

No matter how sound the underlying theory is and no matter how many tests have been done, the most useful feedback on the quality (performance, accuracy) of the product is end-user feedback. The difficulty, as mentioned before, is that the test cases used by development teams typically do not cover completely the real cases used by customers. Hence, there is always a risk involved with releasing improved or new functionalities. The key is again communication to manage expectations, not only internally with sales and product engineering teams, but also with the customer (either directly or via customer-based application engineers): Roughly speaking, one of the first things to do when a customer request (bug report or enhancement request) is filed is to analyze whether there is a real bug in the theory and/or implementation, or whether the result is within accuracy tolerances but outside customer expectations. Ideally this first analysis is done by application or test engineers, but depending on the complexity, development teams may need to be involved as well. When the issue is identified as bug, apart from implementation errors, regularly one will have to go back to the underlying theory, for instance to adjust initially made assumptions or estimates, hence reiterating the phases described in the previous subsections.

When the result is within accuracy tolerances but outside customer expectations, the situation can become more complicated. Not only one has to be sure that the result is indeed within tolerances, but one also has to explain this to the customer: Particular care has to be taken here to avoid breaking long-standing trust relations. Furthermore, it might also be an indication that certain settings and options in the software are not clear for users, which may require software and/or documentation to be improved.

During the circuit simulation-related MOR work described in Section 12.10 all of the above-mentioned scenarios have happened. For example, a bug reporting a too large difference in signal delay was initially identified as a side effect of the way the delay was computed during postprocessing of simulation data. A deeper analysis, however, showed that while the actual delays were still within (user-settable) simulation tolerances, the used error estimations in the code were in fact too optimistic, and hence all phases above had to be reiterated in order to fix the issue. After the release of the first version of the drivetrain simulation tools described in Section 12.8, a user signaled an extension request to improve the usability of the tool for large system-level models that include multiple drivetrains. The user was contacted and asked for feedback about the urgency of the required extension. It was then decided in agreement within the party to take the time to develop a proper interface for the requested extension and release it together with the official product release a few months after.

12.4.6 Concluding remarks

We conclude by repeating what was mentioned in the introduction: The phases described in this section are typically visited in an iterative way. Moreover, they may in fact be visited in any order, for instance when through the acquisition of software (or company) one starts with an actual product that has to be integrated in a larger environment.

12.5 Use case – virtual sensors

12.5.1 Vision

The use of models to enhance or extend test-based engineering processes is one of the key application fields of model-based system testing [27]. Test data exploitation can be greatly enhanced by complementing sparse physical sensor measurements with model-based virtual sensor data [107, 20]. Control system efficiency can be increased by providing optimal control inputs using quantities which cannot be measured directly and operating system performance can be tracked through monitoring internal system states. Traditionally such control inputs or internal states of devices are measured during operation by hardware sensors [53]. However, due to cost restrictions or extreme physical conditions it is not possible to place hardware sensors at any desired position in any device. The goal of virtual sensors in all these applications is to provide online information about internal conditions or system performance based on simulation models instead of hardware sensors. These system models can be used offline to expand data sets or may be running parallel to operation, permanently synchronized with the current operation state, and report the desired internal states at the usual rate of the hardware sensors. From a business perspective such virtual sensor software modules may not only add value to the engineering process but can enable new simulation-based products such as advanced condition monitoring for improved availability or reduced downtimes. Furthermore, when virtual sensor algorithms and existing controllers are integrated into one software architecture, novel model-based controllers can be realized.

However, the systematic application of embedded simulation models for extended data analysis or parallel to operation is still a young field of activity. On the other hand, driven by the need to reduce development cycle times, simulation has become a frequently used tool during the development of products [17]. To draw reliable conclusions during the development process detailed three-dimensional simulation models are needed and the evaluation of these simulation models typically involves significant simulation times. This makes their reusage inside virtual sensor software and related state estimation a challenge.

In fact one of the central requirements for simulation models inside virtual sensors is the capability for fast estimation or even real-time capability when the results should be updated within the usual update frequency of hardware sensors. For this reason, MOR [8, 5] is applied to, e. g., detailed three-dimensional simulation models developed for design engineering purposes. This ensures reusage of the already available information and it allows to obtain fast or even real-time capable surrogate models which nevertheless operate within an acceptable accuracy.

12.5.2 Technological challenges

In this section the required steps for a virtual temperature sensor are described.

For a virtual temperature sensor the starting point is the thermal energy equation which reads for heat conduction with Fourier's law $\boldsymbol{q} = -\kappa \nabla T$ [67, 55, 105] for a computational domain Ω as

$$\begin{aligned} \partial_t(C_p T) + \nabla \cdot (-\kappa \nabla T) &= h && \text{in } \Omega, \\ \boldsymbol{q} \cdot \boldsymbol{n} &= h_f && \text{on } \Gamma_N, \\ \boldsymbol{q} \cdot \boldsymbol{n} &= \alpha(T - T_{\text{amb}}) && \text{on } \Gamma_R. \end{aligned} \tag{12.1}$$

Here, T is the temperature field, T_{amb} is the ambient temperature, C_p is the specific heat capacity, κ is the heat conductivity, and α is the convection coefficient [61, 56]. In a typical industrial setup, Dirichlet boundary conditions are not used. Instead, the thermal losses are captured by the volume heat load h or the heat fluxes h_f at the boundary. The most important boundary condition is the Robin boundary condition, which is also known as Newton's law of cooling [56]. This boundary condition models the thermal communication with the environment. Especially when a thermal model contains only solid bodies which are surrounded by a coolant, the convective heat transfer coming from the coolant flow can be modeled by a given distribution of convection coefficients. For example, this applies to thermal models of electric motors which contain the solid parts of the stator, rotor, and housing, but not the flow domain of the cooling air flow.[1]

To start the MOR procedure, the thermal energy equation has to be written as a state-space system in the form

$$\begin{aligned} E \frac{\mathrm{d}}{\mathrm{d}t} x &= A\, x + B\, u, \\ y &= C\, x. \end{aligned} \tag{12.2}$$

1 A full conjugate heat transfer model would lead to a dramatic increase in complexity and computational time, since the turbulent and thermal air flow in the rotor-stator gap and around the stator cooling fins needs to be resolved [56].

Here $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the input which drives the system, and $y \in \mathbb{R}^p$ is the measurable respectively observable system output. Furthermore, the system matrices are of dimensions $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$.

To obtain the thermal energy equation (12.1) as system (12.2), the following steps need to be performed.

- The heat load h and heat flux h_f are assumed to consist additively of contributions which only vary in time, i. e., $h = h_1(t) + \cdots + h_l(t)$ and $h_f = h_{f,1}(t) + \cdots + h_{f,k}(t)$. This assumption is fulfilled for a typical thermal simulation model in the industry since the usual procedure in commercial three-dimensional simulation software is (a) to mark the relevant model components on which the heat loads and the heat fluxes are applied and (b) to specify the total thermal losses which are produced by these model components. In a subsequent step the commercial software distributes the total thermal losses spatially homogeneous over the marked model component [2]. This leads to $m = l + k$ inputs.
- A finite element method or finite volume discretization approach in space brings the thermal energy equation almost into the desired state-space formulation. Some minor changes are necessary since during the finite element method or finite volume assembly procedure a constant vector b_0 occurs at the right-hand side due to the Robin boundary condition [59]. This part is added to the input terms by extending the input matrix B as $B = (B, b_0)$ and the input vector u as $u = (u, 1)$. This leads in total to $m = l + k + 1$ inputs.
- Additionally the output matrix C has to build up according to the desired location of the virtual sensors. This is done by marking for each virtual sensor its relevant nodes or elements in the computational mesh. This determines for each virtual sensor its corresponding row in the output matrix C.

The major technological challenge in this process is to access the assembled system matrices from commercial CAE software. For Simcenter Thermal Flow [2, 99] this was solved with a special subroutine and for NX Nastran [2, 71] this was solved with DMAP [2, 25]. However, there are commercial CAE software packages which do not provide any customization possibility to access the system matrices or some explicit solvers even do not assemble global system matrices.

Once the state-space system corresponding to the thermal simulation model is obtained, any MOR method can be applied which works on state-space systems of type (12.2) [8, 5]. For thermal simulation models the matrices are huge in size but sparse [59]. In our experience, typical industrial small-sized thermal models contain up to 10^6 degrees of freedom and typical medium-sized thermal models contain up to 10^8 degrees of freedom. For this reason the Krylov subspace MOR methods are a good choice since Krylov subspaces have a long history in connection with linear iterative solvers for especially huge and sparse linear equation systems [39]. A detailed review of Krylov subspace MOR methods can be found, e. g., in [5, 8, 7, 72]. Instead of giving yet another introduction into Krylov subspace MOR methods we concentrate

on what is necessary to realize virtual sensors with these methods in an industrial environment.

However, classical Krylov subspaces MOR methods such as [72] are feasible only for linear and time-invariant systems (12.2). For products with temperature-dependent material properties, the heat equation (12.1) becomes nonlinear due to $\kappa = \kappa(T)$. In this case nonlinear algorithms (e. g., [8, 114, 69]) need to be applied.

12.5.3 Project description

The first goal was to establish a user-friendly work flow for generating ROMs from existing three-dimensional thermal simulation models in an industrial environment. For this goal the determining factors are that (a) the simulation models are constructed in a commercial CAE software and (b) the simulation engineers have profound knowledge in their physical domain and the used CAE software but in general they are not experts in MOR nor they are programmers; see Section 12.4 for more details. More precisely, since all commercial CAE software packages are used through graphical user interfaces, simulation engineers are generally not used to run algorithms in command line tools or software development environments.[2] This starting position requires (a) to interact with the commercial CAE software and (b) to hide the details of the MOR algorithms from the user (Section 12.2). For this reason a MOR plug-in for Simcenter [26], the flagship product of Siemens in the CAE market, was developed. This MOR plug-in adds a ribbon to the Simcenter Graphical User Interface (GUI) which guides the user with buttons and following pop-up windows through the process of generating, applying, and exporting ROMs (Figure 12.3). This MOR plug-in was developed as Siemens internal engineering tool and is in productive usage within different projects and departments. To ensure that the resulting GUI matches the user expectations, several

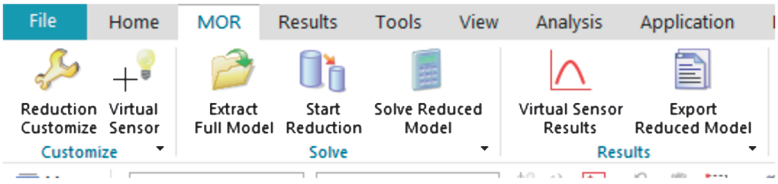


Figure 12.3: Model order reduction plug-in.

2 The main task of simulation engineers is to support or enable the product development process based on simulative information. To accomplish this, simulation models with the relevant physical information are built from CAD models. From the obtained simulation results conclusions are then drawn, e. g., about the product design or the reliability, and this information is fed back in the development process. This means that simulation engineers are focusing on product development and not on algorithm development.

in-house simulation engineers were included in the process of designing the GUI and the work flow of the plug-in (Section 12.4.5). With this plug-in the step to easily generate ROMs from existing three-dimensional thermal simulation models was solved.

For realizing virtual temperature sensors, the next step is to wrap the obtained ROM inside a virtual sensor software module which is runnable on the target hardware and software architecture. For the communication with the surrounding software architecture, the virtual sensor software module must receive the current operating conditions, transform these conditions into the required input for the ROM, call the ROM, transform the ROM results into the required format, and feed the properly formatted ROM results back into the surrounding software architecture. Furthermore, one communication cycle of that kind must be done within an expected frequency.

A crucial point is the available information during operation. Typically the available information is not identical with the required input for the ROM. For example, for electric motors the current is known during operation and can be fed into the virtual sensor software module. However, the ROM obtained from the thermal simulation model requires heat loads as inputs. Thus it must be part of the offline phase, i. e., the creation phase of an ROM, to provide the required information for mapping the available inputs (e. g., current) to the required ones (e. g., heat loads). This task involves detailed product-specific knowledge and is a central key for a vital and accurate virtual sensor software module. In our projects this task was solved with detailed look-up tables which were provided by the respective engineering departments.

Another important ingredient of a virtual sensor software module is to ensure that the ROM is permanently synchronized with the actual operation condition of the product. This requires that the virtual sensor software module receives and adequately processes the relevant information about the current operation state to keep its internal ROM synchronized. In our projects we solved this task with online filtering algorithms [101, 100, 46, 52], such as Kalman filters, where the filtering was done based on the available temperature hardware sensors and the corresponding temperatures coming from the ROM for these locations.

The last step in the development procedure is to run system tests to improve the solutions based on this feedback. Some of our projects have currently reached this stage, whereas in our in-house hardware lab virtual temperature sensor software modules are already running and tested.

12.5.4 Results and summary

The main challenge of our projects was that virtual temperature sensors based on ROMs were realized for the first time for the considered products. This means that not only the software modules had to be developed but we also had to establish a work flow of how to realize these virtual sensor software modules. While customized one-time solutions are sufficient for research projects and first prototypes, they are not a

proper solution for new services or products. New products or services require a sustainable work flow which is integrated into the existing development ecosystem of the involved engineers (Section 12.4). The approach we put into practice started from existing three-dimensional thermal simulation models. These models were compressed with MOR and the resulting ROMs were small and fast enough to be executed within the usual hardware sensor update frequency, either in an embedded environment or in a cloud environment which is connected to the product. In order to integrate this task in the existing development ecosystem of simulation engineers, we developed a plug-in for Simcenter, which is the standard CAE software within Siemens for simulation-based engineering steps.

The following task of integrating the ROM into the target hardware and software system was still realized as customized and manual solution for each product. A potential future integration of this step into the existing development ecosystem of automation engineers are new state-observer blocks within the Totally Integrated Automation portal, which is the engineering platform from Siemens for all kinds of automation tasks [104]. During our projects prototypical blocks for such state-observers based on ROMs were developed but a fully integrated solution is still pending. Nevertheless, exactly the integration into existing automation engineering software tools is the second important step in realizing virtual sensors in a standard way. Overall, in a typical industrial development ecosystem, the simulation engineers create the ROMs for virtual sensors and the automation engineers integrate the virtual sensors into the software architecture of the products. Thus, to establish virtual sensors based on ROMs there must be a fully integrated solution for both, the simulation and automation engineering ecosystems (Section 12.2).

12.6 Use case – predictive maintenance

Data-driven operation support has been a topic for about 10 years. The efficiency of methods such as condition-based monitoring or sensor-based fault detection depends on the amount and the placement of sensors.

12.6.1 Motivation for model-based predictive maintenance

Very new is the demand of simulative operation support [29]. It allows monitoring every position and physical size of a system at any time point. Due to this knowledge, the system state may be predicted at any time. A simulation-based software program runs in parallel to the operation and is synchronized by sensor values at every time point. In many reports this is called the digital twin. The benefits are summarized in Figure 12.4. Among the most important benefits are inspection and service planning,

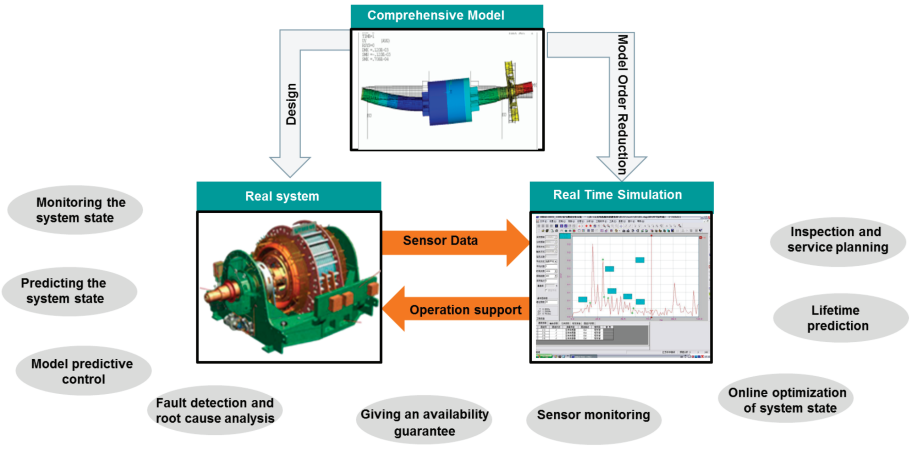


Figure 12.4: Benefits of simulation-based operation support.

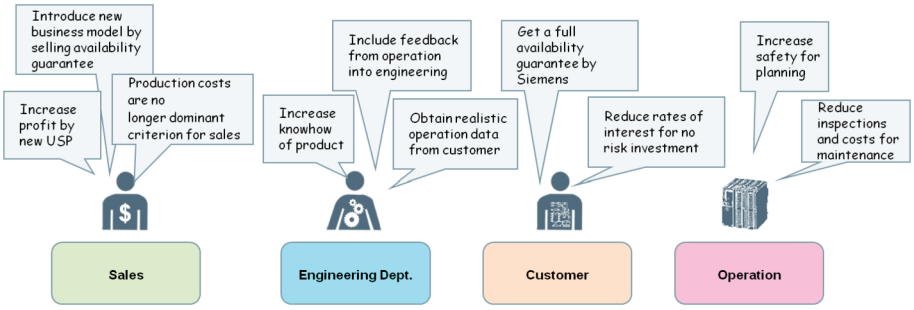


Figure 12.5: Advantages for offering a digital twin additional to the hardware.

lifetime prediction, advanced fault detection, and control and optimization during operation. Selling not only the hardware of the system but also additional services can be a huge advantage in countries with high salaries. There are existing first clients of Siemens who demand this kind of operation support. In a first view some services such as giving an availability guarantee may sound risky for a company. On the other hand this is a unique selling point and selling the risk may bring good profit; see any insurance company. A more accurate analysis leads to the conclusion that all participants may benefit from an availability guarantee (Figure 12.5). Giving an availability guarantee for products cannot mean that there are no downtimes due to faults or inspections. Instead, the downtimes, especially the unexpected downtimes, should be reduced. The main task is to detect faults at a very early stage and predict their degradation. Thus, immediate downtimes are transferred to predictive downtimes. The base for giving an availability guarantee is the early detection of faults. If a fault is detected, then its degree of degradation is predicted. Depending on this prediction an inspec-

tion may be scheduled and/or the performance of the system is reduced in order to achieve the inspection time. Often, the plant is located in very isolated regions. Thus, the execution of maintenance and spare part supply must be planned very carefully. The early knowledge of the cause of failure is of tremendous interest.

12.6.2 Oscillatory mechanical systems

We consider a solid body $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, composed of a material with Young's modulus $E \geq 0$ and Poisson ratio $-1 \leq \nu \leq 0.5$. The body is subject to volume forces $\boldsymbol{f} : \Omega \to \mathbb{R}^3$ and surface forces $\boldsymbol{g} : \partial\Omega \to \mathbb{R}^3$. Displacements $\boldsymbol{d} : \Omega \to \mathbb{R}^3$ from some appropriate function space $\mathcal{E}(\Omega, \mathbb{R}^3)$ are determined by the equations of linear elasticity (see, e. g., [49]):

$$\begin{aligned} -\operatorname{div}(\boldsymbol{A}\boldsymbol{e}(\boldsymbol{d})) &= \boldsymbol{f} && \text{in } \Omega, \\ (\boldsymbol{A}\boldsymbol{e}(\boldsymbol{d})) \cdot \boldsymbol{n} &= \boldsymbol{g} && \text{on } \Gamma_N, \\ \boldsymbol{d} &= 0 && \text{on } \Gamma_D, \end{aligned} \tag{12.3}$$

where the strain $\boldsymbol{e}(\boldsymbol{d})$ is given by the symmetrized gradient of displacements,

$$\boldsymbol{e}(\boldsymbol{d}) = \frac{1}{2}\left(\nabla \boldsymbol{d} + \nabla \boldsymbol{d}^T\right) \in \mathbb{R}^{3\times 3}, \tag{12.4}$$

and the stress $\boldsymbol{A}\boldsymbol{e}(\boldsymbol{d})$ is given by

$$\begin{aligned} \boldsymbol{A}\boldsymbol{e}(\boldsymbol{d}) &= 2\mu \boldsymbol{e}(\boldsymbol{d}) + \lambda \operatorname{trace}(\boldsymbol{e}(\boldsymbol{d}))\boldsymbol{I} \\ &= 2\mu \boldsymbol{e}(\boldsymbol{d}) + \lambda \operatorname{div}(\boldsymbol{d})\boldsymbol{I}. \end{aligned} \tag{12.5}$$

Here, $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$ are the Lame constants and $\boldsymbol{I}$ is the identity matrix.

Equation (12.3) is the strong formulation for linear static elasticity. A Galerkin discretization of the weak formulation (typically by finite elements) yields a linear system

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}, \quad \boldsymbol{K} \in \mathbb{R}^{n\times n}, \quad \boldsymbol{d}, \boldsymbol{f} \in \mathbb{R}^n, \tag{12.6}$$

where n is the dimension of the ansatz space, $\boldsymbol{K}$ is the stiffness matrix, and, by abuse of notation, $\boldsymbol{d}$ is the vector of displacements and $\boldsymbol{f}$ the vector of acting forces.

In the dynamic case, i. e., when $\boldsymbol{d}$ and $\boldsymbol{f}$ are time-dependent, equation (12.6) is extended to [113]

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \boldsymbol{D}(t)\dot{\boldsymbol{d}} + \boldsymbol{K}(t)\boldsymbol{d} = \boldsymbol{f}(t), \tag{12.7}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is the mass matrix and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is the damping matrix. Note that both $\boldsymbol{D}$ and $\boldsymbol{K}$ may be time-dependent. An important special case of this is the rotor dynamic equation

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \left(\boldsymbol{D}(\omega) + \omega \boldsymbol{G}\right)\dot{\boldsymbol{d}} + \boldsymbol{K}(\omega)\boldsymbol{d} = \boldsymbol{f}(t, \omega), \tag{12.8}$$

where ω denotes the angular velocity of the rotor and $\boldsymbol{G}$ is the so-called gyroscopic matrix [36].

12.6.3 Model order reduction

In many real-world applications, the number n of degrees of freedom of the discretized system (12.7) or (12.8) is large and its numerical integration is not possible in real-time. MOR strategies introduce a reduced state $\boldsymbol{q} \in \mathbb{R}^r$ with $r \ll n$, via $\boldsymbol{d} = \boldsymbol{\Psi}\boldsymbol{q}$, $\boldsymbol{\Psi} \in \mathbb{R}^{n\times r}$.

One way to obtain the reduction matrix $\boldsymbol{\Psi}$ for system (12.7) is to use modal reduction. Setting up the eigenvalue problem of equation (12.7)

$$\omega^2 \boldsymbol{M}\boldsymbol{\theta} = \boldsymbol{K}\boldsymbol{\theta} \tag{12.9}$$

and taking the first r eigenvectors, the matrix $\boldsymbol{\Psi}$ may be defined by

$$\boldsymbol{\Psi} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r\}. \tag{12.10}$$

A preferable technique may be the Krylov subspace methods [7, 93]. The subspace is defined by

$$\boldsymbol{\Psi} = \{\boldsymbol{K}_\omega^{-1}\boldsymbol{f}, \boldsymbol{K}_\omega^{-1}\boldsymbol{M}\boldsymbol{K}_\omega^{-1}\boldsymbol{f}, \ldots, (\boldsymbol{K}_\omega^{-1}\boldsymbol{M})^{r-1}\boldsymbol{K}_\omega^{-1}\boldsymbol{f}\}. \tag{12.11}$$

The Krylov basis may be computed by the Arnoldi algorithm, which delivers an orthonormal basis of the subset.

Inserting this into (12.7) and multiplying by $\boldsymbol{\Psi}^T$, one obtains the *reduced equation*

$$\hat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{D}}(t)\dot{\boldsymbol{q}} + \hat{\boldsymbol{K}}(t)\boldsymbol{q} = \boldsymbol{\Psi}^T\boldsymbol{f}(t), \tag{12.12}$$

where

$$\hat{\boldsymbol{M}} = \boldsymbol{\Psi}^T\boldsymbol{M}\boldsymbol{\Psi}, \tag{12.13}$$

$$\hat{\boldsymbol{D}} = \boldsymbol{\Psi}^T\boldsymbol{D}\boldsymbol{\Psi}, \tag{12.14}$$

$$\hat{\boldsymbol{K}} = \boldsymbol{\Psi}^T\boldsymbol{K}\boldsymbol{\Psi} \in \mathbb{R}^{r\times r} \tag{12.15}$$

are the reduced matrices. In the case of rotor dynamics, $\boldsymbol{D}$ and $\boldsymbol{K}$ depend on ω. In the ramp-up phase of an electric engine, where the rotation frequency increases, the reduction operations (12.14) and (12.15) have to be performed in each time step. Interpolation schemes may reduce the computational effort. In the constant phase of the engine, also the reduced matrices remain constant.

As filtering methods are generally applied to first-order equations, we define as usual

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{pmatrix}, \quad \boldsymbol{u}(t) = \begin{pmatrix} \boldsymbol{0} \\ \dot{\boldsymbol{f}}(t) \end{pmatrix}, \tag{12.16}$$

and, assuming that $\hat{\boldsymbol{M}}$ is invertible,

$$\boldsymbol{A}(t) = \begin{pmatrix} 0 & \boldsymbol{I} \\ \hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{K}}(t) & \hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{D}}(t) \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} 0 \\ \boldsymbol{\Psi}^T \end{pmatrix}, \tag{12.17}$$

so that we obtain the equivalent first-order system in state-space form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}(t). \tag{12.18}$$

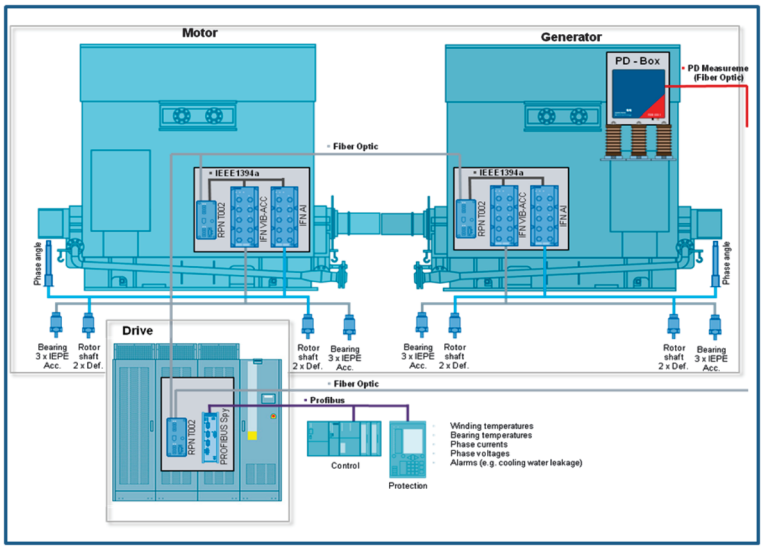


Figure 12.6: Electric engine/generator configuration of about 20 MW.

12.6.4 Fault detection in terms of unbalance of a rotor

Following [13], [64], and [63] we want to identify an unbalance of a rotor during operation. The test configuration is an electric engine which drives directly a generator (Figure 12.6). The two rotors are connected by a clutch. Starting point for the analysis is the rotor dynamic model which was used in the design process of this particular drivetrain (Figure 12.7). According to the strategies described in Section 12.6.3, we reduced the model in order to obtain real-time capability. In order to obtain realistic frequencies for the model, also some nonlinearities in terms of the fluid bearings have to be considered. Therefore, a nested procedure was applied [86], which keeps the nonlinear parts at the bearings and reduces the linear parts in term of the motor and the generator. Both rotors from the considered drivetrain are equipped with four discrete planes meant for balancing the rotor. The validation of the digital twin was done by physically attaching a small test weight to one of the balancing planes. Four sensors located at the bearings (red bars in Figure 12.7) provided the measurement data which are compared to the simulation results.

The comparison or identification was performed by an augmented nonlinear Kalman filter procedure. The unbalance itself enters the model in terms of an external force (see equation (12.7)), where location, orientation, and magnitude are identified by the filtering algorithm. The result of the identification method is presented in Figure 12.8. The blue peak in Figure 12.8 presents the location and the amount of unbalance.

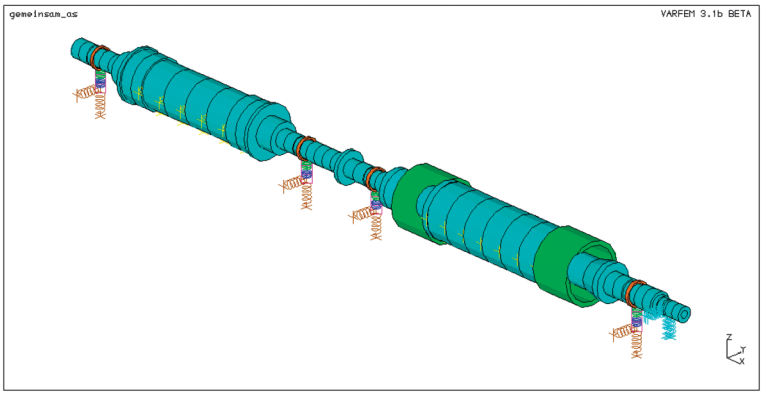


Figure 12.7: Model of the electric engine/generator configuration.

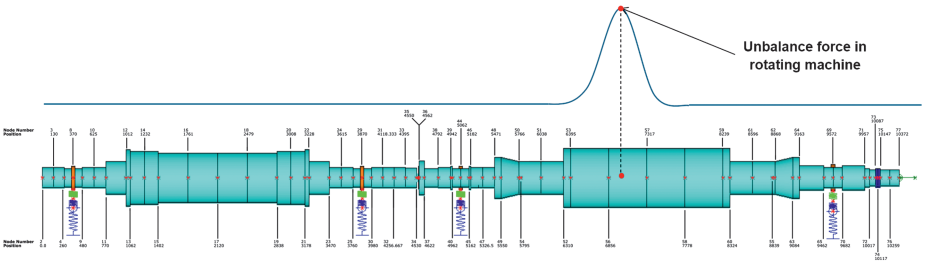


Figure 12.8: Unbalance detection of the rotor during operation.

12.6.5 Summary

By combining MOR techniques and nonlinear identification methods, a digital twin for detecting and localizing faults (in terms of unbalances) has been developed for rotating systems. Further efforts are made in order to predict the increase of vibration during operation. The time a critical vibration is achieved defines the moment for scheduling an inspection. Knowing this time at an early stage, an inspection and spare part supply can be prepared.

12.7 Use case – operation control

12.7.1 Engineering controlled systems

The product race has become an innovation race, reconciling challenges of branding, performance, time-to-market, and competitive pricing while complying with ecological, safety, and legislation constraints. The answer lies in “smart” products of high

complexity, relying on heterogeneous technologies and involving active components. The corresponding design and engineering process hence must take the integration of control functions in the product explicitly into account. This adds an important additional complexity to the design engineering process where the interaction between the control and the system requires these should be optimized concurrently. The current industrial practice however still treats passive system design and controller design as different and separate design loops with their own models and their own validation and verification strategies. Suboptimal designs and unexpected integration problems are the result. Not reusing the wealth of engineering models available from earlier detailed system design stages furthermore leads to inconsistency problems and ineffective engineering processes. Closing this gap offers a significant potential for optimized designs, better product performance, and fewer and shorter design iteration cycles [1, 106].

12.7.2 Technological challenges

Two classes of challenges can be distinguished in relation to the integration of control functions in the product. The first one targets the optimization of the controller architectures, strategies, and settings for a controlled product, hereby using a system or "plant" model in a virtual controller optimization process. The second challenge targets the design of an optimal control solution by including a model of the controlled system into the controller itself, for example in a model predictive control (MPC) approach [82]. For both cases, the used system models are typically developed dedicated for the control application taking into account feasible complexities and system simplifications. One objective to do so is to allow fast virtual testing and optimization cycles and to enable hardware validations in physical control environments. The detailed (for example multiphysics) design engineering models from the system design departments are typically not reused. The main reason for such a suboptimal approach is that either detailed system models from the system engineering design departments are not available in the control department or are of too high complexity to be used together with control simulation.

Focusing on the first challenge of optimizing the controller design, one can distinguish three phases:

Phase 1: The combination of the multiphysics simulation model with that of the controller enables the design of the control logic and the performance engineering of the intelligent system. This is referred to as "model-in-the-loop" (MIL). The simulation is offline, i. e., there is no requirement for real-time performance of the simulation. Basically, two interconnecting objectives can be distinguished: One is to perform systems engineering based on the multiphysics "plant" model, including the representation of (often simplified or idealized) control in the multiphysics model; the other is to perform control engineering, using a model of the system to be controlled ("plant

model"). The first objective, for example, serves the purpose of configuration design (how many actuators and sensors, where to place them, etc.) or concept evaluation studies or the optimization of the mechanical system design taking into account the presence of control and certain control laws. The second objective is oriented towards the development of the optimal control logic and the development and verification of control hardware, control libraries, and embedded software up to the validation and calibration of the control system on the electronic control unit (ECU) (Figure 12.9) [106].

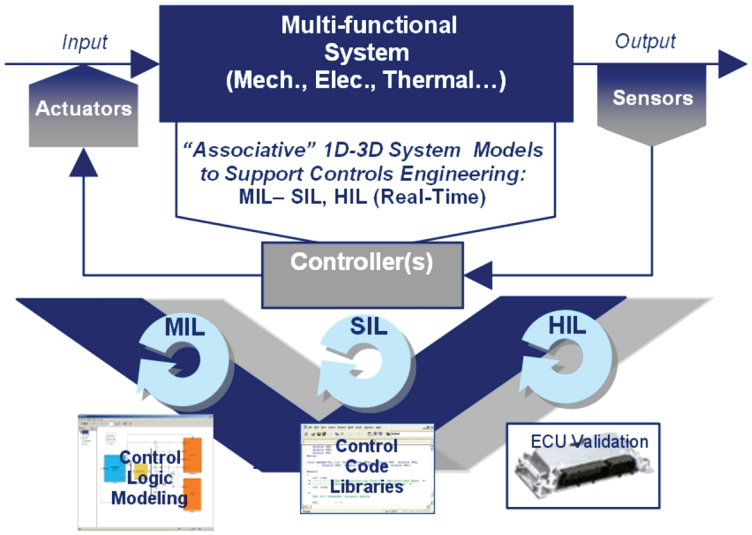


Figure 12.9: Associative plant models for control engineering.

To couple the models, different approaches exist. One may embed state equations with a description of the plant system (e. g., multibody simulation models or one-dimensional ordinary differential equation-based system simulation models) into these of the control (or vice versa) to enable the use of one solver, or adopt a true co-simulation approach where each system part runs its own solver [106, 40].

Alternatively, or in combination with the above approaches, a reduction of the plant model (e. g., a finite element or complex, even nonlinear multibody simulation model) into a description compatible with the controller model (e. g., state-space formulation) can be used. The model reduction step mostly achieves its goals at the expense of the full observability and/or controllability of the physical phenomena, leading to a macroscopic "equivalence" but loosing direct insight in the microscopic observation domain. The challenge is to develop model compression methodologies that allow maintaining a relation with the physical meaning of model parameters. Such

co-simulation and model reduction approaches are used both for MIL applications for systems engineering and for control logic engineering.

Phase 2: The next step is the development and optimization of the "embedded" control software. This needs also to be done in the context of the functioning of the multiphysics system to be controlled. This is referred to as "software-in-the-loop." While some of this can be done in offline simulation (provided software libraries of the controller are available), the final optimization needs to take into account the working of the software in real-time, requiring real-time capable multiphysics simulation models.

Phase 3: The final testing and calibration of the controller software and hardware requires the controller to be connected to a multiphysics simulation model of the components, subsystems or system, in a dedicated computing environment that is referred to as "hardware-in-the-loop" [6]; of course, this requires real-time capable simulation models.

The use of MOR is hence a key factor for enabling a true model-based engineering approach where consistent engineering models can be used throughout the various design phases. The applied MOR methods may depend on the reduction purpose and the nature of the master models. For example, in mechatronics systems, these can be finite element, multibody, or multiphysics models which can be linear or weakly or strongly nonlinear. Two application cases will be briefly discussed.

12.7.3 Application to the design of an active sound quality control system

Active noise reduction (and sound shaping) is a widely studied research topic with many potential industrial applications. Structural-acoustic solutions using smart materials as sensor and/or actuators are explored, enabling intelligent structures. Such solutions are however typically developed as add-on systems which prevents optimizing their potential impact as part of the overall system. A model-based mechatronic engineering approach was developed to enable an integrated solution [22, 23]. It was applied to the active sound control of vehicle engine noise to the car interior. The challenge consisted of relating the large three-dimensional, frequency-domain (finite element- and boundary element-based) vibro-acoustic, and structural models for the vehicle structure and structural components, interior vehicle cavities, and exterior propagation field, with models of smart material sensors and actuators and a time-domain control model. A simplified vehicle structure was developed to allow experimental validation. It consisted of a concrete car body, an engine compartment with an artificial source, and a flexible firewall panel on which the structural-acoustic control was to be applied with piezo-elements. The rigid walls of the concrete car structure can be easily modeled and also treated with a dedicated damping surface (Figure 12.10).

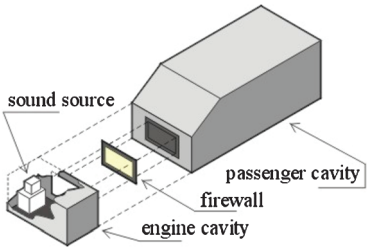


Figure 12.10: Simplified car structure.

MOR was a key element in the modeling approach, allowing to incorporate the reduced model as a plant model in the controller simulation. The component mode synthesis (CMS) approach was used. Very large reduction factors were used, reducing the large structural/vibro-acoustic finite element model (25,000 acoustic degrees of freedom but which can overall easily reach hundreds of thousands of degrees of freedom when multiple flexible panels are included) to a time-domain state-space model of realistic size (200 degrees of freedom). The sensors and actuators were represented by one-dimensional models for their functional performance, while their added mass and stiffness are accounted for in the three-dimensional finite element models. The acoustic propagation was related to the structural outputs by means of an "acoustic transfer vector" approach. The modeling approach included the following steps using multiple software tools:

- generate structural mesh and apply material properties (finite element preprocessor);
- add actuator and sensor mechanical models (finite element preprocessor);
- run a modal analysis (finite element analysis);
- build the acoustic finite element model and perform modal analysis (finite element analysis);
- import the structural model and couple it with the acoustic one (finite element analysis);
- calculate actuator and sensor electromechanical coupling (extended finite element analysis);
- reduce and convert the finite element model into a state-space model (MATLAB);
- implement and optimize the controller with the coupled state-space model (MATLAB/Simulink).

The coupling between acoustic and structural models is shown in Figure 12.11.

After performing a coupled modal analysis, the desired degrees of freedom are taken to derive the state-space model. In this case, the state-space model features two inputs (one actuator on the firewall and a sound source in the engine compartment) and four outputs (three pressures in the passenger compartment and one velocity on the firewall). The state-space model derived from this coupled approach allows the implementation of any controller involving the predefined degrees of freedom, and

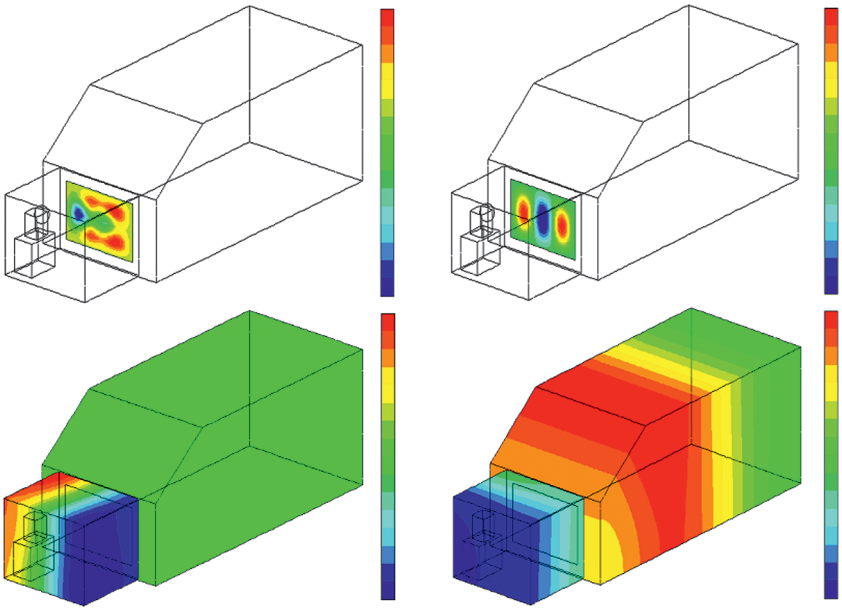


Figure 12.11: Coupled structural acoustic models to be reduced to a modal basis.

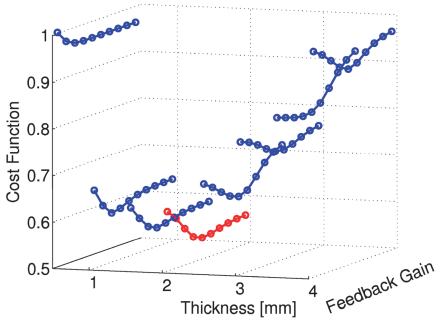


Figure 12.12: Multiattribute cost function for combined mechanical and control parameters.

if the finite element approach involves the systematic representation of the sensors and actuators, the resultant state-space model is, in fact, a representation of the fully coupled electro-vibro-acoustic system, with any possible input-output relationships allowed by the chosen degrees of freedom.

Using this model, an optimization procedure is performed. The cost function takes into account the sound pressure level at the drivers head, the actuator input energy, and the weight of the solution. The firewall thickness and the velocity feedback controller gain are the variables. The position of the collocated sensor/actuator pair (SAP) can be considered fixed or included in the optimization loop. Figures 12.12 shows the cost function for each thickness as a function of the feedback gain, on the best SAP position for each case.

The best SAP position and optimal feedback gain depend on the thickness, which indicates that the global optimum can only be achieved in such a concurrent design between the active and passive system characteristics, proving the effectiveness of an integrated mechatronics simulation approach. The same model can be used to evaluate different controller strategies such as combined feedforward/feedback, filtered-X LMS, and NEX-LMS, and for different performance targets (noise level optimization and/or sound quality control). A more extensive discussion of the various modeling aspects and the detailed optimization procedures can be found in [22, 23].

12.7.4 Application to control development using neural network-based model reduction

The development of the control of mechatronic systems becomes more complex when multiple actuators and sensors are interfacing with a highly dynamic multiphysical system, often not having the sensors available to develop an optimal control. In the automotive industry, controls have mostly been developed using rule-based methods, first by directly writing code, later using a model-based approach. In the automotive industry, the complex balancing of multiple performances of the combustion engine such as emissions, fuel economy, and acceleration performance have increased the complexity of the engine actuators. To develop the controls for such complex mechatronic systems, new methods are required. Optimal control such as MPC in combination with methods to predict virtual controllable quantities using state estimation technologies in combination with Kalman filtering are examples of such new technologies that start to find their entry in the automotive world. MPC and Kalman filtering-based state estimation require, however, models that run fast while keeping a certain level of accuracy. Often, ad hoc simplified models are (re)developed, neglecting the availability of detailed engineering models. Reusing such models would not only save modeling time but would also allow better consistency of the various design engineering models over the different attributes and product versions and variants.

The simulation models for designing mechatronic systems are often created based on a combination of detailed three-dimensional models and test data and have a high level of accuracy but have a too slow calculation time to be used in MPC or state estimation methodologies in real-time on an ECU. To be able to convert such system models into the context of optimal control in combination with virtual sensing, neural networks can be an ideal methodology to develop a control model directly from the detailed plant model to be controlled [50, 65]. Figure 12.13 explains the different steps in the process, showing the reduction of the detailed engine model to a neural network-based model for the virtual sensor as well as the optimal controller. The neural network ROM allows real-time execution of the model for both the sensing and the control action. Initial results using the controls model in closed loop with the detailed plant model for tracking purposes indicate that the performance, for scenarios

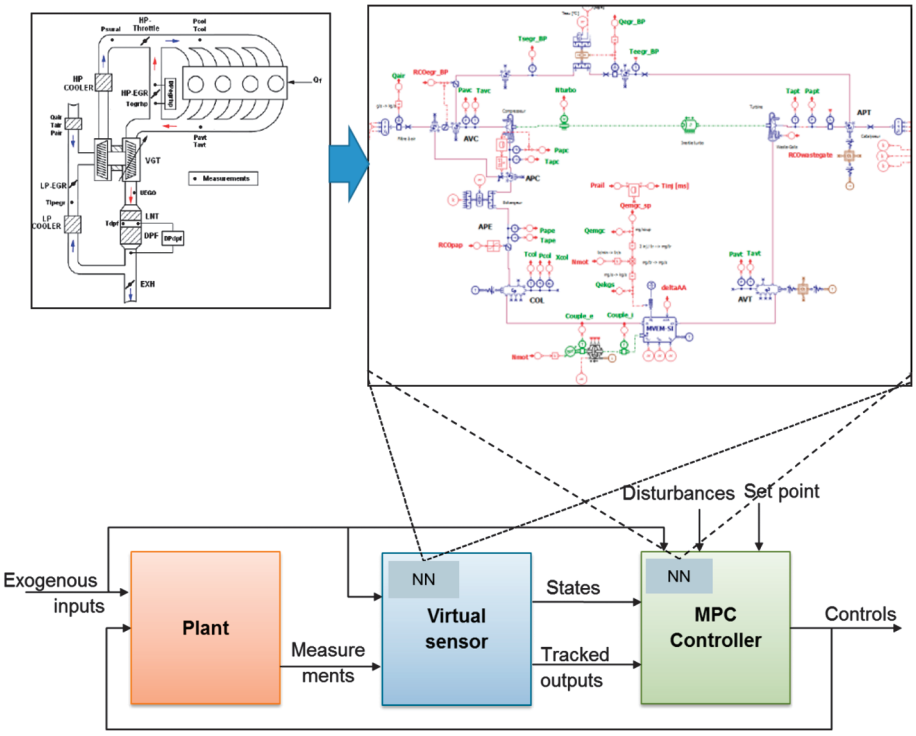


Figure 12.13: Advanced combustion engine (upper left), one-dimensional system model (upper right), and reduction by neural networks in the real-time virtual sensor and MPC controller models.

for which the neural network is not trained, remains within good accuracy as long as the important states of the model are kept observable. Further results in this field will bring more clarity in how broadly this technology can be used for engine controls or other advanced vehicle controllers.

By applying multiple load cycles covering the full operating space of the system, the neural networks can be trained to represent the relevant system behavior even in the case of strongly nonlinear system characteristics.

12.7.5 Summary

Model-based approaches find increasingly their way into the design of (optimal) control systems. In the majority of cases, however, the applied system models ("plant models") are ad hoc developed low-complexity models that are not correlated to the design engineering models developed in the mechanical design stages. This not only leads to inefficient processes redoing efforts that could be recovered, but also gives rise to major issues related to consistency and traceability when design improvements,

versions, or variants are to be processed. MOR can offer an answer for both the control design and control implementation and opens up new opportunities for concurrent design of the mechanical and the control system. Major challenges are still presented by the very large reductions factors to allow fast control optimization or even real-time usage inside state estimators or MPC controllers. The use of a neural network-based approach to reduce complex nonlinear models subject to an envelope of operating conditions offers significant potential that is to be further investigated.

12.8 Use case – drivetrain analysis

12.8.1 MOR for contact mechanics problems

The *simulation of dynamical systems involving contacts between elastic bodies* [112] is a challenging and active research field on its own. In particular high-frequency phenomena, numerical stiffness, high degree of nonlinearity, high dimensionality, and multidisciplinary nature (mechanics, acoustics, fluid dynamics, tribology) are among the major challenges that researchers and software developers must address in order to efficiently solve these types of problems [11]. Despite its seemingly niche description, there is a wide range of applications in which these problems are found and need to be solved. In particular, the simulation of geared transmissions or drivetrains is practically ubiquitous if one has to deal with simulations of electromechanical machines. Drivetrains contain a multitude of components, including bearings, gears, clutches, and spline connections that are known to behave nonlinearly and contain multiple contacts between flexible objects. While several MOR methods have been applied largely and successfully [32] in the field of flexible multibody simulations [97] in both academic and industrial settings with a large growing body of literature, MOR methods dedicated to the field of contact mechanics have been only recently explored [12, 103]. Moreover the developed methods often target high-dynamic contact mechanics simulations with fully flexible bodies and dynamic interactions with flexible eigenmodes of the structure [12]. These problems would indeed remain practically intractable without the usage of MOR and/or computer clusters. On the other hand, a large set of system-level related problems (such as large drivetrains or more complex machines containing several drivetrains) might not need the level of fidelity and the still relatively large computational times necessary to solve a fully nonlinear dynamic problem. They might instead still benefit from MOR solutions that speed up simulation times and decrease memory usage and disk storage, while still retaining the required level of fidelity on both system and component levels (Section 12.2).

The method discussed in this chapter exploits both an advanced MOR strategy and physics-based considerations coming from the targeted application domain of drivetrain simulation and combines them. The result is a numerical strategy that is

efficient and computes accurately most of the drivetrain dynamics-related scenarios that are industrially relevant. The focus is on three-dimensional system-level simulations including lightweight and internal gears and noise-vibration and harshness (NVH) problems in a multibody simulation environment.

12.8.2 Technological challenges

While the application challenge is relatively straightforward to summarize – *efficiently and accurately solve three-dimensional multibody problems involving multiple gear contacts for system-level and NVH purposes* – the technical challenges connected to it are multiple and have their root in the mathematical description of the equations of motion of a multibody system. The following set of equations is an index 3 differential algebraic equation describing the dynamic motion of a flexible multibody problem:

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} + \boldsymbol{G}^{\boldsymbol{T}}(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{f}_{\mathbf{ext}} + \boldsymbol{f}_{\boldsymbol{v}}, \tag{12.19}$$

where $\boldsymbol{M}(\boldsymbol{x})$ is the nonlinear mass matrix, $\boldsymbol{K}$ is the linear stiffness matrix, $\boldsymbol{x}$ is the vector of generalized coordinates, $\boldsymbol{G}$ is the Jacobian of the constraints, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, and $\boldsymbol{f}_{\mathbf{ext}}$ and $\boldsymbol{f}_{\boldsymbol{v}}$ are the vectors of external forces and the quadratic velocity terms. Without entering in more details (which can be found in, e. g., [11]) we mention that despite being a fully nonlinear problem, the equations describing the deformation of the flexible bodies present in the system can be reduced by using linear MOR methods for second-order systems in the following form:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{D}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{12.20}$$

where $\boldsymbol{M}$, $\boldsymbol{D}$, $\boldsymbol{K}$ are the linear mass, damping, and stiffness matrix of the underlying finite element model, $\boldsymbol{u}$ is the vector of linear nodal deformations, and $\boldsymbol{f}$ is the vector of nodal forces. This system can be reduced thanks to Petrov–Galerkin projection methods:

$$\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{M}\boldsymbol{V}\ddot{\boldsymbol{u}} + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{V}\dot{\boldsymbol{u}} + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{K}\boldsymbol{V}\boldsymbol{q}\boldsymbol{u} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{f}, \tag{12.21}$$

where $\boldsymbol{W}$ and $\boldsymbol{V}$ are the left and right subspaces used to obtained the reduced system. Galerkin methods can be used in flexible multibody problems within the assumption of large gross motion but small deformations within each of the body frames. This assumption is amply satisfied in problems involving gear contacts.

While flexible bodies that are not involved in contact interactions can be reduced with very efficient techniques, such as balanced truncation [83, 42], Krylov [32], CMS [19], etc., flexible bodies that include contact interactions suffer from the so-called *interface* problem [102]. In practice, a very large and inefficient reduction space needs to be used for an accurate reduction. The size of the reduction space becomes proportional to the amount of degrees of freedom potentially involved in the contact interactions. This causes problems such as high memory usage, large precomputation time,

large storage requirements, and increased numerical stiffness. Finally, the contact detection phase is also computationally very costly and scales with the (large) number of degrees of freedom that can be involved during contact.

In recent years several methods have been presented to maintain a high level of accuracy – similar to nonlinear finite element full-order dynamic computations – but drastically limit the impact of the above-mentioned issues. In particular, the following works [11, 12] obtain very good results in terms of speedup and memory usage while losing only a fraction of the accuracy obtained with nonlinear finite element problems. The field of hyperreduction [18, 31] is also exploited to tackle the contact detection problem with very promising results. However, this methodology is relatively complex to include in proprietary multipurpose multibody solvers and, moreover, the simulation time and the user expertise needed to use these techniques do not match the requirements of system-level three-dimensional multibody software. In order to develop a novel method for gear contact simulation to be included in a commercial multibody solver, the following decisions have been taken based on the available state of the art:

- **Contact detection:** Hyperreduction for contact detection is still in its infancy and needs further development. For this reason the computational performances are improved by using geometrical considerations that are related to the specific application field of drivetrain dynamics.
- **Dynamic flexibility:** The majority of the applications that involve dynamic simulations must include the modal behavior of the full drivetrain but the eigenfrequencies related to gears bodies and teeth themselves are outside of the frequency range of interest for many applications. From this point of view, it was decided to concentrate on a correct representation of the contact stiffness to properly represent the quasi-static behavior of the gear contact and the overall three-dimensional system-level dynamics. The accurate evaluation of the contact stiffness is of paramount importance in the definition of the dynamic modes of the full drivetrain.
- **Contact stiffness formulation:** The contact interactions of finite element meshes require a very fine spatial discretization to properly capture the correct Hertzian nonlinear behavior during contact. For this reason it was decided to focus the attention on the development of a method that combines the advantages of both general finite element formulations but exploits analytical formulas near the contact regions.

While the first point is important but out of the scope of this chapter, the second and third bullets highlight how available techniques that can be found in the literature [3] have been enhanced thanks to a cooperation between Siemens PLM Software and the Mechanical Engineering Department of KU Leuven (Section 12.4.2). These developments lead to a method based on MOR [102, 12, 16] that allows to describe in an accurate way the gear contact stiffness, including effects such as gear body deforma-

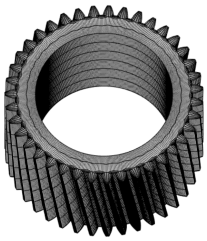


Figure 12.14: Finite element mesh of a spur gear.

tion, Hertzian nonlinear stiffness, and teeth convective couplings, while remaining extremely efficient. The reasons for the efficiency and accuracy of the method are:

- **Efficiency:** The problem is treated quasi-statically; thanks to the MOR technique, very few degrees of freedom are retained to describe the teeth deformation. When possible, potential symmetries in the gears geometry are also exploited for efficiency purposes.
- **Accuracy:** Despite the application of MOR, the contact interaction is statically *quasi-exact* with respect to the full-order finite element model. *Convective* deformation terms that couple the deformation of different teeth are accurately retained. Local dynamic effects such as teeth dynamic vibrations that are less often of relevance during standard operations are instead discarded. While these dynamic effects might be relevant for problems such as dynamic ring gears excitations and high-speed applications, the method is implemented in a modular way so that future extensions are efficient to implement.

12.8.3 User-related challenges

The developed technology based on MOR achieves the objectives targeted at the beginning of the development but a key component still needs to be addressed to allow a smooth user experience and limit the amount of expertise needed to use the method: *usability* (Section 12.4). For this reason the parameters that control the level of *static completeness* of the reduction space can be adjusted with a single parameter that ranges between *zero* and *one* where the limit value of *one* represents exact static completeness at the expense of some longer preprocessing time and slightly slower simulations while lower values allow to obtain a trade-off between accuracy and speed. Moreover the user is provided with a parametric mesher that automatically creates the gears finite element meshes based on a few parameters (Figure 12.14) that is used for the automatic generation of the reduction space while further minimizing the user intervention.

The interfaces between the MOR method proposed and the multibody solver are integrated into a user-friendly *application-driven* user interface – the *Simcenter 3D*

Transmission Builder (Figure 12.15) – that proposes also a simplified work flow for the creation of complex drivetrains. Practically, thanks to the dedicated *Simcenter 3D Transmission Builder* interface and application-specific choices related to the MOR technique it was possible to obtain a seamless and user-friendly usage of advanced MOR numerical techniques available to nonexpert users.

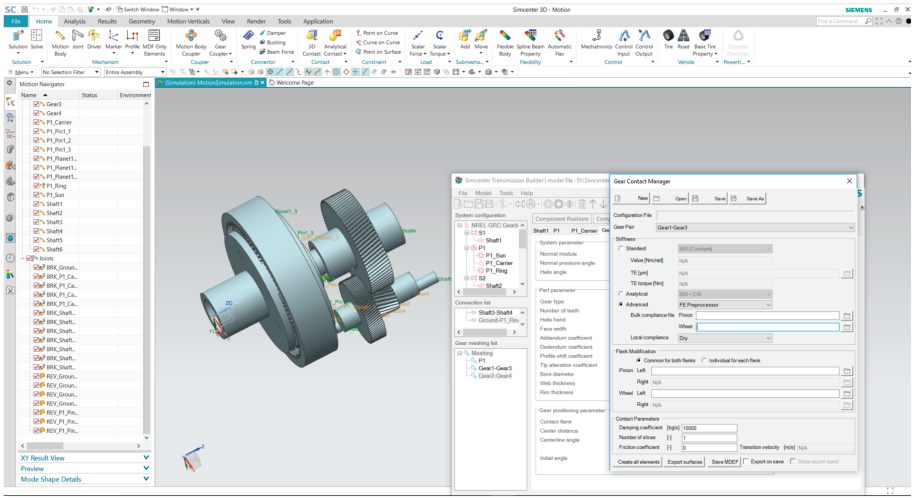


Figure 12.15: Simcenter 3D Motion – Transmission Builder.

12.8.4 Validation of MOR for drivetrains against experimental results

The described numerical method based on MOR is released as a product in Simcenter 3D Motion under the name of *Advanced–FE preprocessor* gear contact. Before the product release, given both the complexity of the method and the number of assumptions made during development, the methodology has been validated using multiple numerical and experimental results. In this chapter we present a subset of the experimental validation results to show the accuracy of the proposed method. For a larger set of examples we refer to [85]. The validation process has been carried out thanks to the usage of an in-house precision gear test rig [74] jointly developed by Siemens PLM Software, KU Leuven, and the University of Calabria. The test rig has been designed and manufactured to assess typical gear-related physical quantities in static and dynamic conditions, under imposed conditions of misalignment and shaft compliances. Particular attention is given to the measurement of gear pair transmission error (TE) [75], which is a typical key performance indicator used for the assessment of drivetrain NVH performances.

As an illustrative example we present validation results related to the complex case of gear contact between two spur gears, including large friction, microgeometry modifications, and different loading conditions. The measured TE is shown in Figure 12.16. It can be seen that despite the wide range of torques applied, the proposed method is able to match the TE with a high degree of accuracy. This is particularly striking since the effects of microgeometry, friction (as noticeable in the discontinuous jumps in the TE), teeth flexibility, and local contact nonlinearities are highly interacting with each other. The experimental results and multiple numerical validations carried out confirmed the good performances of the method in terms of both accuracy and speed.

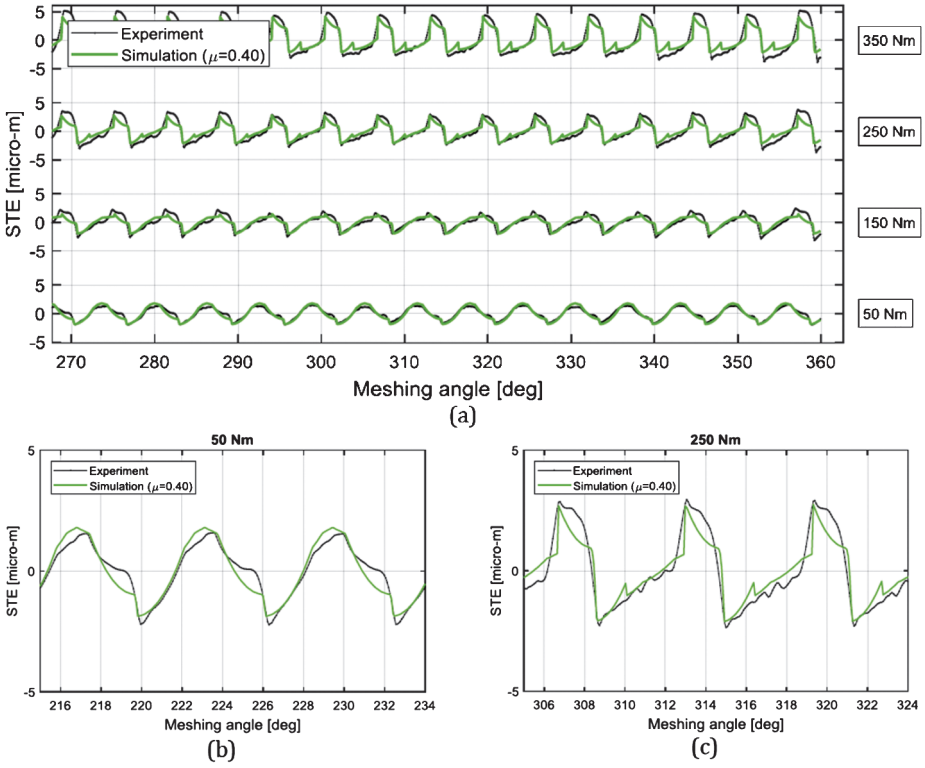


Figure 12.16: Experimental–numerical comparison of the proposed approach for TE evaluation.

12.9 Use case – lifetime analysis

In this section, we consider lifetime analysis for railway axles as an example for a digital twin in the design phase. The engineer would like to know already in the design

phase how lifetime depends on the inspection scheme, the inspection interval, the type of load, and the fatigue crack growth. Stochastic MOR serves as a key element for the realization of this digital twin.

12.9.1 Efficient computation of failure probabilities

An important part of lifetime analysis is the computation of small failure probabilities, which is a challenge for practical problems with CPU time-intensive function calls. As already pointed out in Chapter 10 of this volume of *Model order reduction*, Section 10.2.3, failure probabilities are described by multidimensional probabilistic integrals which may be discretized by a multivariate quadrature rule. In the case of failure probabilities, the integrand of the probabilistic integral is discontinuous such that common quadrature rules will not provide sufficient accuracy. Here the key idea is to reformulate the integral into an integral with a smooth integrand and then to apply the unscented Kalman filter (UKF) [51] for evaluating the integral. The reliability of a product, system, or process is often indicated by a failure function:

$$g(\mathbf{x}) = \begin{cases} \leq 0 & \text{unsafe,} \\ > 0 & \text{safe,} \end{cases} \tag{12.22}$$

where $\mathbf{x}$ is the vector of stochastic variables. The failure function $g(\mathbf{x})$ describes a damage mechanism, e. g.,

1. the mechanical stress or the temperature exceeds a given threshold (finite element analysis, computational fluid dynamics);
2. the amplitude of oscillations exceeds a given threshold (linear and nonlinear modal frequency analysis);
3. changes of the microstructure of the material as a prestage of cracks differs from a given pattern (stochastic Voronoi techniques, finite element method);
4. the crack size exceeds a given length (finite element method + crack size analysis);
5. a chemical species exceeds a given concentration (computational fluid dynamics, ChemKin).

Using this failure function, the failure probability reads

$$P(g(\mathbf{x}) \leq 0) = \int\limits_{g(\mathbf{x})\leq 0} \rho_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \tag{12.23}$$

where $\rho_{\mathbf{x}}(\mathbf{x})$ is the stochastic density of $\mathbf{x}$. We restrict our presentation to the case of independent standard normally distributed variables $\mathbf{x} = (x_1, \ldots, x_n)^T$. In the general case of a failure function depending on nonnormally distributed variables $\tilde{x}_i$, e. g., the Rosenblatt transformation may be applied [90], mapping $\tilde{x}_i$ to x_i for $i = 1, \ldots, n$. In

order to obtain an integral over $\mathbb{R}^n$, an indicator function is introduced,

$$P(g(\mathbf{x}) \leq 0) = \int_{\mathbb{R}^n} \Gamma_g(\mathbf{x}) \rho_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \tag{12.24}$$

where

$$\Gamma_g(\mathbf{x}) = \begin{cases} 0 & g(\mathbf{x}) > 0, \\ 1 & g(\mathbf{x}) \leq 0. \end{cases} \tag{12.25}$$

For practical applications, the Monte Carlo method is too expensive to evaluate the integral in (12.24) because of the high computing time. So a stochastic MOR method is required. A standard method for approximation of the integral in (12.24) is the first-order reliability method (FORM) [48, 47]. This method first computes the so-called beta point (the point of highest failure probability) and then constructs a linear approximation of the failure function in the beta point. For highly nonlinear failure functions g, the failure probability of the FORM will not be accurate. An extension of the FORM, the second-order reliability method [14], is more accurate but requires second-order derivatives, which are often not available in practical applications. Because of the discontinuous integrand in (12.24), standard quadrature formulas will in general lead to bad approximation properties. Our stochastic model order method now consists of a reformulation of (12.24) in an integral with a continuous integrand and subsequent application of a nonlinear filter method. The reformulation is possible if the failure function $g(\mathbf{x})$ is continuously differentiable in its coordinates and strictly monotone in at least one coordinate (backmapping approach); see [110]. Without loss of generality, let $g(\mathbf{x})$ be monotone in x_n. It follows that the critical $\bar{x}_n$ defining the limit state can be expressed as a function of $x_1, \ldots, x_{n-1}$:

$$\begin{aligned} \bar{x}_n &= \zeta(x_1, \ldots, x_{n-1}), \\ 0 &= g(x_1, \ldots, x_{n-1}, \bar{x}_n). \end{aligned} \tag{12.26}$$

Then the failure integral (12.24) reads

$$\begin{aligned} &P(g(\mathbf{x}) \leq 0) \\ &\quad = \int_{\mathbb{R}^{n-1}} \rho_1(x_1) \cdots \rho_{n-1}(x_{n-1}) \int_{\bar{x}}^{\infty} \rho_n(x_n) dx_1 \ldots dx_n \\ &\quad = \int_{\mathbb{R}^{n-1}} \rho_1(x_1) \cdots \rho_{n-1}(x_{n-1}) h(x_1, \ldots, x_{n-1}) dx_1 \ldots dx_{n-1}, \end{aligned} \tag{12.27}$$

with

$$h(x_1, \ldots, x_{n-1}) = \frac{1}{2} \operatorname{erfc}\left(\frac{\zeta(x_1, \ldots, x_{n-1})}{\sqrt{2}} \right).$$

The smoothness of h follows from the implicit function theorem. The failure probability $P(g(\mathbf{x}) \leq 0)$ can thus be interpreted as the mean of the smooth function h. For evaluation of this mean a nonlinear filter method, called UKF [51], is applied. This filter can be used to estimate the mean and covariance of a nonlinear stochastic process $f(\boldsymbol{w})$, where $\boldsymbol{w} \in \mathbb{R}^{n_w}$ is a normally distributed random vector with mean $E(\boldsymbol{w})$ and covariance $P^{ww} \in \mathbb{R}^{n_w \times n_w}$. So-called sigma points $\mathcal{X}^{(i)}$, together with weights W_i^{mean} and W_i^{cov}, are constructed and mapped to $\mathcal{Z}^{(i)} = f(\mathcal{X}^{(i)})$ for $i = 0, \ldots, p$. The unscented filter then yields an approximation of the mean μ and covariance P^{zz} of the nonlinear function by

$$E(z) \approx \sum_{i=0}^{p} W_i^{\text{mean}} \mathcal{Z}^{(i)},$$

$$P^{zz} \approx \sum_{i=0}^{p} W_i^{\text{cov}} \left(\mathcal{Z}^{(i)} - \boldsymbol{y}\right)\left(\mathcal{Z}^{(i)} - E(z)\right)^T.$$

By Taylor expansion one can show second-order accuracy of mean and covariance [51].

12.9.2 Stochastic crack growth

In stochastic crack growth, the crack depth a is a function of the stochastic parameter vector $\mathbf{x}$ and load cycle N,

$$a = a(\mathbf{x}, N), \tag{12.28}$$

and the failure function (12.22) is given by

$$g(\mathbf{x}, N) = a_{\text{crit}} - a(\mathbf{x}, N), \tag{12.29}$$

where a_{crit} denotes a critical crack depth indicating failure of the component. We consider elliptical surface cracks given by crack depth a and crack form $b = a/c$. Both the initial crack depth $\alpha = a_0$ and the initial crack form b_0 are stochastic. Further stochastic parameters are given by the so-called POD curve and the probability of crack initiation. In contrast to most other places in this handbook, here "POD" does not mean "proper orthogonal decomposition," but "*probability of detection*." The POD curve gives the probability of crack detection during inspection and so characterizes the inspection scheme. The final crack depth monotonically depends on the initial crack size a_0 such that the reformulation of the failure integral of Section 12.9.1 can be applied with $x_n = a_0$ in (12.26). The goal is to compute the cumulative failure probability for a given number of equidistant inspection intervals, under consideration of:

- the replacement of a component if a crack is detected during inspection;
- the probability of crack initiation (input).

We use the following notation:

T_i	i-th inspection time
$[a_{\min}, a_{\max}]$	domain of definition of crack depth a
α	initial crack depth
β	minimum detectable crack size at time T_k (according to POD curve)
$[\beta_{\min}, \beta_{\max}]$	domain of definition of β
$\hat{x}$	vector with realizations of all stochastic variables except for α and β
$\overline{\alpha}_n$	critical initial crack size leading to failure at time T_n, $a(\hat{x}, \overline{\alpha}_n, T_n) = a_{\text{crit}}$
P_f^n	probability that the crack is not detected during inspections and reaches the critical crack depth at time T_n
P_d^n	probability that the crack does not exceed the critical depth and is detected at time T_n
P_c^n	cumulative failure probability at time T_n under consideration of failure of replaced components
c_n	probability of crack initiation in interval $[T_{n-1}, T_n]$
$\rho_{\hat{x}}, \rho_\alpha, \rho_\beta$	stochastic densities of $\hat{x}, \alpha, \beta$

The probability of detection of a crack with depth a is given by

$$I_\beta(a) = \int_{\beta_{\min}}^{a} \rho_\beta d\beta.$$

The probability of detection of a crack at time T_n is

$$I_\beta^n = I_\beta^n(\hat{x}, \alpha) = I_\beta(a(\hat{x}, \alpha, T_n)).$$

Probabilities P_d^n and P_f^n are given by

$$P_d^n = \int_\Omega I_d^n(\hat{x}) \rho_{\hat{x}} d\hat{x}, \qquad P_f^n = \int_\Omega I_f^n(\hat{x}) \rho_{\hat{x}} d\hat{x}, \tag{12.30}$$

with

$$I_d^n(\hat{x}) = \begin{cases} \int_{a_{\min}}^{\overline{\alpha}_1} I_\beta^1 \rho_\alpha d\alpha & \text{for } n = 1, \\ \int_{a_{\min}}^{\overline{\alpha}_n} (1 - I_\beta^1) \cdots (1 - I_\beta^{n-1}) I_\beta^n \rho_\alpha d\alpha & \text{for } n > 1 \end{cases}$$

and

$$I_d^n(\hat{x}) = \begin{cases} \int_{\overline{\alpha}_1}^{a_{\max}} \rho_\alpha d\alpha & \text{for } n = 1, \\ \int_{\overline{\alpha}_n}^{\overline{\alpha}_{n-1}} (1 - I_\beta^1) \cdots (1 - I_\beta^{n-1}) \rho_\alpha d\alpha & \text{for } n > 1. \end{cases}$$

The cumulative failure probability can then be computed by the following recursive scheme:

$$P_c^1 = c_1 P_f^1, \tag{12.31}$$

$$\begin{aligned} P_c^{n+1} = {} & P_f^1(c_1 + \cdots + c_{n+1}) + P_f^2(c_1 + \cdots + c_n) + \cdots + P_f^{n+1} c_1 \\ & + \left(c_1 P_d^1\right) P_c^n \\ & + \left(c_1 P_d^2 + c_2 P_d^1\right) P_c^{n-1} \\ & + \cdots + \\ & + \left(c_1 P_d^n + c_2 P_d^{n-1} + \cdots + c_n P_d^1\right) P_c^1. \end{aligned} \tag{12.32}$$

The integrals in (12.30) are the mean values of $I_d^n(\hat{x})$ and $I_f^n(\hat{x})$ and are evaluated by UKF as previously described. The critical initial crack depths $\overline{\alpha}_{n-1}$, $\overline{\alpha}_n$ appearing as integral limits in the definition of $I_d^n(\hat{x})$ and $I_f^n(\hat{x})$ are computed by a bisection algorithm. This procedure of evaluating the failure integrals has been validated by Monte Carlo for a model problem in [73].

12.9.3 Lifetime of railway axles

For lifetime investigation of railway axles, we use the failure function in (12.29) with $a_{\text{crit}} = 10$ mm. The stochastic distributions of the initial depth a_0 and initial form b_0 of the elliptical surface crack are given in Table 12.1. In this study two inspection schemes are considered:

- ultrasound far end scan;
- ultrasound mechanized.

In the first case the axle is scanned by sound from one end of the shaft to the other in the longitudinal direction, in the second case in the radial direction. The POD curves of these schemes are shown in Figure 12.17. The cumulative density function of crack initiation is given later together with the results of the results of lifetime analysis. Fracture mechanics for railway axles are subject of current research [54, 108]. Here the fracture mechanical simulations are accomplished by the simulation program ERWIN from the

Table 12.1: Stochastic parameters of crack growth.

	Type of distribution	Meaning
b_0	Uniform distribution on $[0.6, 1]$	Crack form: quotient of crack depth and length ($b_0 = a/c$)
a_0	Shifted exponential distribution with $\lambda = 100$ and shift 0.012	Initial crack depth

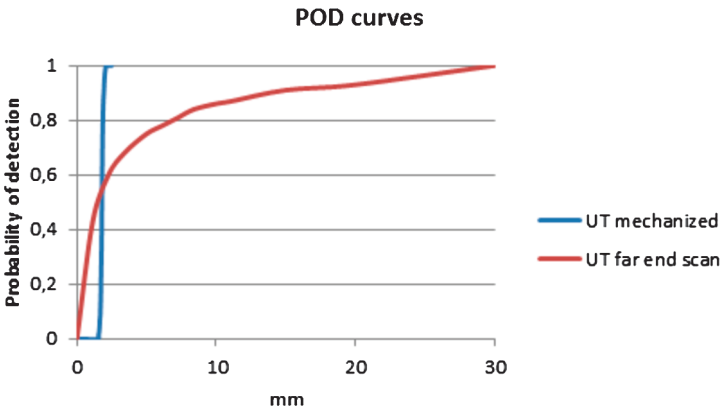


Figure 12.17: POD curves.

Fraunhofer Institute IWM in Freiburg, Germany [34]. It predicts the propagation of elliptical surface cracks, for different types of bendings. The stress intensity factors are represented by so-called polynomial influence factors [15]. Inputs of the crack simulation are internal and external stress profiles (due to the external load spectrum), da/dN-curves, and the initial crack depth and form.

It should be noted that we call the crack simulation a black box. Inputs are the initial crack depth and length, and output is the final crack depth, which is subsequently used for evaluation of (12.29).

Table 12.2: Test cases.

Test cases	Type of load	Inspection scheme	Inspection interval
Case 1	Partial load	UT far end scan	100,000 km
Case 2	Full load	UT far end scan	100,000 km
Case 3	Full load	UT far end scan	50,000 km
Case 4	Full load	UT mechanized	100,000 km

Four test cases are considered with different loads, inspection schemes, and inspection intervals (Table 12.2). The resulting lifetimes are shown in Figure 12.18. Figure 12.18 also shows the probability of crack initiation, which is input to the lifetime calculations. Lifetime is represented as a function of the deferred distance in kilometers, where one kilometer corresponds to 354 load cycles. The results show how lifetime depends on the probability of crack initiation, the type of load, and the inspection scheme. As expected, the full load case with the worst inspection scheme and inspection interval 100,000 km (case 2) has the shortest lifetime. The second worst is the partial load case (case 1) with the same inspection scheme and interval. Lifetime of the full load case can be improved by either switching to shorter inspection intervals

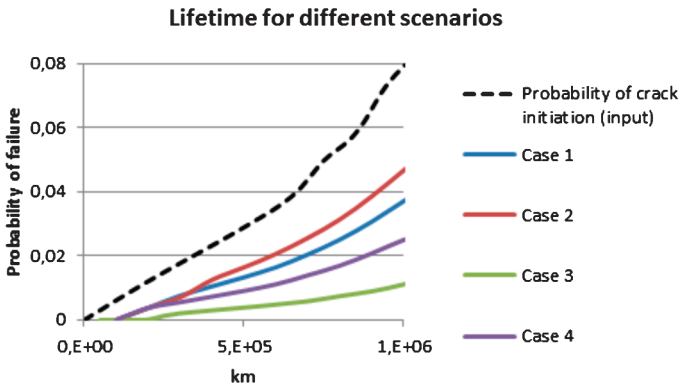


Figure 12.18: Lifetime for different scenarios, with specified probability of crack initiation (black dotted curve).

(case 3) or to a better inspection scheme (case 4). Both cases lead to longer lifetimes than cases 1 and 2, where shorter inspection intervals are more effective than a better inspection scheme. One call of the crack simulation program takes approximately 15 CPU seconds on a 3.2 GHz processor. For each curve shown, 3,000 to 4,000 simulation calls are required.

12.9.4 Conclusion

As an example of a digital twin in the design phase, lifetime analysis for railway axles has been presented, for different inspection and load case scenarios. By using adapted stochastic MOR methods, stochastic crack growth under consideration of inspections can be computed in reasonable computational time, which would not be possible with pure Monte Carlo methods. Key elements are the reformulation of the failure integrals as mean values of continuous integrands so that a nonlinear filter like the UKF can be applied with sufficient accuracy.

12.10 Use case – circuit simulation

12.10.1 MOR in the electronics industry

MOR has been part of the standard techniques used in circuit simulation for a long time, with publications dating back to at least 1990 [78]. The relation is bidirectional, with the circuit simulation and semi-conductor industry providing several benchmark cases [92, 88]. It is not only Moore's law [70], which states that circuit design complexity roughly doubles every 2 years, that drives this relationship; it is also the growing

need of circuit designers to include more physical details in their simulations and at the same time to have control over accuracy and performance. From a technological point of view, MOR methods will have to be developed that can deal with the growing complexity: The number of unknowns typically increases with the size of the design and the advance of the technology node (decrease of transistor dimensions), while the (Jacobian) density of the problem typically increases with the amount of detail included (wire resistance, capacitive coupling, inductive coupling). The MOR challenge lies hence not only in the problem's dimension, but also in the problem's complexity in terms of coupling and detail. From a business point of view, it is clear that scalability of simulation software is key (Section 12.2). What is less clear, however, is in which part of the flow the scalability should apply. For instance, before actually simulating a circuit, the differential algebraic equations describing the behavior of the circuit first need to be constructed. This process is called *extraction* and the resulting description of the circuit that can be translated into a system of differential algebraic equations is called *netlist*. Whether to apply MOR during this extraction phase and/or the simulation phase is not always clear, not only for reasons of robustness and reliability, but also for commercial reasons. In the remainder of this section we will focus mainly on the technical challenges.

12.10.2 Technological challenges

At first sight, MOR problems arising in circuit simulation may seem easy as they fall into the most elementary class of linear time-invariant dynamical systems. Electrical circuits that include nonlinear elements such as CMOS transistors are described by systems of differential algebraic equations of the form

$$\mathbf{j}(\mathbf{x}) + \frac{d\mathbf{q}(\mathbf{x})}{dt} = \mathbf{s}(t),$$

with node voltages and currents $\mathbf{x}(t) \in \mathbb{R}^n$, (non)linear vector-valued $\mathbf{q}(t, \mathbf{x})$, $\mathbf{j}(t, \mathbf{x}) \in \mathbb{R}^n$ with the electrical branch contributions, and sources $\mathbf{s}(t) \in \mathbb{R}^n$. Typically, only a linear subsystem is considered for reduction. This linear system models the behavior of linear resistors (R) and capacitors (C) and is usually considered in the frequency domain:[3]

$$G\mathbf{v} + sC\mathbf{v} = B\mathbf{u},$$

with node voltages $\mathbf{v} \in \mathbb{R}^n$, inputs $\mathbf{u} \in \mathbb{R}^k$, Laplace variable s, conductance and capacitance matrices $G, C \in \mathbb{R}^{n \times n}$, and input mapping $B \in \mathbb{R}^{n \times k}$. One distinguishes between internal nodes and terminals (or ports): Internal nodes only have connections to other

3 For simplicity we do not include inductors (L).

nodes via RLC elements, while terminals have also connections to non-RLC elements like transistors. This means that internal nodes are candidates for elimination while terminals need to be preserved, in general, because they connect the linear subsystem to the rest of the system. In the circuit simulation community, MOR is also known as netlist reduction or parasitic reduction, with the adjective parasitic referring to the nonintentional nature of the RLC elements that model the wire resistance and capacitive and inductive coupling.

Methods from several well-known categories are used for reduction:

- Krylov subspace projection, among the first of MOR methods to be applied to electrical circuits [33, 72];
- balanced truncation, with a priori error bounds [84, 9];
- modal truncation, used for the construction of behavioral models [87];
- nodal elimination methods, which have as advantages the existence of error bounds and ease of implementation [98, 96].

An advantage of nodal elimination methods (and to a lesser extent modal truncation), especially in the context of circuit simulation software, is that the ROMs can naturally be translated into a reduced circuit with meaningful RLC elements. For Krylov subspace and balanced truncation methods, the ROMs are typically dense with nonphysical (negative) RLC elements, and integration requires interfaces to deal with matrix-based circuit descriptions.

Despite the developments in the MOR domain, even the problem of reduction of linear circuits is still not considered as solved. The following key challenges can be identified for linear circuits:

- Linear solve costs: For subcircuits that contain only resistors or capacitors, projection- and elimination-based MOR procedures are error-free [89, 98], but the question of how to minimize the fill-in created by node elimination for the full design system matrix factors is still open (and becomes more difficult for mixed RLC circuits).
- Coupled problems: With decreasing feature sizes and increasing frequencies, capacitive and inductive coupling becomes stronger and denser. As a result, the original system matrices become denser, the reduction procedure becomes more expensive, and the ROM may become even denser, rendering MOR less effective.
- Precise accuracy performance tuning: For users it is important to be able to trade off between accuracy and performance. For instance, for top-level verification, one can (and often has to) accept less accuracy in order to improve simulation speed or to make simulation possible at all. The challenge is here twofold: (1) how to estimate the effect on accuracy when integrating the reduced circuit into the full design and (2) how to estimate the effect on the overall simulation time.
- Which method to apply when: There is not a single best method for MOR that fits all problems. Hence there is a need to be able to select automatically and dynamically, based on certain characteristics, which method to use.

- Variability-aware analysis: Uncertainty quantification of the impact of process variability on design robustness requires ROMs that are valid for ranges of design parameters (with as additional complication that there can be many parameters).
- With designs having a growing number of nonlinear devices like CMOS transistors, also the need for robust and efficient MOR methods for nonlinear systems increases.

12.10.3 SRAM memories: critical path simulations

SRAM memory designs [109] are of interest for MOR methods for several reasons. SRAM designs typically have a relatively large memory block of 6- or 8-transistor bitcells (depending on the memory size) and some smaller control blocks. Although designers often replace the large bitcell matrix by a much smaller model (manually), for top-level simulations an automatic, accuracy-preserving method is required. Furthermore, the extraction (modeling) of the many and long wordlines and bitlines may result in netlists with not only many resistors but also many coupling capacitors. Especially for so-called critical path simulations, where one wants to ensure that the delay for read and write operations is within specifications, reduction must be done with care to guarantee that the delay error is within picoseconds or even less. Additionally, to assess robustness of the design against process variations, one needs to run many simulations and hence simulation time needs to be minimized. In short, for memory designs, MOR has to deal with all the challenges mentioned in the previous section.

In Figure 12.19 we show the results for time-domain simulations with reduction settings varying from conservative to aggressive. The main impact on accuracy (error in delay) and performance (simulation time) is caused by how coupling capacitors are reduced. It depends on the type of verification how much error is acceptable: This can vary from tens of picoseconds to less than one picosecond. The results are produced using the circuit simulator Eldo Premier [68].

12.10.4 Concluding remarks

Driven by rapidly increasing design sizes and complexity, MOR, also known as parasitic or netlist reduction, has become a standard and indispensable option in modern circuit simulators. Although current MOR methods are suitable for robust accuracy performance control of simulation of advanced CMOS designs, more advanced CMOS nodes and verification requirements will require the development of new methods and approaches. Not only accuracy and performance remain key priority, also methods that can be used in the context of other applications, such as variability-aware design, are required. In particular parameterized MOR methods for linear and nonlinear systems will need to be further developed in order to make them suitable for use in future industrial software (Section 12.2).

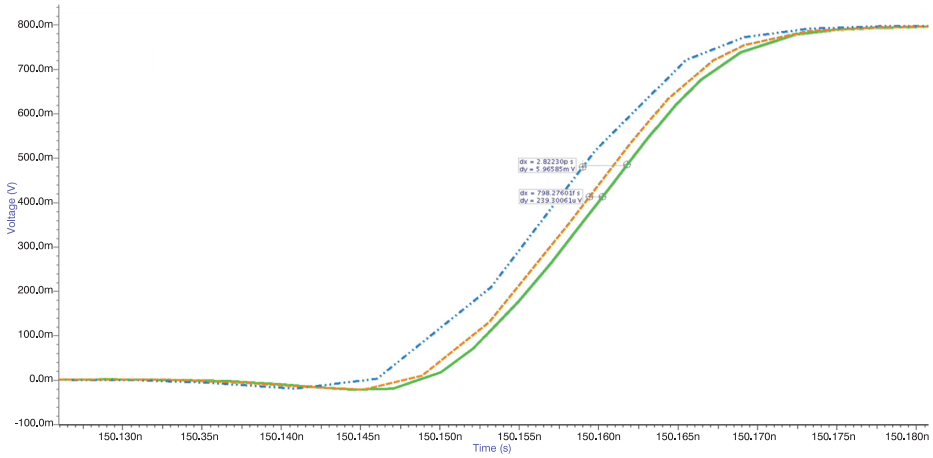


Figure 12.19: Voltage on vertical axis between −100 mV and 800 mV. Time on the horizontal axis between 150.125 ns and 150.180 ns. Time-domain simulation results of Eldo Premier [68] for conservative (orange dashed) and aggressive (blue dashed-dotted) reduction settings, compared to the golden reference (green solid). The simulation with aggressive reduction settings is two times faster than the simulation with conservative settings, but one has to pay with less accurate results (which however may still be acceptable depending on the type of verification): The delay error in the signal increases from less than 1 ps to almost 3 ps.

12.11 Conclusions and outlook

Within this contribution, we have reviewed a number of industrial success stories of MOR in the context of digital twins (Sections 12.5–12.10). Through MOR the corresponding simulations could be accelerated and reduced in their memory footprint. This enabled novel applications which would not have been possible without these improvements. Therefore, MOR is a key enabler for a new generation of digital twins (Sections 12.2 and 12.3). With respect to sustainable industrial applications and commercial software packages containing MOR engines it is crucial to close the gap from algorithms to products. Here, professional software development plays a crucial role, which we have addressed in Section 12.4.

MOR allows to reduce computational execution time of models while controlling accuracy. Application- and purpose-specific models with different requirements in terms of speed and accuracy can be realized. At the same time, MOR liberates simulation models form their execution engines, i. e., their specific simulation tools and numerical solvers. This allows a separation of the creator of a digital twin – typically a simulation engineer – and the consumer – anyone downstream including machines themselves – through appropriate application interfaces. Furthermore, this allows not only to reuse the models during operation as highlighted by some of the use cases, such as virtual sensors (Section 12.5), but also novel licenses and business models [37], such as pay-per-execution time. In particular, realizing a pay-per-execution busi-

ness model during operation allows to scale with the number of products sold by a company assuming that each product contains a digital twin. Typical business models in the context of simulation tools only scale with the number of engineers working in a company, assuming that each engineer uses corresponding tools. Therefore, the impact does not only lead to novel application areas but also to the way how industrial value streams are organized.

Summarizing the current rapid advancements in MOR, a novel generation of digital twins, so-called executable digital twins [45], is likely to emerge in the near future. An executable digital twin is a specific encapsulated realization of a digital twin with its execution engines. As such they enable the reuse of simulation models outside R&D departments. In order to do so, the executable digital twin needs to be prepared suitably for a specific application out of existing data and models. In particular, it must have the right accuracy and speed. The executable digital twin can be instantiated on edge devices, on premise servers, or in cloud environments and used autonomously by a nonexpert or a machine through a limited set of specific application programming interfaces (APIs).

In order to realize this vision, several key challenges remain open though many of them are subject to active research efforts:

- **How to prevent virtual reverse engineering?** Thanks to fast execution times, digital twins could be executed many times allowing to reverse engineer specific features, e. g., optimal control logics.
- **How to leverage MOR with black-box solvers?** Many commercial simulation tools do not provide APIs for systematic interaction with their kernels. However, this is a central requirement for integrating novel MOR tools. At the same time the development of such APIs will take significant time due to the existing development processes. That is, first, such APIs must be ranked high enough in feature backlogs for next software releases, and second, these features need to be validated and verified before they are available.
- **How to provide certifiable accuracy bounds for ROMs?** The usage of MOR to enhance operations, e. g., in the context of model predictive control, requires certifiable models, e. g., ensuring conservation of important quantities.
- **How to combine/integrate machine learning and MOR technologies better?** Machine learning technologies, e. g., neural networks, are rapidly expanding in industrial applications addressing similar aspects as MOR. However, combined concepts are still missing.
- **How to package ROMs appropriately?** Even though containerization technologies, e. g., Docker, have matured over the last years, it is not clear how to leverage them in the context of MOR, e. g., appropriate interfaces and standards are missing.

Mathematical research in MOR as well as close collaboration with industrial software providers and users will be key to address these challenges and ultimately realize the vision of executable digital twins.

Acknowledgment
The digital twin concerning the predictive maintenance was implemented by Christoph Ludwig; see again [13], [64], and [63].

Bibliography

[1] A. A. Alvarez Cabrera, K. Woestenenk, and T. Tomiyama, An architecture model to support cooperative design for mechatronic products: A control design case, *Mechatronics*, **21** (3) (2011), 534–547.
[2] R. Anderl and P. Binde, *Simulations with NX / Simcenter 3D: Kinematics, FEA, CFD, EM and Data Management*, Carl Hanser Verlag GmbH & Company KG, 2018.
[3] A. Andersson and L. Vedmar, A dynamic model to determine vibrations in involute helical gears, *J. Sound Vib.*, **260** (2) (2003), 195–212.
[4] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, vol. 6, Siam, 2005.
[5] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, A survey of model reduction methods for large-scale systems, *Contemp. Math.*, **280** (2001), 193–220.
[6] M. Bacic, On hardware-in-the-loop simulation, in *Decision and Control 2005 and 2005 European Control Conference, CDC-ECC'05, 44th IEEE Conference on*, pp. 3194–3198, IEEE, 2005.
[7] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, *Appl. Numer. Math.*, **43** (1–2) (2002), *9*–44.
[8] U. Baur, P. Benner, and L. Feng, Model order reduction for linear and nonlinear systems: a system-theoretic perspective, *Arch. Comput. Methods Eng.*, **21** (4) (2014), 331–358.
[9] P. Benner, Advances in balancing-related model reduction for circuit simulation, in J. Roos and L. R. J. Costa (eds.) *Scientific Computing in Electrical Engineering SCEE 2008*, pp. 469–482, Springer Berlin Heidelberg, 2010.
[10] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, H. Elmqvist, A. Junghanns, J. Mauß, M. Monteiro, T. Neidhold, and D. Neumerkel et al., The functional mockup interface for tool independent exchange of simulation models, in *Proceedings of the 8th International Modelica Conference; March 20th–22nd; Technical University; Dresden; Germany, number 063*, pp. 105–114, Linköping University Electronic Press, 2011.
[11] B. Blockmans, *Model Reduction of Contact Problems in Flexible Multibody Dynamics with Emphasis on Dynamic Gear Contact Problems*, PhD thesis, KULeuven, 2018.
[12] B. Blockmans, T. Tamarozzi, F. Naets, and W. Desmet, A nonlinear parametric model reduction method for efficient gear contact simulations, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1162–1191.
[13] H. Brandtstaedter, L. Huebner, A. Jungiewicz, C. Ludwig, E. Tsouchnika, and U. Wever, Digital twins for large electric power trains, in *15th Petroleum and Chemical Industry Conference Europe*, pp. 24–28, 2018.
[14] K. W. Breitung, *Asymptotic Approximations for Probability Integrals*, Lecture Notes in Mathematics, vol. 1592. Springer, 1994.

[15] M. Busch, M. Petersilge, and I. Varfolomeev, KI Faktoren und Polynomiale Einflussfunktionen für axiale und Oberflächenrisse in Zylindern, Bericht T 18/94, IWM Halle, 1994.

[16] N. Cappellini, T. Tamarozzi, B. Blockmans, J. Fiszer, F. Cosco, and W. Desmet, Semi-analytic contact technique in a non-linear parametric model order reduction method for gear simulations, *Meccanica*, **53** (1–2) (2018), 49–75.

[17] K.-H. Chang, *Design Theory and Methods Using CAD/CAE: The Computer Aided Engineering Design Series*, Academic Press, 1st edition, 2014.

[18] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[19] R. Craig and M. Bampton, Coupling of substructures for dynamic analyses, *AIAA J.*, **6** (7) (1968), 1313–1319.

[20] R. Cumbo, T. Tamarozzi, K. Janssens, and W. Desmet, Kalman-based load identification and full-field estimation analysis on industrial test case, *Mech. Syst. Signal Process.*, **117** (2019), 771–785.

[21] G. De Luca et al., ASIVA14, http://www.itn-asiva14.eu/ (2018-06-13).

[22] L. P. R. De Oliveira, M. M. Da Silva, P. Sas, H. Van Brussel, and W. Desmet, Concurrent mechatronic design approach for active control of cavity noise, *J. Sound Vib.*, **314** (3–5) (2008), 507–525.

[23] L. P. R. de Oliveira, K. Janssens, P. Gajdatsy, H. Van der Auweraer, P. S. Varoto, P. Sas, and W. Desmet, Active sound quality control of engine induced cavity noise, *Mech. Syst. Signal Process.*, **23** (2) (2009), 476–488.

[24] Digital twins – believe the hype? https://www.nafems.org/publications/benchmark/archive/april-2018/, April 2018.

[25] DMAP, https://docs.plm.automation.siemens.com/data_services/resources/nxnastran/10/help/en_US/tdocExt/pdf/dmap.pdf (2018-10-02).

[26] DMAP, https://www.plm.automation.siemens.com/global/en/products/simcenter/simcenter-cae-simulation.html (2018-10-02).

[27] F. L. M. dos Santos, R. Pastorino, B. Peeters, C. Faria, W. Desmet, L. C. Sandoval Góes, and H. Van der Auweraer, Model based system testing: bringing testing and simulation close together, in *Structural Health Monitoring, Damage Detection & Mechatronics*, vol. 7, pp. 91–97, Springer, 2016.

[28] Double vision: Using digital twins to pair virtual and physical worlds, https://youtu.be/AtYEpvnEpp0, 2018.

[29] M. Eigner, T. Dickopf, H. Apostolov, P. Schaefer, K. G. Faisst, and A. Kessler, System lifecycle management: Initial approach for a sustainable product development process based on methods of model based systems engineering, in *PLM 14*, 2014.

[30] EU-MORNET, http://www.eu-mor.net/ (2018-06-13).

[31] C. Farhat, T. Chapman, and P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1077–1110.

[32] J. Fehr and P. Eberhard, Simulation process of flexible multibody systems with non-modal model order reduction techniques, *Multibody Syst. Dyn.*, **25** (3) (2011), 313–334.

[33] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, *IEEE Trans. Comput.-Aided Des.*, **14** (1995), 639–649.

[34] I. W. M. Fraunhofer, Fracture mechanics and structural integrity, https://www.iwm.fraunhofer.de/en/services/component-safety-lightweight-construction/fracture_mechanics_structural_integrity.html.

[35] A. Fuller, Z. Fan, and C. Day, Digital twin: Enabling technology, challenges and open research, arXiv preprint arXiv:1911.01276, 2019.

[36] R. Gasch and H. Pfützner, *Rotordynamik*, Springer, 1975.

[37] O. Gassmann, K. Frankenberger, and M. Csik, *The st. Gallen business model navigator*, 2013.

[38] E. Glaessgen and D. Stargel, The digital twin paradigm for future NASA and US air force vehicles, in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, p. 1818, 2012.

[39] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.

[40] F. González, M. Á. Naya, A. Luaces, and M. González, On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics, *Multibody Syst. Dyn.*, **25** (4) (2011), 461–483.

[41] M. Grieves, Digital twin: Manufacturing excellence through virtual factory replication, *White paper*, 2014.

[42] S. Gugercin and A. C. Antoulas, A survey of model reduction by balanced truncation and some new results, *Int. J. Control*, **77** (8) (2004), 748–766.

[43] X. Guo, W. Li, and F. Iorio, Convolutional neural networks for steady flow approximation, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 481–490, ACM, 2016.

[44] B. Haas, Predictive control systems in heavy-duty commercial vehicles, in *Proc. Automotive Powertrain Control Systems*, 2012.

[45] D. Hartmann and H. Van der Auweraer, Digital twins, arXiv preprint arXiv:2001.09747, 2020.

[46] C. Heij, A. C. M. Ran, and F. van Schagen, *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*, Birkhäuser, 2006.

[47] M. Hohenbichler, S. Gollwitzer, W. Kruse, and R. Rackwitz, New light on first- and second-order reliability methods, *Struct. Saf.*, **4** (4) (1987), 267–284.

[48] M. Hohenbichler and R. Rackwitz, First-order concepts in system reliability, *Struct. Saf.*, **1** (3) (1983), 177–188.

[49] T. J. Hughes, *Linear Static and Dynamic Finite Element Analysis*, Dover Publication INC., 2000.

[50] Institute for Mathematics and its Applications, Minneapolis (MN, USA). Integrating machine learning and predictive simulation: From uncertainty quantification to digital twins, 2018.

[51] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, *IEEE Trans. Autom. Control*, **AC-45** (3) (2000), 477–482.

[52] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.*, **82** (1) (1960), 35–45.

[53] T. Kenny, Sensor fundamentals – chapter 1, in J. S. Wilson (ed.) *Sensor Technology Handbook*, pp. 1–20, Newnes, Burlington, 2005.

[54] M. Koch, A. Deisl, H.-P. Gaenser, and S. Jenne, Internationales Forschungsprojekt Eisenbahnfahrwerke 3, *ZEVrail*, **138** (2014), 93–97.

[55] D. Kondeputi and I. Prigogine, *Modern Thermodynamics*, Wiley, 1998.

[56] F. Kreith, R. M. Manglik, and M. S. Bohn, *Principles of Heat Transfer*, Cengage Learning, 7th edition, 2010.

[57] KU Leuven University of Calabria and Siemens Industry Software NV, *"demetra" (design of mechanical transmissions: Efficiency, noise and durability optimization)*, ec fp7 marie curie project nr. 324336.

[58] R. R. Lam, L. Horesh, H. Avron, and K. E. Willcox, Should you derive, or let the data drive? an optimization framework for hybrid first-principles data-driven modeling, arXiv preprint arXiv:1711.04374, 2017.
[59] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, Texts in Computational Science and Engineering, Springer, 2013.
[60] J. H. Lee, Model predictive control: Review of the three decades of development, *Int. J. Control. Autom. Syst.*, **9** (3) (2011), 415.
[61] E. Lifshitz and L. Landau, *The Classical Theory of Fields*, Course of Theoretical Physics, vol. 2, Elsevier, 4th edition, 1975.
[62] Z. Liu, W. Gao, W. Yih-Huei, and E. Muljadi, Wind power plant prediction by using neural networks, in *Energy Conversion Congress and Exposition (ECCE), IEEE 2012*, pp. 3154–3160, IEEE, 2012.
[63] C. Ludwig, O. Junge, and U. Wever, Online parameter identification methods for oscillatory systems: Estimation of changes in stiffness properties, *J. Vib. Control*, 10.1177/1077546318810265, 2018.
[64] C. Ludwig and U. Wever, Online fault identification for rotating machinery, in *ISMA2018 Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2018.
[65] P. Mas, S. B. Maddina, F. L. M. Dos Santos, C. Sobie, and H. Van der Auweraer, The application of artificial neural networks in mechatronics system development, in *ISMA2018 Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2018.
[66] MathWorks, https://mathworks.com/products/matlab.html.
[67] P. Mazur and S. R. de Groot, *Non-Equilibrium Thermodynamics*, North-Holland Publishing, 1969.
[68] Mentor Graphics, a Siemens Business, Eldo Premier 2018.1, https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/eldo-platform.
[69] S. S. Mohseni, M. J. Yazdanpanah, and N. Abolfazl Ranjbar, New strategies in model order reduction of trajectory piecewise-linear models, *Int. J. Numer. Model.*, **29** (4) (2016), 707–725.
[70] G. E. Moore, Cramming more components onto integrated circuits, *Electronics*, (April 1965), 114–117.
[71] N. X. Nastran, https://www.plm.automation.siemens.com/global/en/products/simcenter/nx-nastran.html (2018-10-02).
[72] A. Odabasioglu and M. Celik, PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **17** (8) (1998), 645–654.
[73] M. Paffrath and U. Wever, Stochastic integration methods: Comparison and application to reliability analysis, in *Proceedings of ASME Turbo Expo 2012*, Copenhagen, Denmark, June 11–15, 2012.
[74] A. Palermo, J. Anthonis, D. Mundo, and W. Desmet, A novel gear test rig with adjustable shaft compliance and misalignments part I: design, in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, pp. 497–506, Springer, 2014.
[75] A. Palermo, L. Britte, K. Janssens, D. Mundo, and W. Desmet, The measurement of gear transmission error as an NVH indicator: Theoretical discussion and industrial application via low-cost digital encoders to an all-electric vehicle gearbox, *Mech. Syst. Signal Process.*, **110** (2018), 368–389.
[76] K. Panetta, Top 10 strategic technology trends for 2018, https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/, 2017.
[77] C. Pettey, Prepare for the impact of digital twins, https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins/, 2017.
[78] L. T. Pillage and R. A. Rohrer, Asymptotic Waveform Evaluation for timing analysis, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **9** (4) (1990), 352–366.

[79] Python.org, https://www.python.org/.

[80] A. Quarteroni, G. Rozza, and A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, *J. Math. Ind.*, **1** (1) (2011), 3.

[81] A. Rasheed, O. San, and T. Kvamsdal, Digital twin: Values, challenges and enablers, arXiv preprint arXiv:1910.01719, 2019.

[82] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*, 2009.

[83] T. Reis and T. Stykel, Balanced truncation model reduction of second-order systems, *Math. Comput. Model. Dyn. Syst.*, **14** (5) (2008), 391–406.

[84] T. Reis and T. Stykel, PABTEC: Passivity-preserving balanced truncation for electrical circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **29** (9) (2010), 1354–1367.

[85] A. Rezayat, S. Shweiki, M. Park, D. Vivet, S. Donders, S. Flock, P. Jiranek, and T. Tamarozzi, A novel efficient high fidelity approach to gear contact simulation in multibody systems, in *Proceedings of the 6th European Conference on Computational Mechanics (ECCM 6) 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, 2018.

[86] D. J. Rixen, Dual Craig–Bampton method for dynamic substructuring, *J. Comput. Appl. Math.*, **168** (2004), 383–391.

[87] J. Rommes, *Methods for Eigenvalue Problems with Applications in Model Order Reduction*, PhD thesis, Utrecht University, 2007.

[88] J. Rommes, http://sites.google.com/site/rommes (2018-06-13).

[89] J. Rommes and W. H. A. Schilders, Efficient methods for large resistor networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **29** (1) (2010), 28–39.

[90] M. Rosenblatt, Remarks on a multivariate transformation, *Ann. Math. Stat.*, **23** (1952), 470–472.

[91] U. Rüde, K. Willcox, L. Curfman McInnes, H. De Sterck, G. Biros, H. Bungartz, J. Corones, E. Cramer, J. Crowley, and O. Ghattas et al., Research and education in computational science and engineering, arXiv preprint arXiv:1610.02608, 2016.

[92] J. Saak et al., https://morwiki.mpi-magdeburg.mpg.de/morwiki (2018-06-13).

[93] B. Salimbahrami and B. Lohmann, Order reduction of large scale second-order systems using Krylov subspace methods, *Linear Algebra Appl.*, **415** (2005), 385–405.

[94] R. R. Schaller, Moore's law: past, present and future, *IEEE Spectr.*, **34** (6) (1997), 52–59.

[95] W. H. A. Schilders, H. A. Van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13, Springer, 2008.

[96] E. Schrik and N. P. van der Meijs, Comparing two $y - \delta$ based methodologies for realizable model reduction, in *ProRISC IEEE 14th Annual Workshop on Circuits, Systems and Signal Processing*, pp. 148–152, 2003.

[97] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, 2013.

[98] B. N. Sheehan, Realizable reduction of *RC* networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **26** (8) (2007), 1393–1407.

[99] F. Simcenter Thermal, https://www.plm.automation.siemens.com/global/en/products/simulation-test/thermal-analysis.html (2018-10-02).

[100] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, Wiley, 2006.

[101] T. S. Söderström and P. G. Stoica, *System Identification*, Prentice Hall International Series in Systems and Control Engineering, Prentice Hall, 1989.

[102] T. Tamarozzi, G. H. K. Heirman, and W. Desmet, An on-line time dependent parametric model order reduction scheme with focus on dynamic stress recovery, *Comput. Methods Appl. Mech. Eng.*, **268** (2014), 336–358.

[103] T. Tamarozzi, P. Jiranek, A. Rezayat, and S. Shweiki, An efficient hybrid approach to gear contact simulation in multibody systems leveraging reduced order models, in *International Gear Conference*, Lyon, France, 2018.
[104] TIA, https://www.siemens.com/global/en/home/products/automation/industry-software/automation-software/tia-portal.html (2018-10-02).
[105] R. A. Toupin and C. Truesdell, Principles of classical mechanics and field theory, in S. Fluegge (ed.) *Encyclopedia of Physics*, vol. III/1, Springer, 1960.
[106] H. Van der Auweraer, J. Anthonis, S. De Bruyne, and J. Leuridan, Virtual engineering at work: the challenges for designing mechatronic products, *Eng. Comput.*, **29** (3) (2013), 389–408.
[107] H. Van der Auweraer, S. Gillijns, S. Donders, J. Croes, F. Naets, and W. Desmet, State estimation: A model-based approach to extend test data exploitation, in *Special Topics in Structural Dynamics*, vol. 6, pp. 119–128, Springer, 2016.
[108] I. Varfolomeev and M. Luke, Consideration of fatigue crack growth aspects in the design and assessment of railway axles, in G. Hutter and L. Zybell (eds.) *Recent Trends in Fracture and Damage Mechanics*, pp. 103–124, Springer, Cham, Heidelberg, New York, 2016.
[109] H. Veendrick, *Deep-Submicron CMOS ICs: From Basics to ASICs*, second, Kluwer Academic Publishers, 1999.
[110] M. Wendt, P. Li, and G. Wozny, Nonlinear chance-constrained process optimization under uncertainty, *Ind. Eng. Chem. Res.*, **41** (15) (2002), 3621–3629.
[111] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA J.*, **40** (11) (2002), 2323–2330.
[112] P. Wriggers and G. Zavarise, Computational contact mechanics, in *Encyclopedia of Computational Mechanics*, 2004.
[113] S. Y. Yoon, Z. Lin, and P. E. Allaire, *Control of Surge in Centrifugal Compressors by Active Magnetic Bearings: Theory and Implementation*, chapter 2, Springer, 2013.
[114] A. Yousefi, B. Lohmann, J. Lienemann, and J. Korvink, *Nonlinear heat transfer modelling and reduction*, 06 2004.

Bernard Haasdonk

13 MOR software

Abstract: This chapter is devoted to an important requirement of successful model order reduction (MOR) application, namely, the software aspect. The most common situation is the existence of a so-called full model, i. e., a high-fidelity, high-dimensional simulation model, that needs to be accelerated by MOR techniques, optimally without reimplementing the partially complex reduction techniques, as presented in the first volume of this handbook.

Initially, as neither full simulation models nor MOR algorithms are to be reprogrammed, but ideally are reused from existing implementations, we concentrate on the aspect of the interplay of such packages. We will discriminate, discuss, and exemplify different levels of solver "intrusiveness" that allow corresponding reduction techniques to be applied. On the one hand, most effective MOR techniques require deep access into the full model's simulation code. On the other hand, application-specific full model simulators may only offer very restricted access to internals, especially in case of commercial packages. This gap in requirements and practical accessibility motivates the discrimination into "white-box," "gray-box," and "black-box" simulation scenarios. In particular, we exemplify the ideal case of MOR for white-box situations on two examples, namely, parametric linear elliptic PDE and parametric nonlinear ODE systems. Depending on those access classes, different corresponding reduction techniques can be applied.

The second part of the current chapter then discusses existing MOR software. Several program packages exist which provide MOR techniques. They differ in availability, licensing, programming language, system types, physical application domains, external simulator bindings, etc. We give an overview of the most relevant of those MOR packages, such that applicants can identify potential suitable software library candidates.

Keywords: Model order reduction, reduced basis methods, software

MSC 2010: 65D15, 65-04, 93C15, 68N30

Acknowledgement: The author acknowledges feedback from P. Buchfink, D. Wittwar, S. Shuva, A. Schmidt, J. Rommes, N. Walker, J. Saak, C. Himpe, F. Ballarin, S. Werner, M. Cruz Varona, J. Wollner, and S. Rave which helped to improve the presentation and pointed to further software packages.

Bernard Haasdonk, Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Stuttgart, Germany, e-mail: haasdonk@mathematik.uni-stuttgart.de

https://doi.org/10.1515/9783110499001-013

13.1 Introduction

Following the main motivation of this handbook, we recall that model order reduction (MOR) is a key technique to enable high-level simulation scenarios. By reducing the dimensionality or order of a high-fidelity or "full" model, the generation of a "reduced" model is expected to give approximate but still accurate results while decreasing runtime, memory, and ideally financial costs for simulation and product development. By reduced models, more complex simulation settings are possible beyond single forward solves: "Multiquery" sampling is possible enabling uncertainty quantification or surrogate-based optimization, and "real-time" response is ideally achieved enabling interactive design or real-time control applications. We want to refer to the multitude of textbooks that have appeared during the last two decades such as [35, 4, 2, 30]. The key of efficient model reduction is an "offline-online decomposition," which relates to the separation of the MOR into two phases: The reduced model construction phase (possibly computationally intensive, performed only once) is denoted "offline phase," while the execution of the reduced model (ideally computationally cheap, performed multiple times) is very fast due to small memory and computational demands.

In this chapter we want to discuss from a rather general level some perspectives on the state and perspectives of the interplay between high-fidelity simulation software packages and MOR algorithms and code. Currently, many commercial simulation software packages exist, but few offer MOR technology or access to required internals. By this, such existing software packages cannot straightforwardly be combined with MOR and can thus not be used for modern simulation tasks, potentially representing an economic disadvantage.

On the other hand, MOR researchers have a high interest to make use of these commercial packages due to efficiency, accessibility to industrial-sized relevant applications, etc. As a result, MOR researchers frequently struggle or fail to get their algorithms to work with such commercial tools. Typically, they then either are content with "toy examples," and implement the high-dimensional solvers by themselves, or try to use the existing commercial packages by workarounds or "hacks." The ideal way, though, of developing elegant and goal-oriented interface access between solvers and MOR technology is rarely realized.

The ideal and frequent goal of MOR is to have reduced models that are computationally independent of the full dimension during the online phase. This is called "ideal online efficiency." In order to obtain this, a suitable offline-online decomposition must realize suitable interaction between the full-order and the reduced-order model. In particular for obtaining ideal online efficiency, MOR techniques are typically very code-intrusive in the sense that specific details of the full model must be accessed in an efficient manner. If a full simulator provides all of this necessary information to the outside by suitable functionality, we call such reduction scenario "white-box," as

from an MOR viewpoint the full model is fully transparent. As mentioned earlier, high-fidelity simulation packages are frequently not fully capable to provide the required ingredients, because of "closed-source" policies, code inaccessibility, code complexity, or algorithm strategy. For instance, if a full model performs discretizations in a matrix-free fashion by full mesh traverse, or a package only can assemble full finite element matrices, then a rapid partial evaluation of discretization operators or single matrix entries is difficult without (costly) traversing the mesh or assembling the global matrix, which is relevant for many MOR procedures.

In particular, beside the clean, transparent, and optimal white-box scenarios mentioned earlier, we want to introduce notions of "gray-box" and "black-box" scenarios, where the high-fidelity model provides partial or none of the required information. For such situations, there exist several practical solution strategies, workarounds, or "hacks" to enable MOR without full access to required ingredients. We will particularly also address such strategies.

As a general observation, the tedious and time-intensive development of software is not warranted suitable acknowledgement from the scientific communities in applied sciences. Regrettably, this is always just expected and accepted as a by-product of some main disciplinary scientific work/progress. This is reflected in this chapter, as we will not have many references to journal articles but mostly students theses, proceedings articles, presentations, or PhD theses that are partially devoted to the aspect of MOR software.

By nature, many reduction methods will be mentioned in this chapter, and most of these techniques are described in detail in Volumes 1 and 2 of the current handbook. In order to avoid an abundant reference list, at first use of the methods, we solely refer to the corresponding chapters.

This contribution should serve both applicants as well as MOR researchers: Applicants can identify which kind of internals they could or are willing to provide for the most elementary reduction techniques. Correspondingly, they can decide about which software packages to use and which "hacks" exist to enable MOR for their problem. Scientists working on MOR software should get an incentive to extend or contribute to existing academic MOR software packages.

This chapter is structured as follows. In the next section, we exemplify the interplay between full simulation packages and MOR algorithms. For this we choose two model examples that in certain sense represent different "corners" in the space of MOR model problems. The interplay of full and MOR software packages needs to be realized for white-box and gray-box, as well as black-box scenarios, and we comment on some main coupling techniques. In Section 13.3 we then give an overview of existing MOR software packages, both toolboxes of existing commercial simulator packages and packages developed at academic institutions. We give short characterizations of the packages such that users have some guideline on which package to choose in their respective application. We conclude in Section 13.4 with a summary and some recom-

mendations for both commercial simulator developers/companies and academic MOR researchers.

13.2 Interplay of full and reduced solvers

In order to clarify and illustrate the needs and challenges in interplay of MOR software with full solvers, we remain conceptional in this section, but still are very specific in terms of two model types. We address both the reduced basis (RB) methods community, aiming at solving parameterized partial differential equations (PDEs) (Chapters 1, 4, and 6 in Volume 2 of *Model order reduction*), and control theory researchers, aiming at ordinary differential equation (ODE) systems. Therefore, we choose two model examples representing those fields. For those models we exemplify each a plain vanilla state-of-the-art reduction scheme, which despite simplicity is intrusive in the sense that it requires deep access into the full solver. Therefore, we denote these as white-box approaches. Correspondingly, we exemplify how, in realistic cases with limited information about or restricted access to the full model, MOR can still be realized in gray-box or black-box scenarios.

As our first model we select a parametric variational form, e. g., as appearing by finite element discretizations of a linear elliptic PDE. As our second model we use a nonlinear ODE system. This covers both a linear and a nonlinear, a steady and an unsteady, and a parametric and a nonparametric case.

For these models and reduction scenarios, we necessarily must repeat some notation and terminology which appears at various places within this handbook, but which we consider essential to make the point. In the last subsection, we comment on possible coupling techniques of full and reduced solver packages. Readers who are familiar with those concepts or who are primarily interested in MOR software packages may skip this section and directly continue with Section 13.3.

13.2.1 Model 1: parametric stationary variational problem

We assume to have a Hilbert space X of functions, which is assumed to be finite-dimensional of high dimension n and spanned by basis functions $\psi_i, i = 1, \dots, n$. We want to solve a linear parametric variational problem as is typical in RB methods (Chapters 1, 4, and 6 in Volume 2 of *Model order reduction* or [14, 17]), omitting the output for ease of presentation: For a parameter $\mu \in \mathcal{P}$ from a parameter domain $\mathcal{P} \subset \mathbb{R}^p$, find a solution $u(\mu) \in X$ such that

$$a(u(\mu), v; \mu) = f(v; \mu), \quad v \in X. \tag{13.1}$$

Here, for any μ, the form $a(\cdot,\cdot;\mu) : X \times X \to \mathbb{R}$ is bilinear and $f(\cdot;\mu) : X \to \mathbb{R}$ is linear. Introducing the system matrix and right-hand side

$$\mathbf{A}(\mu) := \left(a(\psi_j, \psi_i; \mu)\right)_{i,j=1}^{n} \in \mathbb{R}^{n\times n}, \quad \mathbf{f}(\mu) := \left(f(\psi_i; \mu)\right)_{i=1}^{n} \in \mathbb{R}^{n}, \tag{13.2}$$

the solution is obtained by solving the corresponding linear system for the degree of freedom vector $\mathbf{x}(\mu) = (x_i(\mu))_{i=1}^{n} \in \mathbb{R}^n$ and subsequent linear combination

$$\mathbf{A}(\mu)\mathbf{x}(\mu) = \mathbf{f}(\mu), \quad u(\mu) = \sum_{i=1}^{n} x_i(\mu)\psi_i. \tag{13.3}$$

The problem is typically assumed to be solvable, e. g., by ellipticity or inf-sup stability assumptions. Note that by choosing $X = \mathbb{R}^n$, ψ_i being the standard basis, we obtain $u(\mu) = \mathbf{x}(\mu)$ and hence this model formulation also simply covers parametric linear equation systems of the form (13.3) without the need for a variational form (although such can easily be constructed [14]).

To enable efficient model reduction, the parametric dependency is assumed to be based on parameter separable forms ("affine assumption"), i. e., there exist Q_a coefficient functions $\theta_a^q : \mathcal{P} \to \mathbb{R}, q = 1, \ldots, Q_a$, which can be rapidly evaluated, and parameter-independent system components $\mathbf{A}^q \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{A}(\mu) = \sum_{q=1}^{Q_a} \theta_a^q(\mu)\mathbf{A}^q$$

and a similar expansion for $\mathbf{f}$ using coefficients $\theta_f^q(\mu)$ and components $\mathbf{f}^q, q = 1, \ldots, Q_f$. We denote the inner product matrix of the space X as

$$\mathbf{K} := \left(\langle \psi_i, \psi_j \rangle\right)_{i,j=1}^{n} \in \mathbb{R}^{n\times n}, \tag{13.4}$$

which allows to compute norms (or errors, projections, orthogonalizations, Riesz representers for error estimation, etc.), e. g., $\|u(\mu)\|_X = \sqrt{\mathbf{x}(\mu)^T\mathbf{K}\mathbf{x}(\mu)}$.

13.2.1.1 White-box reduction scenario

In parametric problems we discriminate between the offline phase (reduced model construction, potentially computationally expensive and involving full system components or solves) and the online phase (assembly and solve of reduced system, rapidly executable). The latter can then be evaluated in many-query contexts. As reduction technique we consider Galerkin projection. For this, in the offline phase, a matrix $\mathbf{V} \in \mathbb{R}^{n\times r}$ is constructed, where $r \ll n$ indicates the reduced dimension. In RB methods, $\mathbf{V}$ can be obtained by various techniques, e. g., as concatenation of solution snapshots

(Lagrangian basis), Gram–Schmidt orthogonalization of such, proper orthogonal decomposition (POD) (Chapter 2 in Volume 2 of *Model order reduction*), or greedy procedures. All of those have in common that several full system solves (for the snapshots) must be executed and, as mentioned earlier, the inner product matrix $\mathbf{K}$ may be required for orthogonalization.

After basis matrix generation, the full system components are projected,

$$\mathbf{A}_r^q := \mathbf{V}^T \mathbf{A}^q \mathbf{V} \in \mathbb{R}^{r\times r}, \quad \mathbf{f}_r^q := \mathbf{V}^T \mathbf{f}^q \in \mathbb{R}^r \tag{13.5}$$

for q ranging from 1 to Q_a and Q_f, respectively. This concludes the offline step.

Then in the online step, for any given parameter $\mu \in \mathcal{P}$ only the coefficient functions need to be evaluated and the reduced system is assembled by linear combination,

$$\mathbf{A}_r(\mu) := \sum_{q=1}^{Q_a} \theta_a^q(\mu)\mathbf{A}_r^q, \quad \mathbf{f}_r(\mu) := \sum_{q=1}^{Q_f} \theta_f^q(\mu)\mathbf{f}_r^q. \tag{13.6}$$

The reduced solution then results via its r-dimensional coefficient vector $\mathbf{x}_r(\mu) \in \mathbb{R}^r$ from solving

$$\mathbf{A}_r(\mu)\mathbf{x}_r(\mu) = \mathbf{f}_r(\mu).$$

If reconstruction of the approximate solution is desired (e. g., for visualization, etc.), the approximation $\tilde{\mathbf{x}}(\mu) \in \mathbb{R}^n$ of $\mathbf{x}(\mu)$ is obtained as

$$\tilde{\mathbf{x}}(\mu) = \mathbf{V}\mathbf{x}_r(\mu).$$

By realizing this reduction strategy, an ideal online efficiency is obtained, meaning that the reduced computations in the online stage do not involve any operations of high complexity n, but only of complexity depending on small quantities r, Q_a, Q_f. If a full discretization scheme provides all the required internals to realize the above reduction and reduced solution steps, we call it a white-box reduction scenario.

Figure 13.1 illustrates and summarizes this decomposition and white-box interplay of full-order and reduced-order simulation software.

Note that instead of providing the full-dimensional matrices $\mathbf{K}, \mathbf{A}^q$, it can as well be sufficient that the solver provides suitable matrix–vector multiplication routines. By this, also PDE discretization schemes that work with matrix-free implementations can be used in a white-box fashion.

13.2.1.2 Gray-box reduction scenarios

Now, in practice, the full solver package may not give access to all of the details. In particular, we are not aware of any commercial simulator package giving explicit access

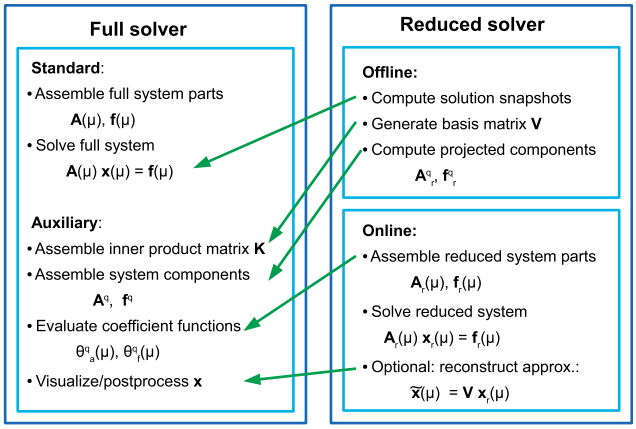


Figure 13.1: Required functionality and interplay of full (high-dimensional) and reduced solver in white-box MOR scenario for stationary variational problems.

to parametric matrix components and coefficient functions. We thus call this scenarios gray-box, as the full solver does not give complete insight into the problem, but still MOR is desired.

Missing K

A first problem may consist in the situation that the full model does not provide an inner product matrix (13.4). This prevents measuring errors, computing projections, and orthogonalizations in the correct function space norms.

If geometry information of the underlying mesh and information about the discrete function spaces (polynomial degree, order of the nodes) is available, this matrix may be constructable "by hand" outside of the simulator code. Still this is of the same technical complexity as assembling a finite element matrix, which may require considerable effort.

Alternatively, instead of working with the correct $\mathbf{K}$, one could choose certain alternative matrices. This corresponds to choosing another inner product on the solution space. A common choice is $\mathbf{K} := \mathbf{I}_n$, the identity, i. e., choosing $X = \mathbb{R}^n$ as standard Euclidean space. Note that this approach is simple but may lead to severely suboptimal bases as the meaning/weighting of the different entries of the degree of freedom vector is not respected. For example, if those degrees of freedom relate to values on grid cells of largely varying size, the small cells are given the same weight as the large cells, which may mislead basis generation (e. g., when computing a POD without proper weighted inner product) and thus deteriorate the global accuracy. This choice is common in engineering practice, but should be applied with care. Another choice for the inner product matrix could be $\mathbf{K} = \mathbf{A}(\bar{\mu})$ for a fixed reference parameter $\bar{\mu}$ if the system provides a symmetric, positive definite system matrix. This choice actually cor-

responds to the "energy inner product" in case of thermal diffusion problems, which is known to be beneficial for error estimation [14].

Missing parameter separable decomposition

Most prominently the parameter separable decomposition may not be available. A first modification of the procedure could consist of the following. One can omit to work with components $\mathbf{A}_r^q, \mathbf{f}_r^q$, and instead directly assemble the reduced system by projection of the system matrices during the online phase,

$$\mathbf{A}_r(\mu) = \mathbf{V}^T\mathbf{A}(\mu)\mathbf{V}, \quad \mathbf{f}_r(\mu) = \mathbf{V}^T\mathbf{f}(\mu). \tag{13.7}$$

We remark that the definitions of the reduced system and the parameter separable decomposition in the previous subsection exactly yield this equivalent representation in the case of parameter separable decomposition. Thus this is no approximation step but rather a reformulation of computing the reduced system. However, in (13.7) there is a computational bottleneck, as the assembly of the full system $\mathbf{A}(\mu)$ and $\mathbf{f}(\mu)$ is expensive, in particular polynomial in n. Hence, the online efficiency is sacrificed.

A second way of solving the missing parameter separability is to produce an approximate parameter separable approximation of the problem. Several approaches can be found in the literature that help to solve the issue of the missing separable decomposition in the system components. For the right-hand side vector the discrete empirical interpolation method (DEIM) [8] can be applied, which will be treated in more detail in the context of nonlinear unsteady problems in Section 13.2.2. A variant of this procedure has been formulated for matrices, which is called the matrix DEIM (MDEIM) [40, 28]. In the online phase this method only requires the evaluation of a few matrix entries $(\mathbf{A}(\mu))_{i_m,j_m}$ for a set of M "magic index pairs" $(i_m, j_m), m = 1, \ldots, M$. Still, this pointwise assembly might not be possible or highly inefficient with a given solver package, for instance, if it only can assemble the complete system matrix and not single entries. Then obviously, extraction of single matrix entries has complexity polynomial in n due to the required assembly of the complete matrix. But if this local entry assembly is possible in an effective way, then this procedure can be online-efficient. Also, the procedure can generate an exact parametric representation: If the parametric matrices lie within a finite-dimensional space, this MDEIM procedure will find such an exact representation, where M is exactly the dimension of this covering finite dimensional space.

An alternative can be parametric regression or interpolation of system matrices [41], denoted as "operator extraction": Based on a given set of system matrix snapshots, a polynomial approximation or interpolation in the parameters is realized and successfully used. In the online phase, access to neither $\mathbf{A}(\mu)$ nor its entries is required, recovering the ideal online efficiency. However, no error control for assessing the parametric approximation quality is possible by this approach.

We want to mention another approach in a PDE context, which was coined the "two-grid finite element/reduced basis approach" [7]: One constructs a RB on the given (fine) grid in a traditional way, e. g., Lagrangian, POD, or greedy basis. Then in the online phase, for a new parameter, a full problem is solved but on a coarser mesh. Finally, this coarse mesh solution is projected on the obtained (fine mesh) RB. By this, missing details on the coarse mesh are potentially included in the projected solution, as such fine details are contained in the fine mesh snapshots/RB. The computational cost in the online phase clearly is not online-efficient as it depends on the coarse mesh resolution. But it still can be computationally faster than solving the full problem on the fine mesh. In this method, internal details on the geometry, in particular the nesting of the two meshes, are required, and details on the discrete function spaces, in order to compute the prolongation operator for mapping the coarse mesh solution degree of freedom vector to a fine mesh degree of freedom vector, one further needs the inner product matrix for projection onto the RB.

So overall, in the case of the missing parameter separable decomposition gray-box scenario, either one sacrifices the ideal online efficiency while maintaining accuracy of the reduced model, or one must accept another approximation stage in the reduction chain for obtaining the optimal online runtime complexity.

These are the most common approaches of workarounds or "hacks" around missing system information for stationary problems.

13.2.1.3 Black-box reduction scenario

We denoted the previous section as gray-box, even if some references rather call their approaches black-box. The reason for our nomenclature is that the mentioned approaches still require some insight into the discretization or sampling of system components, i. e., knowledge of and access to the system structure. In contrast to this, the real notion black-box would be even more restrictive as not allowing any insight into the system components or discretization. The most limiting situation would be that the system only is observable via input–output pairs $(\mu_i, \mathbf{x}(\mu_i))$, $i = 1, \ldots, n_{\text{train}}$. Then, based on these training data, machine learning approaches could be used to infer a functional relation between the input parameters and the solution (degree of freedom vector), e. g., kernel methods (Chapter 9 in Volume 1 of *Model order reduction*), or neural networks. As a learning of a high-dimensional quantity (degree of freedom vector $\mathbf{x}(\mu)$) may suffer from the curse of dimensionality, we recommend to first identify a subspace (e. g., by POD) with suitable basis matrix $\mathbf{V}$ corresponding to an orthogonal basis; then the training data can be projected orthogonally to the reduced space $(\mu_i, \mathbf{V}^T\mathbf{K}\mathbf{x}(\mu_i))$ and the mapping from input parameters to reduced state vectors can be learned. Such an approach, however, does not involve any knowledge about the system structure. Alternative approaches would consist of assuming a certain (parametric) system structure and inferring the missing parameters purely from the observed

data. These approaches in MOR are denoted to be "data-driven." In control theory such problems already have a long tradition in the area of system identification. As this, however, is not so much related to MOR software and interplay between full and reduced solvers, we do not comment on such options.

13.2.2 Model 2: time-dependent ODE systems

As second model we assume to have a time-dependent nonlinear ODE system,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{13.8}$$

on a time interval $[0, T]$ with final time $T \in \mathbb{R} \cup \{\infty\}$, unknown state $\mathbf{x} : [0, T] \to \mathbb{R}^n$, input $\mathbf{u} : [0, T] \to \mathbb{R}^{n_u}$, and system nonlinearity $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^{n_u} \to \mathbb{R}^n$. Continuity of the right-hand side then implies existence of a solution. Additionally, there may be an output $\mathbf{y} : [0, T] \to \mathbb{R}^{n_y}$, which, for simplicity, we assume to be linear in the state

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$$

for $\mathbf{C} \in \mathbb{R}^{n_y \times n}$. We now focus on local evaluations of the system nonlinearity, as this will be the key for full online-efficient reduction. For a given set of few indices $I = \{i_1, \dots i_M\} \subset \{1, \dots n\}$ with typically $M \ll n$ (e. g., DEIM "magic indices," Chapter 5 in Volume 2 of *Model order reduction*), we define

$$\mathbf{f}_I := (f_{i_1}, \dots, f_{i_M})^T \tag{13.9}$$

as local evaluation of $\mathbf{f} = (f_i)_{i=1}^n$. We now require the following, which are typically nontrivial for given off-the-shelf simulation packages:

(i) The local evaluation $\mathbf{f}_I$ can be computed without assembling the full nonlinearity $\mathbf{f}$.[1]

(ii) The evaluation of those components can be performed rapidly based on only a subset of the state vector $\mathbf{x} = (x_i)_{i=1}^n$ in the following sense: There exists an input index subset $\bar{I} := \{\bar{i}_1, \dots \bar{i}_{\bar{M}}\}$, typically with $n \gg \bar{M} \geq M$, that defines $\mathbf{x}_{\bar{I}} := (x_{\bar{i}_1}, \dots, x_{\bar{i}_{\bar{M}}})^T \in \mathbb{R}^{\bar{M}}$ as the restriction of $\mathbf{x}$ to the indices $\bar{I}$. Then we assume that there exist functions $\bar{f}_{i_m} : \mathbb{R}^{\bar{M}} \times \mathbb{R}^{n_u} \to \mathbb{R}, m = 1, \dots M$, such that $f_{i_m}(\mathbf{x}, \mathbf{u}) = \bar{f}_{i_m}(\mathbf{x}_{\bar{I}}, \mathbf{u})$. This means that overall there is a function $\bar{\mathbf{f}}_I := (\bar{f}_{i_1}, \dots, \bar{f}_{i_M})^T : \mathbb{R}^{\bar{M}} \times \mathbb{R}^{n_u} \to \mathbb{R}^M$ satisfying

$$\bar{\mathbf{f}}_I(\mathbf{x}_{\bar{I}}, \mathbf{u}) = \mathbf{f}_I(\mathbf{x}, \mathbf{u}) \tag{13.10}$$

---

1 Note that in the DEIM literature frequently the sampling matrix $\mathbf{P} := [\mathbf{e}_{i_1}, \dots \mathbf{e}_{i_M}] \in \mathbb{R}^{n \times M}$ is defined, where $\mathbf{e}_i \in \mathbb{R}^n$ denote the standard Euclidean basis vectors. Then we verify that $\mathbf{f}_I = \mathbf{P}^T\mathbf{f}$, but the latter equation may give the wrong understanding that the sampling matrix multiplication is a computational recipe, which it is not, as this is of complexity n. Thus, we refrain from adopting this DEIM notation here and use the definition (13.9).

for any $\mathbf{x}$ and $\mathbf{u}$. We emphasize that $M, \bar{M}, n_u$ typically are small, thus this function $\bar{\mathbf{f}}_I$ can be evaluated in complexity independent of n, and hence is online-efficient. If the system corresponds to a discretized PDE these properties are useful (and realistic): Property (i) corresponds to a local evaluation of the system nonlinearity in a few grid points. Property (ii) means that such a pointwise evaluation, e. g., of a finite difference pencil, only requires the knowledge of the state in a local neighborhood around the evaluation grid points, also understandable as a submesh. This property is typical for discretizations of differential operators using basis functions of local support such as finite difference, finite element, finite volume, or discontinuous Galerkin bases. If the model has some physical meaning, there may exist a weight matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ that represents a physically relevant inner product and norm $\|\mathbf{x}\|_{\mathbf{K}} := \sqrt{\mathbf{x}^T\mathbf{K}\mathbf{x}}$.

13.2.2.1 White-box reduction scenario

Note that for linear time-invariant (LTI) systems, i. e., $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, there are plenty of reduction strategies, e. g., approximating the state trajectories for special input signals (snapshot-based such as POD, POD-greedy using the proper inner product given by $\mathbf{K}$) or approximating the input–output behavior in some sense (e. g., optimal $\mathcal{H}_2$-approximation, moment-matching, Hankel norm approximation, etc. [2]). Again for simplicity, we assume to have some basis $\mathbf{V} \in \mathbb{R}^{n\times r}$ with $r \ll n$ of orthogonal columns obtained by orthogonalizing any basis matrix of any of the mentioned procedures.

Then, a straightforward Galerkin projection of the system results in an approximation $\tilde{\mathbf{x}} := \mathbf{V}\mathbf{x}_r$ with $\mathbf{x}_r : [0, T] \to \mathbb{R}^r$ being the solution of

$$\dot{\mathbf{x}}_r(t) = \mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r(t), \mathbf{u}(t)), \quad \mathbf{x}_r(0) = \mathbf{x}_{r,0}, \tag{13.11}$$

with initial data projection $\mathbf{x}_{r,0} := \mathbf{V}^T\mathbf{x}_0$ and potential output approximation $\tilde{\mathbf{y}}(t) = \mathbf{C}\tilde{\mathbf{x}}(t) = \mathbf{C}\mathbf{V}\mathbf{x}_r(t)$. Even though this is a low-dimensional model, it is not online-efficient, as (a) the reconstruction $\mathbf{V}\mathbf{x}_r$, (b) the nonlinearity evaluation, and (c) the projection $\mathbf{V}^T\mathbf{f}$ are operations of the original complexity n. This is a well-known computational bottleneck of reduction methods such as POD. In fact, the integration of (13.11) may even be more expensive than integrating the original system (13.8), rendering the reduced model practically useless. To decrease this computational complexity, sampling-based approximations of the nonlinearity are applied, for example by gappy POD [11] or POD-DEIM [8]. Here, a further approximation stage is employed: An additional basis matrix $\mathbf{U}$ is assumed that approximates $\mathbf{f}$ well by $\tilde{\mathbf{f}} := \mathbf{U}\mathbf{c}$. The coefficient vector $\mathbf{c}$ is computed by a linear transformation of a local nonlinearity evaluation $\mathbf{c} := \mathbf{M}\mathbf{f}_I$ as introduced in (13.9). This can both be an interpolation-type

approximation, e. g., DEIM,[2] but as well a least-squares approximation by oversampling of $\mathbf{f}$. In both cases, $\mathbf{U}$ must approximate the range of $\mathbf{f}$ well, such as obtained by an additional POD of snapshots of the nonlinearity [8], or a greedy basis based on $\mathbf{f}$ snapshots (cf. the early references [16, 15]). If we replace $\mathbf{f}$ by $\tilde{\mathbf{f}}$ in (13.11) and use $\mathbf{V}^T\tilde{\mathbf{f}} = \mathbf{V}^T\mathbf{U}\mathbf{M}\mathbf{f}_I = \mathbf{W}\mathbf{f}_I$ with $\mathbf{W} := \mathbf{V}^T\mathbf{U}\mathbf{M} \in \mathbb{R}^{r\times M}$ and $(\mathbf{V}\mathbf{x}_r)_{\bar{I}} = \mathbf{V}_{\bar{I}}\mathbf{x}_r$ with $\mathbf{V}_{\bar{I}} \in \mathbb{R}^{\bar{M}\times r}$ containing the rows of $\mathbf{V}$ corresponding to the indices in $\bar{I}$, we obtain a "hyperreduced" model as alternative to (13.11):

$$\dot{\mathbf{x}}_r(t) = \mathbf{V}^T\tilde{\mathbf{f}}\big(\mathbf{V}\mathbf{x}_r(t), \mathbf{u}(t)\big) = \mathbf{W}\mathbf{f}_I\big(\mathbf{V}\mathbf{x}_r(t), \mathbf{u}(t)\big) = \mathbf{W}\bar{\mathbf{f}}_I\big(\mathbf{V}_{\bar{I}}\mathbf{x}_r(t), \mathbf{u}(t)\big), \tag{13.12}$$

assigned with the identical initial conditions and a potential output as (13.11).

We give some intuition about the operations involved in solving such a system: In the offline phase the initial state is projected, i. e., $\mathbf{x}_{r,0}$ is computed, and the small matrix $\mathbf{W}$ is assembled. Also we identify the appearance of a restriction of an RB $\mathbf{V}_{\bar{I}}$. This means, for example in the case of a finite difference discretization, that the RB needs to be stored nodewise on "all" finite difference nodes in the stencils around the sampling points I. This matrix is of small size. Now, when turning to the online phase, i. e., integration of (13.12), we clearly see the requirement of repeated evaluation of the local nonlinearity evaluation function $\bar{\mathbf{f}}_I$ that we assumed to be available by the high-fidelity model and should be rapidly evaluated in a complexity independent of n. Also, instead of reconstructing the full state approximation $\mathbf{V}\mathbf{x}_r$ as required in the model (13.11), the hyperreduced model only requires a local reconstruction of the approximated state $\mathbf{V}_{\bar{I}}\mathbf{x}_r$ which corresponds to the values of the reduced solution on a subgrid defined by the indices $\bar{I}$. This local reconstruction of the state then is sufficient to exactly evaluate the system nonlinearity in the local sampling points required for the evolution of the reduced model.

This concludes the "clean" white-box reduction scenario. We are not aware of implementation of all of the full model requirements in commercial packages to realize this. In particular the rapid local evaluation of the system nonlinearity is an extremely intrusive – and mostly also code-intrusive – requirement of making complex discretization packages suitable for online-efficient model reduction. Some publications that presented this ideal reduction for nonlinear problems include [16, 10, 8].

13.2.2.2 Gray-box reduction scenario

Now again, as white-box scenarios are rarely implemented in usual high-fidelity simulation packages, we again must refrain to gray-box scenarios.

---

2 Note that in the case of the DEIM, the matrix $\mathbf{U}$ is of size $n \times M$ and $\mathbf{M} = (\mathbf{P}^T\mathbf{U})^{-1}$. Then obviously the interpolation property is verified:

$$\mathbf{P}^T\tilde{\mathbf{f}} = \mathbf{P}^T\mathbf{U}\mathbf{M}\mathbf{f}_I = \mathbf{P}^T\mathbf{U}\big(\mathbf{P}^T\mathbf{U}\big)^{-1}\mathbf{f}_I = \mathbf{f}_I = \mathbf{P}^T\mathbf{f}.$$

.

Missing inner product weighting matrix K

If no weighting matrix $\mathbf{K}$ is available by the full model, the same workarounds as in Section 13.2.1.2 apply, i. e., working with the Euclidean inner product or reconstructing $\mathbf{K}$ by potential knowledge about the underlying PDE discretization.

Missing local evaluation routine $\bar{\mathbf{f}}_I$

In the context of nonlinear problems, the main problem is a potential missing local evaluation routine. If no access to the full model apart from full samples of $\mathbf{f}$ are possible, then one could sacrifice online efficiency for accuracy and just accept the (inefficient) nonhyperreduced model (13.11). As mentioned above, conceptually a dimension reduction will be obtained by this, but no or limited computational gain is to be expected.

If aiming at online efficiency, one can add yet another approximation stage, e. g., realized by the Kernel-DEIM approach [39, p. 83ff], [22]: The idea is to provide kernel-based approximants for the local function evaluations: $f_{i_m}(\mathbf{V}\mathbf{x}_r) \approx \hat{f}_{i_m}(\mathbf{V}_{\bar{I}}\mathbf{x}_r)$, where $\hat{f}_{i_m}$ is for instance a kernel interpolant or a more general approximant (Chapter 9 in Volume 1 of *Model order reduction*). This can be constructed purely based on state and nonlinearity snapshots.

Missing local evaluation set $\bar{I}$

If the discretization/interpretation of the full model is unknown, it may be a priori unclear which state entries influence the required sampling entries of the nonlinearity, i. e., the index set $\bar{I}$ may be unknown. A general approach for this is suggested: If the full nonlinearity of the system can be sampled, snapshot-based POD can be computed and a DEIM can be executed resulting in a possible choice of sampling points I. Now, the set of indices $\bar{I}$ that the magic index evaluations depend on must be generated. One possibility for this is analysis of the Jacobian of $\mathbf{f}$ (which clearly requires this as additional functionality from the full solver): The nonzero entries in row i_m of the Jacobian indicate those entries of the input state vector that induce changes in the value of $\mathbf{f}$. Thus, analysis of the sparsity pattern of $\nabla\mathbf{f}$ is sufficient for identification of the set $\bar{I}$.

Alternatives

Now clearly, also other gray-box approaches can be followed if one refrains from the above Galerkin projection structure. For instance by working with state and velocity snapshots of a linear system, dynamic mode decomposition (DMD) [25] allows to infer an approximate linear model. Note that velocity, i. e., nonlinearity snapshots, require a sort of intrusiveness as this is more than just state observations, and therefore not black-box. If a system can be observed in terms of input and output observations in frequency space, then the Loewner framework [20] enables to efficiently find an approximate LTI realization of a system with the observed behavior. For parametric un-

steady problems, e. g., LTI systems, the parametric matrix interpolation idea can also be applied (e. g., [13]).

13.2.2.3 Black-box reduction scenario

The mentioned gray-box approaches still assumed to have access to the systems nonlinearity (or frequency response measurements). Now, what can be done if the only way of accessing the system consists of sampling state trajectories and outputs? First, a direct mapping of input state to output quantities can be approximated with machine learning techniques. However, such techniques do not involve any model knowledge nor model assumptions. If one assumes a certain model structure, i. e., linear or polynomial nonlinearity for $\mathbf{f}$, then the system coefficients of a reduced model can be inferred from observed data [31]. In control theory this field is long known as system identification.

If one is interested in approximating $\mathbf{x}$ instead of an output, the prediction of a high-dimensional target quantity may suffer from the curse of dimensionality. So blended approaches combining model reduction and machine learning exist. For example [38] suggest a nonintrusive reduced-order model (ROM) approach for nonlinear problems combining POD and neural networks: By sampling state trajectories, a suitable approximating space/basis $\mathbf{V}$ can be computed by POD. Then each of the state snapshots can be mapped to the POD space, obtaining POD coefficients. Then a neural network can be learned to map time to the POD coefficients, which thus directly predicts an approximation of the state even without integration of a dynamical system.

13.2.3 Coupling techniques

In the above we have seen that online efficiency in MOR is based on a tight interplay of full-order solvers and MOR implementation. This requires information exchange between those levels in suitable ways. Therefore, we now discuss the main two different coupling options for MOR software. As additional reference on the coupling aspect, we want to point to a presentation that also discusses various approaches for MOR interfaces,[3] with a focus on motivating the package pyMOR as mentioned in the next section.

Write high-dimensional data to disk

The principle is the following: The full solver is agnoscent of the reduction scheme and reduction package but provides required high-dimensional data by file output, e. g., finite element method system matrices, or system matrix components (in the case of

3 https://www.stephanrave.de/talks/cse_2015.pdf

parameter separability) L^2 or H^1 scalar product weight matrices, grid description for visualization, etc. Also, the user or the full solver then must provide or export the coefficient functions of the separable parameter decompositions as function strings, etc.

The reduction scheme or MOR package then needs to import these data, project the high-dimensional objects to low-dimensional quantities, assemble the reduced system, and rapidly solve the reduced system. Potentially, the MOR code exports then high-dimensional state approximations for visualization or postprocessing by the full solver package. We refer again to Figure 13.1 for illustration of these exchange steps.

This approach by file exchange is applicable for linear problems or even problems with low polynomial structure or suitably Taylor-approximated systems.

There are several advantages or beneficial options resulting from this decomposition: One can realize a straightforward coupling of different programming platforms for the full and MOR solvers by only agreeing on a joint file format. By this, the MOR code can be written in language of choice and can be agnoscent of the programming language of the full system. Also, full solvers can be easily exchanged. One can realize a full decoupling of high-dimensional offline and online operations.

Some drawbacks and limitations of this approach are apparent: The reduced model simulation might be slow due to the disk access bottleneck. Further, the MOR code needs to process high-dimensional data. File export routines for the full solver must be implemented, which might be nontrivial (e. g., mass matrices in matrix free full solvers). This file exchange does not work for nonlinear problems or problems with complex parametric system components unless gray-box strategies, i. e., "hacks" or approximations of the preceding sections, are applied. Also, this full decomposition of offline and online phases prevents modern adaptive simulation schemes, where offline and online phases are blended (e. g., optimization with reduced model adaptation, local MOR with adaptive basis enrichment, etc.). It may be that iteration of full solver steps, reduced system creation, and solve are impossible or difficult.

Alternatively, the file exchange can also be based only on reduced quantities, if the full solver package is extended by MOR basis generation and projection functionality. Then the MOR package only needs to read reduced quantities, solve reduced systems, and communicate reduced coefficient vectors to the full solver package that is responsible for visualization/postprocessing.

Communication of full and reduced solver by function interfaces

The principle here is that the full solver is extended in order to provide function access to high-dimensional quantities used for the reduction, e. g., inner product matrices, system nonlinearities, etc. An improved approach even consists in avoiding access to matrices, but only providing access to matrix–vector multiplication for inner-product matrices or system matrices.

This offers some clear advantages: It is potentially very fast if optimal and minimally invasive connection to the full solver is realized. Nonlinear problems can be

treated by giving the MOR package access to the exact system nonlinearity. The provisioning of matrix–vector multiplication routines allows matrix-free operations, and hence MOR for discretizations that do not rely on matrix representations (e. g., finite volume or finite difference discretizations). A potential tight and repeated interaction between full and reduced solvers is possible in modern simulation scenarios with adaptive ROMs such as in optimization or local MOR. If those function interfaces are designed in a minimal and generic way, very general model reduction is possible, as various full solvers can implement those interfaces and various MOR packages can use those different full solver interface implementations. If the interface classes make use of (references to) high-dimensional objects stored in the full solver's memory, no communication of high-dimensional data is required, and the MOR code can be very fast and memory-efficient.

Again, also clear disadvantages appear: This approach is more complex as coupling via online bindings, shared libraries, mex-interfaces, or network communication, etc., is required. The realization of the required internal access may be difficult or impossible in certain full solver packages (e. g., local system matrix access to matrix-free discretization operators).

13.3 MOR software packages

After these conceptual aspects of MOR software, we want to become very specific in this section and give an overview of existing MOR software packages. First we address commercial packages, which we understand as being developed by companies with commercial interest. Then, we give a brief survey of academic packages, developed at universities or public research institutions. By the overview of this section, users may be guided (to some extent) to select the suitable packages for their application case.

13.3.1 Commercial packages with MOR functionality

Many simulator packages have realized the need and usefulness of model reduction on top of the traditional simulation chain. The following is a short and certainly incomplete list of such commercial packages. We do not go into details but refer to the corresponding websites. Mostly – but with notable exception of Scilab – the following packages are closed-source and subject to license fees.

ANSYS® (CADFem):[4] The package *Model Reduction inside ANSYS* provides reduction techniques for linear systems, in particular three-dimensional finite element from ANSYS Mechanical™, enabling piezoelectrical and thermomechanical models.

4 https://www.cadfem.de/produkte/cadfem-ansys-extensions/model-reduction-inside-ansys.html

The resulting low-dimensional matrices can be imported in ANSYS Simplorer®, MATLAB/Simulink, or other system simulation languages VHDL-AMS or SPICE. An early reference to the initial version "MOR for ANSYS" has been published [3].

Akselos Integra™(Akselos):[5] This package enables numerical simulation of large-scale mechanical assets. Among others, the underlying technique is the static condensation RB element method [19]. Being based on reduction of components, this technique is very suitable for technical component-based structures. Linear systems can be treated, as well as systems with local nonlinear subsystems.

CST MICROWAVE STUDIO® (CST):[6] In order to obtain network models, that are compatible with the circuit simulator SPICE and have the same port behavior as three-dimensional structures, suitable MOR techniques can be applied. In particular stability and passivity preservation is obtained by the reduction techniques.

MATLAB®[7] (The Mathworks, Inc.):[8] In its *control system toolbox* various control-theoretic reduction techniques are provided, in particular pole zero simplification, mode selection or balanced truncation (BT) (Chapter 2 in Volume 1 of *Model order reduction*). These methods are available in the Model Reducer App, which can be interactively used.

MOR toolbox (MOR DIGITAL SYSTEMS):[9] The MOR toolbox is a MATLAB-based toolbox gathering algorithms for (i) reduction of large-scale linear dynamical models and (ii) creation of linear dynamical models from input–output frequency data. The algorithms gathered in the MOR toolbox generate a linear state-space model whose input–output behavior is close to the initial model. Some methods take advantage of the sparse nature of the models and can therefore be applied to very large-scale models with several thousands of states. Such models arise very often in physics, biology, etc. The MOR toolbox is free for academic use, and licenses are available for commercial purposes.

SLICOT (Niconet e.V.):[10] This collection of MATLAB toolboxes also contains a *SLICOT Model and Controller Reduction Toolbox*, providing algorithms for many control-theoretic reduction methods, e. g., BT, singular perturbation approximation, frequency-weighted balancing, Hankel norm approximation, or co-prime factorization.

SciMOR (ESI Group, Scilab Enterprises SAS):[11] This model reduction toolbox has recently been developed and provides modern techniques for reduction of paramet-

5 https://akselos.com

6 https://perso.telecom-paristech.fr/begaud/intra/MWS_Getting_Started.pdf

7 We acknowledge MATLAB to be a registered trademark throughout this chapter, but for readability refrain from indicating this at various further occasions.

8 https://de.mathworks.com/help/control/ug/about-model-order-reduction.html

9 http://mordigitalsystems.fr/en/

10 http://slicot.org/matlab-toolboxes/model-reduction

11 https://scilab.io/scilab-model-reduction-toolbox

ric PDEs. In particular, POD in combination with parameter interpolation is implemented. This toolbox is part of the Scilab open-source project with the goal of democratizing computational science and is free of charge.

There are some common limitations with most of those MOR implementations: Apart from very recent developments, those commercial packages typically do not provide the most efficient latest MOR methods. This in particular holds true for packages that are already more than 10 years old, as the last decade has shown tremendous improvement in MOR technology. In these packages, in particular the full models are typically not fully accessible. As discussed in the previous sections, this prevents white-box implementation of modern reduction techniques, which mostly require a high level of intrusiveness.

13.3.2 Academic MOR software packages

Next, we want to give short descriptions of existing MOR software packages developed by groups from academia. The packages are free of charge, but may be subject to some licensing options. The packages are mostly open-source, except two of them, which are not publicly available for download. Typically, those packages are developed at universities or public research institutions.

The software packages differ in various aspects. In addition to the open-source policy, the main aspect is the programming language. Differences can also be observed with respect to the ease of installation, whether paper references are given, and whether documentation, benchmark models, or tutorial examples are provided. Some packages are under active development by large development teams, while some are single-programmer projects that resulted from PhD theses and are "frozen" or only updated in a minimal fashion. Large differences exist in the provisioning of full-scale solvers within the package and the extent of the coupling to external high-fidelity models/solvers. The purpose of those packages can be either fundamental research, teaching, or even industrial application. The physical application fields also vary widely: Some packages only support one application domain (e. g., electronics or mechanical systems) while others do not restrict the type of applications, but allow all kinds of physical domains, including electronics, fluid dynamics, biology, chemistry, finance, etc. Most discriminating are the types of systems that can be reduced such as LTI systems, nonlinear ODE systems, parametric problems, elliptic PDEs, parabolic PDEs, hyperbolic PDEs, first-order systems, second-order systems, differential algebraic equations (DAEs), and parametric, dense, or sparse systems. Finally, the implemented reduction techniques are very different in the packages, e. g., BT, moment-matching (Chapter 3 in Volume 1 of *Model order reduction*), RB methods, POD, Hankel norm approximation, optimal $\mathcal{H}_2$-norm reduction, etc.

In Table 13.1 we give some metadata for several packages from academic development teams that are available at the time of finalization of this overview, i. e., June

Table 13.1: Metadata of academic MOR software packages (as of Aug. 2019).

Acronym	Ref.	Language	License type	Latest version
DPA	[33]	MATLAB/Octave	–	Nov. 2015
emgr	[18]	MATLAB/Octave	BSD-2-Clause	5.8, May. 2020
ITHACA	[36]	C++	LGPL 3.0 / MIT License	Mar. 2020
KerMor	[39]	MATLAB	GNU GPL & BSD	0.9, Aug. 2015
M.E.S.S.	[34]	MATLAB/Octave, C	GNU GPL 2	2.0.1, Feb. 2020
MOREMBS	[12]	MATLAB, C++	–	on request
MORLAB	[5]	MATLAB	GNU Affero GPL 3	5.0, Dec. 2019
MORPACK	[26]	MATLAB	–	on request
pyMOR	[27]	Python	BSD-2-Clause	2019.2, Dec. 2019
RBmatlab	[14]	MATLAB	–	1.16.09, Sept. 2016
RBniCS	[17]	Python	GNU lesser GPL 3	v0.1.0, Jun. 2019
SparseRC	[21]	MATLAB	–	Nov. 2011
sssMOR	[6]	MATLAB	BSD-2-Clause	v2.00, Sept. 2017

2020. We do not express any preference by the order of the packages but present them in alphabetic order. For each of the packages we give a reference that either is specifically devoted to that software package or is a reference using that package. We specify the corresponding (main) programming languages, not excluding that some of the packages provide bindings or some optimized routines for some other language. Most of the packages are MATLAB-based, some using as little MATLAB-specific functionality that they are also executable from Octave. The packages devoted to MATLAB or Python do mainly not have restrictions on the operating system, as long as MATLAB or Python is installed in corresponding versions. Only the package versions providing C/C++ versions typically are limited to either Windows or Linux operating systems. About half of the packages specify some open-source GNU- or BSD-type license. The column "latest version" lists version numbers and release dates if provided by the supporting websites. Some packages do not provide access by download but only on request, in particular packages that are used for simulation of multibody systems (Chapter 2 in this volume). The recent release dates indicate that most of the packages are under active development and we recommend to consult the corresponding webpages and github sites for most recent information. Especially the packages maintained at github frequently have most recent commits that yield functional versions more recent than the latest release tags given in the table. In particular we want to emphasize that the following detailed descriptions may relate to more recent git-commits than the versions mentioned in the table. In the context of MOR software, we want also to refer to the excellent software list at the MOR Wiki.[12] Note that, by nature, our package list has a considerable overlap to that online reference list.

12 http://www.modelreduction.org

We now give some more details on the packages, including nonabbreviated name, open-source policy, online availability, license type, download (or project) URL, programming language, installation procedure, documentation, main reduction methods, application fields, and further individual comments.

DPA:[13] A collection of MATLAB algorithms related to versions of the *Dominant Pole Algorithm* [33] are provided for download as source code without license restrictions. No installation steps are required, and operation is performed via simple execution of the *.m files. Documentation is available as comments in the program files. Reduction by those methods is related to modal reduction with connections to moment-matching (Chapter 3 in Volume 1 of *Model order reduction*). The algorithms allow reduction of first- and second-order single-input, single-output systems and first-order multiple-input, multiple-output systems. The motivating field of application is power system electronics, but the code can be used for reduction and modal analysis of dynamical systems from other domains as well, including electronics (RLC parasitics), acoustics, and mechanics. Quite a number of test systems and (large-scale) system matrices are provided.

emgr:[14] The MATLAB/Octave package provides an *Empirical Gramian Framework* [18] for model reduction of nonlinear input–output systems. The program files are available for download under the open-source BSD-2-clause license. No installation steps are required, as the single MATLAB file can directly be executed without dependency on further packages. The empirical Gramians basically extend the concept of system Gramians for first-order LTI systems (Chapter 2 in Volume 1 of *Model order reduction*) to nonlinear systems. Overall, these reduction techniques can be related to BT. For parametric problems with high-dimensional parameter spaces, empirical Gramians also allow combined reduction of parameter and system order. Due to its generality, there is no restriction to the field of application: Models from neural science, mechanical systems, electrical networks, or discretized PDEs are contained as benchmark systems on the website. Extensive documentation of the programs is also provided online.

ITHACA:[15] This C++ package on *In real Time Highly Advanced Computational Applications* comes in several versions: ITHACA-FV with finite volume full-order solver https://mathlab.sissa.it/ITHACA-FV [36, 37] based on OpenFOAM, ITHACA-SEM with spectral element detailed solver https://mathlab.sissa.it/ITHACA-SEM based on Nektar++ and ITHACA-DG https://mathlab.sissa.it/ITHACA-DG based on discontinuous Galerkin full-order solver based on HopeFOAM. All versions are open-source, the former under a LGPL license, the latter two under an MIT License. The packages' code and documentation are available on corresponding

13 https://sites.google.com/site/rommes/software

14 https://gramian.de

15 http://mathlab.sissa.it/

github pages. The packages ITHACA-SEM and ITHACA-DG are at an early development stage, thus we focus on the more mature ITHACA-FV package in the following: The installation of ITHACA-FV requires an existing implementation of OpenFOAM 5.0, 6.0, or 1812, and then the cloning and compilation of the package are realized with few commands. The package provides several well-documented tutorial examples. The documentation is extracted from the code by doxygen. The package provides the implementation of several reduced-order modeling techniques for parameterized problems. In particular, the thermal block, steady and unsteady Navier–Stokes, additionally coupling with an energy equation, and the Boussinesq equation are contained. As reduction techniques, POD, nonintrusive POD-interpolation, DEIM (Chapter 5 in Volume 2 of *Model order reduction*), and DMD (Chapter 7 in Volume 2 of *Model order reduction*) are provided.

KerMor:[16] The MATLAB package provides *Kernel and MOR methods* for surrogate modeling of nonlinear systems [39]. It is open-source partly under the GNU GPL 3 and BSD license. The source files are maintained and are accessible freely at a corresponding github repository. The package is using sophisticated object-oriented features of MATLAB, and hence is not suitable to be used under Octave. Program documentation is provided online but can as well be generated offline by the `mtoc++` and `doxygen` documentation tools. After download or cloning of the package, some installation steps are required for compiling suitable mex functions and setting environment variables. Then a single startup file needs to be executed in MATLAB in order to use the package. The considered model classes are simple IO function maps, LTI and parametric nonlinear systems. As surrogate modeling techniques, mainly projection-based methods (POD-DEIM) (Chapter 5 in Volume 2 of *Model order reduction*) and kernel methods (VKOGA, SVR) (Chapter 9 in Volume 1 of *Model order reduction*) are provided. The field of application is not limited, demo examples contain electric circuit as well as system-biological (programmed cell death) models, discretized PDEs (Burgers), or biomechanics (nonlinear elasticity for muscle models).

M.E.S.S.:[17] The *Matrix Equation Sparse Solver* library [34] provides algorithms for approximate matrix equation solving such as large-scale Lyapunov or (differential) Riccati equations. Since the solution of such matrix equations represents the core of many algorithms in model reduction and control, this package is essential for reduction of large-scale problems. The package consists of a version for MATLAB/Octave and a version for C; additionally it provides Python bindings. The code is accessible via a public git repository and covered by a GNU GPL 2 license with some exceptions. The MOR functions comprise BT and the iterative rational Krylov algorithm (IRKA) (Chapter 3 in Volume 1 of *Model order reduction*)

16 https://www.morepas.org/software/kermor/index.html

17 http://www.mpi-magdeburg.mpg.de/projects/mess

for first-order LTI systems. Documentation is provided within the MATLAB source files. Installation is straightforward by simple unpacking and a startup script for initialization at each MATLAB session. Due to its generality, model examples both contain discretized transport PDEs (heat equation, advection-diffusion) and mass-spring-damper systems for structure-preserving second-order techniques. Further demonstration examples explain the extension to structured DAE systems that allow for implicit index reduction.

MOREMBS:[18] This package on *Model order reduction for elastic multibody systems* [12] consists of both a MATLAB version (MatMorembs) and a C++ version (Morembs++). The package is not available for download, but can be provided on request for academic use. The supported reduction techniques for second-order systems comprise modal techniques such as the Craig–Bampton method for component mode synthesis (CMS), as well as Krylov subspace techniques (Chapter 3 in Volume 1 of *Model order reduction*) or SVD/Gramian-based techniques (Chapter 2 in Volume 1 of *Model order reduction*). The field of application is clearly focused on elastic mechanical systems. The strength of the package lies in a multitude of coupling options, in particular importing system matrices from various commercial finite element programs (ABAQUS, ANSYS, PERMAS, Nastran) and exporting reduced system descriptions to multibody simulator programs (MATLAB Simulink, SIMPACK, Neweul-M^2, Adams, LMS).

MORLAB:[19] This *Model Order Reduction LABoratory* package [5] is aiming for spectral-projection-based model reduction of dynamical systems. It is freely available for download as a MATLAB toolbox, Octave package, or zip archive. The package is subject to the GNU Affero General Public License 3. Installation is simple by executing a startup file for adding paths or by using the automatic mechanisms for MATLAB toolboxes and Octave packages.

There is an extensive HTML documentation included in the package. The package aims at dense first-order LTI, descriptor, or second-order systems. The spectrum of methods is based on the solution of matrix equations [29], in particular modal truncation, BT with many variants (frequency-limited BT, time-limited BT, bounded-real BT, positive-real BT, balanced stochastic truncation, linear quadratic-Gaussian BT, and $\mathcal{H}_\infty$-BT), as well as a variant of the Hankel norm approximation. Due to its generality of systems, the package does not focus on special application fields.

MORPACK:[20] This *Model Order Reduction PACKage* is a MATLAB library for reducing elastic multibody systems [26, 24]. The package is not available for download but test versions for academic purposes can be provided on request. The appli-

18 http://www.itm.uni-stuttgart.de/research/morembs/MOREMBS_en.php

19 http://www.mpi-magdeburg.mpg.de/projects/morlab

20 https://tu-dresden.de/ing/maschinenwesen/ifkm/dmt/forschung/projekte/morpack

cation field is limited to second-order mechanical systems for elastic multibody dynamics. The package acts as a general interface between finite element software (ANSYS, ABAQUS, NASTRAN, LS-DYNA) and the multibody simulators (SIMPACK, ANSYS, EMBS-Matlab). Therefore, validation of the reduction and choice of the master degrees of freedom have a high priority. Over 60 correlation criteria are available to compare reduced-order models against the original model as well as measurement data. Likewise, there are many methods to select optimal master degrees of freedom. The reduction methods that are available comprise Guyan, CMS, Krylov subspace methods, (second-order) BT, and minimal model generation by mode truncation. Also multistep reduction processes can be performed. The package is steered by a graphical user interface. Several benchmark models are provided ranging from 100,000 to 1,200,000 degrees of freedom.

pyMOR:[21] The *Model Order Reduction with Python* package is an open-source library under active development [27]. It is accessible by a repository at github under a (modified) BSD-2-Clause license. Installation basically works via `pip`, and detailed installation instructions are given on the website. Extensive program documentation is provided online and within the program source code. The package aims at covering all types of reduction techniques, ranging from RB methods for parameterized PDEs up to MOR for control systems, e. g., BT or Krylov subspace methods. Also general nonlinearities can be treated by empirical interpolation (Chapters 1 and 5 in Volume 2 of *Model order reduction*). Due to its generality, no application fields are excluded. By using abstract interfaces, coupling of external high-fidelity solvers is possible and several of such dockers exist, e. g., for Dune, FEniCS, deal.II, or NGSolve. Also, some finite element and finite volume discretizations are included based on NumPy/SciPy. Two jupyter notebooks are provided for interactive exploration of corresponding models (heat equation, spring) and reduction techniques. Many demo applications are contained within the package such as PDE-based models (Burgers equation or elliptic and parabolic equations).

RBmatlab:[22] The *Reduced Basis Matlab* package is an open-source library for numerical approximation of parameterized problems. The code is publicly available for download via the Model Reduction of Parametrized Systems (MoRePaS) website without license restrictions. The master branch is maintained as a git repository, for which access can be granted on request. Documentation is provided online and within the MATLAB function headers. This documentation can be generated offline by the `mtoc++` and `doxygen` tools. Installation is simple by unzipping, setting two environment variables, and optionally extending the MATLAB startup script to get RBmatlab started automatically during the initialization of each

21 https://github.com/pymor/pymor

22 https://www.morepas.org/software/rbmatlab

MATLAB session. As system types parametric PDEs (elliptic, parabolic, hyperbolic) mostly motivated by transport problems (heat equation, Burgers equation, two-phase flow), mechanics (elasticity), or finance (variational inequalities) are supported, as well as parametric control systems, e. g., the chemical master equation. Snapshot-based reduction methods (POD, greedy, POD-greedy) are implemented, including certification by error estimators. Coupling to the scientific computing packages Dune, Alberta, and COMSOL is realized. Additionally, the package contains PDE discretization techniques (finite element method, finite volume method, discontinuous Galerkin) to be used as high-fidelity solvers for reduction procedures. Use of those reduced models in parameter optimization and feedback control are some of the implemented multiquery settings. Many demos are implemented and can be interactively accessed for getting insight into the functionality of the package. A pedagogical model of the well-known thermal block model is provided as tutorial example [14].

RBniCS:[23] This package on *Reduced Order Modelling in FEniCS* is understood to be accompanying the book [17]. Installation prerequisites are the availability of FEniCS (with PETSc, SLEPc, petsc4py, and slepc4py), numpy, and scipy. The remaining installation of RBniCS is then easily done by cloning the git repository and requesting python3 to install the package. As model classes the package comprises parametric elliptic and parabolic problems. Both linear and nonlinear problems (using empirical interpolation) are considered. Advection-diffusion as well as Stokes and Navier–Stokes problems are readily available. The code uses clear naming, and is hence sufficiently comprehensive. A documentation can be generated but is not available online. As basis generation procedures POD, greedy, and Gram–Schmidt algorithms are realized. The successive constraint method for rapid computation of stability factor lower bounds is implemented. Almost 20 tutorials are available and suitable to be used in model reduction courses. A particular feature of this package is the ease of specifying new problems, i. e., not only changing coefficient functions but also specifying and changing the differential operators of the PDE by high-level FEniCS commands.

SparseRC:[24] SparseRC [21] is a collection of MATLAB routines which performs partitioning/reordering-based model reduction for RC netlists with nodes up to hundreds of thousands, and terminals up to tens of thousands. The motivating field of application is analog circuit design, where parasitic extraction of the physical layout may result in large-scale networks of resistors (R) and capacitors (C). These networks can be modeled by dynamical systems and SparseRC can be used to reduce these systems, exploiting and preserving properties specific to such networks.

23 https://mathlab.sissa.it/rbnics

24 https://sites.google.com/site/rionutiu2/research/software

sssMOR:[25] The *Sparse State-Space and Model Order Reduction Toolbox* [6] and the extension `psssMOR`[26] for parametric problems are MATLAB libraries distributed under the BSD-2-Clause license. The code is accessible via a git repository at github. Installation is realized by unzipping the packages and executing a few installation commands as specified on the webpage. The MATLAB functions are very well documented, and hence accessible by the MATLAB help functionality. The particular motivation of those packages is extending the MATLAB-inherent state-space toolbox, which is restricted to dense matrices and thus only allows treatment of moderately sized problems. The `sss` toolbox provides this functionality using sparse matrices, and hence can treat considerably larger system orders. The `sssMOR` library then implements MOR techniques using those sparse system representations. The system types considered are LTI control systems. Basic reduction techniques implemented are modal truncation, BT, and rational Krylov subspace methods. At the same time, it provides the IRKA as well as some more recent algorithms such as the CUmulative REduction framework (CURE), the Stability-Preserving, Adaptive Rational Krylov algorithm (SPARK), and the confined IRKA (CIRKA). Some model samples from well-known benchmark collections (CD player, building, gyro) are provided. In principle, the application scope is not limited as long as the systems can be cast as (parametric) LTI control systems. The toolboxes are currently being extended to cope with nonlinear sparse state-space systems, such as bilinear (bsssMOR) and quadratic-bilinear (qbsssMOR) models. Extensions for other system classes, e. g., port-Hamiltonian (spHMOR) and second-order (ssoMOR) systems, are naturally also conceivable under the same guiding principle.

We do not claim completeness of this above package list, as many researchers have their private code collection, libraries, or repositories. But the list covers the main currently available software packages that we are aware of. Several smaller packages exist, such as `pydmd`[27] on DMD and `ezyrb`[28] on POD with interpolation. Some academic PDE discretization packages also include MOR functionality, e. g., `libmesh`,[29] which has RB capabilities via `rbOOmit` [23], or `Feel++`,[30] which also provides MOR methods. Further packages exist which however are no longer under active development and not provided by download. Among those we want to mention `dune-rb`, whose capabilities and principles of coupling with RBmatlab have been explained in [9]. The thesis [1] describes the extension to localized model reduction approaches. Also the package

25 https://www.rt.mw.tum.de/?sssmor
26 https://www.rt.mw.tum.de/?psssmor
27 https://mathlab.sissa.it/pydmd
28 http://mathlab.sissa.it/ezyrb
29 http://libmesh.github.io/
30 http://www.feelpp.org

rbMIT is a software package that provides the most elementary RB algorithms. This software package was awarded with the Springer Computational Science and Engineering Prize in 2009. The package is currently not available by a website, but can be accessed via the internet archive.[31] Similarly the package PABTEC on BT for electronics applications is listed at the swMATH portal[32] but seems no longer to be publicly accessible. The academic MORE package[33] [32] is a precursor of the commercial MOR toolbox mentioned in the previous subsection.

Disadvantages of academic packages (which more or less applies to the different packages mentioned above) must certainly also be stated. The level of documentation of those code packages may be incomplete, development of packages may be discontinued, and support can typically not be offered in an extensive manner or instantaneously. So using these packages typically requires some self-study, reproducing of running models and reduction techniques, or even some reverse engineering, or trial and error with changing parameters.

13.4 Conclusions and recommendations

We argued that online-efficient and accurate MOR algorithms require access to data, functionality, or other internals of the full solver. In particular, in many cases, a special design, decomposition or structure of the full-order model needs to be established in order to optimally apply corresponding MOR techniques. Elementary routines from the full solver or high-fidelity simulation package that are required for white-box MOR may comprise the following: For snapshot-based MOR algorithms, e. g., POD, greedy, and POD-greedy, a triggering of a full-order simulation with specified input parameters and return of state snapshots must be available. Information about nodal interpretation of the state vector is required. In the simplest case of linear finite element or finite difference discretization this is equivalent to enumeration of the mesh nodes. In general, a routine for assembly or matrix–vector multiplication with the inner product (mass) matrix is very helpful. This enables computing L^2-norms, orthonormalization, and projections. Similarly, a subroutine for assembly or matrix–vector multiplication with the stiffness matrix, which enables computing Sobolev H^1-semi-norms, is helpful. Export of other system matrices, e. g., Jacobian matrices of nonlinear terms, may be required. For parametric problems an export of parameter separable decompositions is essential; this means access to (nonparametric) system matrix and vector components and parametric coefficient functions. For sampling-based methods for

31 https://web.archive.org/web/20171110212818/http://augustine.mit.edu:80/methodology/methodology_rbMIT_System.htm

32 https://swmath.org/software/4061

33 https://w3.onera.fr/more

nonlinear problems, local evaluation of system nonlinearities based on local reconstruction of the state vector needs to be provided. Optionally, visualization or postprocessing of a given state vector by the full solver is useful. If those functionalities are not provided by the full solver, either a loss of online efficiency is unavoidable, or some gray-box or black-box workarounds or hacks might be possible as we have exemplified for general stationary and unsteady, linear, and nonlinear problems.

A multitude of software packages from groups from academia have been developed. Typically those packages not only provide the latest MOR technology, but also implementations of some full system solvers. Only by control of all implementational details of the full models, which are typically code-intrusive, optimal online efficiency can be realized. This is mostly not possible by full-order models from commercial packages. This full control of the high-fidelity models enables independence of scientists from external simulator packages, but at the same time is very time-intensive, distracting one from main disciplinary research.

We want to close with some recommendations. Addressing commercial software developers, we think that simulation packages will not remain competitive if MOR technology is not included as essential enabler for modern higher-level simulation tasks such as uncertainty quantification, parametric studies, optimization, design, etc. Simulation software engineers should be aware that their solver is no longer the last element in the simulation analysis pipeline but those are embedded in more complex simulation tasks. Inclusion of MOR algorithms in simulator packages can thus be a competitive advantage. When planning to include MOR technology, one should be clearly aware that access to more internals than just the solution/state degree of freedom vector is required. Access to system matrices, components, local nonlinearity evaluations, local subgrid geometry, etc., can be useful or required for modern and efficient MOR algorithms as listed above. Creating suitable interfaces or other means of access to those internals will enable applying efficient MOR algorithms. Apart from exporting system parameters, matrices, or geometry information, also importing facilities for effective bases, error estimation routines, etc., would be recommended for expanding the scope of applicability of commercial simulation packages. In particular, by realizing error estimation techniques, a certification and therefore guarantee of reliability of methods/packages is obtained. The last decade has enabled hundreds of PhD students to specialize and graduate in the field of MOR. These would be excellent candidates for transfer of those technologies into business and industry. A further option for realizing MOR algorithms is the foundation of public–private partnerships or joint programs, e. g., European Industrial Training Networks, etc.

Some recommendations for academic MOR researchers might be to develop further "hacks" of industrial software, i. e., using solvers as black-box or gray-box leading to new algorithms and analysis. For reproducibility of results, we strongly encourage to provide open-source and online accessible program code. If possible, it is advisable to contribute to existing MOR libraries instead of reinventing the wheel by developing new packages from scratch. If it is required or desired to reinvent MOR code or solvers,

practitioners or simulation engineers should think of useful central interfaces, which will ease exchange and coupling of the code to other packages. This is particularly relevant for bridging different programming platforms such as C++, MATLAB, Python, Fortran, etc.

A general wish for the development in computational sciences is the increased acknowledgment of and respect for the effort in development, maintenance, and documentation of software packages. Typically, applications and computational approaches are so involved, that software packages cannot be set up from scratch by scientists within the typical scientific "lifetime," e. g., a PhD thesis. In particular, existing open-source packages directly enable a high entrance level for further developments. Software development thus is crucial and one core scientific service for the community. Scientists and students who develop, maintain, and provide well-designed software should more easily obtain scientific credits by accepting such results as major scientific results in theses as well as being able to publish journal articles on such software – in contrast to the current state of expecting such developments as minor by-product of disciplinary scientific contributions. But also in funding agencies such major achievements or proposals on scientific software development should be accepted as foundation for disciplinary progress. Only slowly this awareness in the scientific communities is developing, e. g., reflected by the recent Software branch at the SIAM journal on Scientific Computing that now also enables to publish journal articles on software. But certainly this general awareness can be further strengthened by each individual researcher.

Bibliography

[1] S. Alebrand, *Efficient Schemes for Parametrized Multiscale Problems*. PhD thesis, University of Stuttgart, 2015.

[2] A. C. Antoulas, *Approximation of Large–Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005.

[3] T. Bechtold, E. B. Rudnyi, and J. Korvink, Selected model reduction software, in *Fast Simulation of Electro-Thermal MEMS: Efficient Dynamic Compact Models*, Springer, 2007.

[4] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox (eds.), *Model Reduction and Approximation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[5] P. Benner and S. W. R. Werner, MORLAB – Model Order Reduction LABoratory (version 4.0), December 2018. See also: http://www.mpi-magdeburg.mpg.de/projects/morlab.

[6] A. Castagnotto, M. Cruz Varona, L. Jeschek, and B. Lohmann, sss & sssMOR: analysis and reduction of large-scale dynamical systems in MATLAB, *Automatisierungstechnik*, **65** (2) (2017), 134–150.

[7] R. Chakir and Y. Maday, A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE, in *9e Colloque National en Calcul des Structures*, 2009.

[8] S. Chaturantabut and D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.

[9] M. Drohmann, B. Haasdonk, and M. Ohlberger, A software framework for reduced basis methods using DUNE-RB and RBMATLAB, in A. Dedner, B. Flemisch, and R. Klöfkorn (eds.), *Advances in DUNE: Proceedings of the DUNE User Meeting, Held in October 6th–8th 2010 in Stuttgart*, Springer, Germany, 2012.

[10] M. Drohmann, B. Haasdonk, and M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM J. Sci. Comput.*, **34** (2) (2012), A937–A969.

[11] R. Everson and L. Sirovich, Karhunen–Loeve procedure for gappy data, *J. Opt. Soc. Am. A*, **12** (1995), 1657–1664.

[12] J. Fehr, D. Grunert, P. Holzwarth, B. Fröhlich, N. Walker, and P. Eberhard, MOREMBS – a model order reduction package for elastic multibody systems and beyond, in *Reduced-Order Modeling (ROM) for Simulation and Optimization*, pp. 141–166, Springer, 2018.

[13] M. Geuss, *A Black-Box Method for Parametric Model Order Reduction based on Matrix Interpolation with Application to Simulation and Control*. PhD thesis, Technische Universität München, 2015.

[14] B. Haasdonk, Reduced basis methods for parametrized PDEs – a tutorial introduction for stationary and instationary problems, in P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), *Model Reduction and Approximation: Theory and Algorithms*, pp. 65–136, SIAM, Philadelphia, 2017.

[15] B. Haasdonk and M. Ohlberger, Reduced basis method for explicit finite volume approximations of nonlinear conservation laws, in *Hyperbolic Problems: Theory, Numerics and Applications*. Proc. Sympos. Appl. Math., vol. 67, pp. 605–614, Amer. Math. Soc., Providence, RI, 2009.

[16] B. Haasdonk, M. Ohlberger, and G. Rozza, A reduced basis method for evolution schemes with parameter-dependent explicit operators, *Electron. Trans. Numer. Anal.*, **32** (2008), 145–161.

[17] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Mathematics, Springer International Publishing, 2015.

[18] C. Himpe, emgr – The empirical gramian framework, *Algorithms*, **11** (7) (2018), 91.

[19] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, A static condensation reduced basis element method: complex problems, *Comput. Methods Appl. Mech. Eng.*, **259** (2013), 197–216.

[20] A. Ionita and A. Antoulas, Data-driven parametrized model reduction in the Loewner framework, *SIAM J. Sci. Comput.*, **36** (3) (2014), A984–A1007.

[21] R. Ionutiu, J. Rommes, and W. H. A. Schilders, SparseRC: sparsity preserving model reduction for RC circuits with many terminals, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **30** (12) (2011), 1828–1841.

[22] L. Kazaz, Black box model order reduction of nonlinear systems with kernel and discrete empirical interpolation. Bachelor's thesis, University of Stuttgart, 2014.

[23] D. J. Knezevic and J. W. Peterson, A high-performance parallel implementation of the certified reduced basis method, *Comput. Methods Appl. Mech. Eng.*, **200** (13–16) (2011), 1455–1466.

[24] P. Koutsovasilis and M. Beitelschmidt, MORPACK toolbox for coupling rigid and elastic multi-body dynamics, in *Proc. of NAFEMS World Congress*, 2009.

[25] N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM, 2016.

[26] C. Lein and M. Beitelschmidt, MORPACK-Schnittstelle zum Import von FE-Strukturen nach SIMPACK, *Automatisierungstechnik*, **60** (9) (2012), 547–559.

[27] R. Milk, S. Rave, and F. Schindler, pyMOR – generic algorithms and interfaces for model order reduction, *SIAM J. Sci. Comput.*, **38** (5) (2016), S194–S216.

[28] F. Negri, A. Manzoni, and D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, *J. Comput. Phys.*, **303** (2015), 431–454.
[29] S. W. R. Werner and P. Benner, Model reduction of descriptor systems with the MORLAB toolbox, in *Proc. 9th Vienna International Conference on Mathematical Modelling MATHMOD 2018*, IFAC-PapersOnLine, vol. 51, pp. 547–552, 2018.
[30] A. T. Patera and G. Rozza, *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*, to appear in (tentative) MIT Pappalardo Graduate Monographs in Mechanical Engineering, MIT, 2007.
[31] B. Peherstorfer and K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, *Comput. Methods Appl. Mech. Eng.*, **306** (2016), 196–215.
[32] C. Poussot-Vassal and P. Vuillemin, Introduction to MORE: a MOdel REduction toolbox, in *Proc. IEEE International Conference on Control Applications*, pp. 776–781, 2012.
[33] J. Rommes and N. Martins, Efficient computation of transfer function dominant poles using subspace acceleration, *IEEE Trans. Power Syst.*, **21** (3) (2006), 1218–1226.
[34] J. Saak, M. Köhler, and P. Benner, M-M.E.S.S.-1.0.1 – the matrix equations sparse solvers library. DOI:10.5281/zenodo.50575, April 2016. See also: www.mpi-magdeburg.mpg.de/projects/mess.
[35] W. H. A. Schilders, H. A. Van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13, Springer, 2008.
[36] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza, POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder, *Commun. Appl. Ind. Math.*, **8** (1) (2017), 210–236.
[37] G. Stabile and G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations, *Comput. Fluids*, (2018).
[38] Q. Wang, J. S. Hesthaven, and D. Ray, Non-intrusive reduced order modelling of unsteady flows using artificial neural networks with application to a combustion problem, *J. Comput. Phys.*, **384** (2019), 289–307.
[39] D. Wirtz, *Model Reduction for Nonlinear Systems: Kernel Methods and Error Estimation*. PhD Thesis, University of Stuttgart, October 2013.
[40] D. Wirtz, D. C. Sorensen, and B. Haasdonk, A posteriori error estimation for DEIM reduced nonlinear dynamical systems, *SIAM J. Sci. Comput.*, **36** (2) (2014), A311–A338.
[41] O. Zeeb, *A numerical framework for semi-automated Reduced Basis Methods with blackbox solvers*. PhD thesis, University of Ulm, 2015.

Index