Contents

Preface v

1 Introduction to computer simulation 1

1.1 Physics and computational physics 1

1.2 Choice of programming language 5

1.3 Outfitting your PC for scientific computing 13

1.4 History of computing in a nutshell 17

1.5 Number representation: bits and bytes in computer memory 22

1.5.1 Addition and subtraction of dual integer numbers 24

1.5.2 Basic data types 29

1.6 The role of algorithms in scientific computing 41

1.6.1 Efficient and inefficient calculations 43

1.6.2 Asymptotic analysis of algorithms 51

1.6.3 Merge sort and divide-and-conquer 56

1.7 Theory, modeling and computer simulation 59

1.7.1 What is a theory? 59

1.7.2 What is a model? 67

1.7.3 Model systems: particles or fields? 72

1.7.4 The linear chain as a model system 74

1.7.5 From modeling to computer simulation 78

1.8 Exercises 81

1.8.1 Addition of bit patterns of 1 byte duals 81

1.8.2 Subtracting dual numbers using two's complement 81

1.8.3 Comparison of running times 81

1.8.4 Asymptotic notation 82

1.9 Chapter literature 83

2 Scientific Computing in C 84

2.1 Introduction 84

2.1.1 Basics of a UNIX/Linux programming environment 87

2.2 First steps in C 99

2.2.1 Variables in C 101

2.2.2 Global variables 103

2.2.3 Operators in C 104
2.2.4 Control structures 108
2.2.5 Scientific "Hello world!" 111
2.2.6 Streams – input/output functionality 116
2.2.7 The preprocessor and symbolic constants 119
2.2.8 The function *scanf()* 122

2.3 Programming examples of rounding errors and loss of precision 125
2.3.1 Algorithms for calculating e^{-x} 130
2.3.2 Algorithm for summing $1/n$ 133

2.4 Details on C-Arrays 137
2.4.1 Direct initialization of certain array elements (C99) 141
2.4.2 Arrays with variable length (C99) 141
2.4.3 Arrays as function parameters 142
2.4.4 Pointers 144
2.4.5 Pointers as function parameters 152
2.4.6 Pointers to functions as function parameters 154
2.4.7 Strings 159

2.5 Structures and their representation in computer memory 161
2.5.1 Blending structs and arrays 163

2.6 Numerical differentiation and integration 165
2.6.1 Numerical differentiation 166
2.6.2 Case study: the second derivative of e^x 169
2.6.3 Numerical integration 176

2.7 Remarks on programming and software engineering 181
2.7.1 Good software development practices 181
2.7.2 Reduction of complexity 184
2.7.3 Designing a program 188
2.7.4 Readability of a program 189
2.7.5 Focus your attention by using conventions 190

2.8 Ways to improve your programs 191

2.9 Exercises 193
2.9.1 Questions 193
2.9.2 Errors in programs 194
2.9.3 *printf()*-statement 197
2.9.4 Assignments 198
2.9.5 Loops 199
2.9.6 Recurrence 199
2.9.7 Macros 200
2.9.8 Strings 200
2.9.9 Structs 201

2.10 Projects . . . 203
2.10.1 Decimal and binary representation . . . 203
2.10.2 Nearest machine number . . . 203
2.10.3 Calculating e^{-x} . . . 203
2.10.4 Loss of precision . . . 204
2.10.5 Summing series . . . 204
2.10.6 Recurrence in orthogonal functions . . . 205
2.10.7 The Towers of Hanoi . . . 205
2.10.8 Spherical harmonics and Legendre polynomials . . . 207
2.10.9 Memory diagram of a battle . . . 208
2.10.10 Computing derivatives numerically . . . 208
2.11 Chapter literature . . . 210

3 Fundamentals of statistical physics 211

3.1 Introduction and basic ideas . . . 212
3.1.1 The macrostate . . . 216
3.1.2 The microstate . . . 218
3.1.3 Information conservation in statistical physics . . . 219
3.1.4 Equations of motion in classical mechanics . . . 225
3.1.5 Statistical physics in phase space . . . 229
3.2 Elementary statistics . . . 235
3.2.1 Random Walk . . . 236
3.2.2 Discrete and continuous probability distributions . . . 241
3.2.3 Reduced probability distributions . . . 242
3.2.4 Important distributions in physics and engineering . . . 244
3.3 Equilibrium distribution . . . 249
3.3.1 The most probable distribution . . . 251
3.3.2 A statistical definition of temperature . . . 253
3.3.3 The Boltzmann distribution and the partition function . . . 255
3.4 The canonical ensemble . . . 258
3.5 Exercises . . . 261
3.5.1 Trajectories of the one-dimensional harmonic oscillator in phase space . . . 261
3.5.2 Important integrals of statistical physics . . . 261
3.5.3 Probability, example from playing cards . . . 261
3.5.4 Rolling dice . . . 262
3.5.5 Problems, using the Poisson density . . . 262
3.5.6 Particle inside a sphere . . . 262

4 Inter- and intramolecular potentials 264
4.1 Introduction 265
4.2 The quantum mechanical origin of particle interactions 266
4.3 The energy hypersurface and classical approximations 270
4.4 Non-bonded interactions 271
4.5 Pair potentials 274
4.5.1 Repulsive Interactions 275
4.5.2 Electric multipoles and multipole expansion 280
4.5.3 Charge-dipole interaction 280
4.5.4 Dipole-dipole interaction 283
4.5.5 Dipole-dipole interaction and temperature 284
4.5.6 Induction energy 285
4.5.7 Dispersion energy 287
4.5.8 Further remarks on pair potentials 288
4.6 Bonded interactions 291
4.7 Chapter literature 292
5 Molecular Dynamics simulations 294
5.1 Introduction 295
5.1.1 Historical notes on MD 299
5.1.2 Limitations of MD 303
5.2 Numerical integration of differential equations 309
5.2.1 Ordinary differential equations 309
5.2.2 Finite Difference methods 310
5.2.3 Improvements to Euler's algorithm 316
5.2.4 Predictor-corrector methods 317
5.2.5 Runge–Kutta methods 318
5.3 Integrating Newton's equation of motion: the Verlet algorithm 320
5.4 The basic MD algorithm 323
5.5 Basic MD: planetary motion 327
5.5.1 Preprocessor statements and basic definitions 327
5.5.2 Organization of the data 328
5.5.3 Function that computes the energy 328
5.5.4 The Verlet velocity algorithm 329
5.5.5 The force calculation 329
5.5.6 The initialization and output functions 332
5.5.7 The *main()*-function 332
5.6 Planetary motion: suggested project 335

5.7 Periodic boundary conditions 337

5.8 Minimum image convention 338

5.9 Lyapunov instability 338

5.10 Case study: static and dynamic properties of a microcanonical LJ fluid 341
5.10.1 Microcanonical LJ fluid: suggested projects 346

5.11 Chapter literature 356

6 Monte Carlo simulations 357

6.1 Introduction to MC simulation 357
6.1.1 Historical remarks 360

6.2 Simple random numbers 362
6.2.1 The linear congruential method 365
6.2.2 Monte Carlo integration – simple sampling 369

6.3 Case study: MC simulation of harddisks 378
6.3.1 Trial moves 378
6.3.2 Case study: MC simulation of harddisks – suggested exercises 380

6.4 The Metropolis Monte Carlo method 381

6.5 The Ising model 383
6.5.1 Case study: Monte Carlo simulation of the 2D Ising magnet . 386

6.6 Case Study: NVT MC of dumbbell molecules in 2D 392

6.7 Exercises 394
6.7.1 The GSL library 394
6.7.2 Calculating π 395
6.7.3 Simple and importance sampling with random walks in 1D . . 399
6.7.4 Simple sampling and importance sampling
with random walks in 1D 399

6.8 Chapter literature 407

7 Advanced topics, and applications in soft matter 408

7.1 Partial differential equations 408
7.1.1 Elliptic PDEs 410
7.1.2 Parabolic PDEs 411
7.1.3 Hyperbolic PDEs 411

7.2 The finite element method (FEM) 412

7.3 Coarse-grained MD for mesoscopic polymer and biomolecular
simulations 414
7.3.1 Why coarse-grained simulations? 414

7.4 Scaling properties of polymers 415

7.5 Ideal polymer chains 416
7.6 Single-chain conformations 420
7.7 The ideal (Gaussian) chain model 421
7.8 Scaling of flexible and semiflexible polymer chains 422
7.9 Constant temperature MD 429
7.10 Velocity scaling using the Behrendsen thermostat 430
7.11 Dissipative particle dynamics thermostat 431
7.12 Case study: NVT Metropolis MC simulation of a LJ fluid 434
7.13 Exercise 443
7.13.1 Dumbbell molecules in 3D 443

A The software development life cycle 444

B Installation guide to Cygwin 445

C Introduction to the UNIX/Linux programming environment 448

C.1 Directory structure 448
C.2 Users, rights and privileges 450
C.3 Some basic commands 453
C.4 Processes 455
C.4.1 Ending processes 455
C.4.2 Processes priorities and resources 455
C.5 The Bash 457
C.6 Tips and tricks 458
C.7 Useful programs 458
C.7.1 Remote connection: ssh 459
C.7.2 Gnuplot 459
C.7.3 Text editors: vi, EMACS and others 459

D Sample program listings 470

D.1 Sample code for file handling 470

E Reserved keywords in C 473

F Functions of the standard library <*string.h*> 474

G Elementary combinatorial problems 475

G.1 How many differently ordered sequences of N objects are possible? . 475
G.2 In how many ways can N objects be divided into two piles, with n and m objects, respectively? 475

G.3 In how many ways can N objects be arranged in $r + 1$ piles with n_j objects in pile number j with $j \in [0, 1, \ldots, r]$? 476

G.4 Stirling's approximation of large numbers 476

H Some useful constants 477

I Installing the GNU Scientific Library, GSL 478

J Standard header files of the *ANSI-C* library 479

K The central limit theorem 480

Bibliography 481

Acronyms 505

Index 506

Authors 509

The source code of more than two dozen of the program listings is available on the Walter de Gruyter webpage:

`http://www.degruyter.com/staticfiles/pdfs/9783110255904AccompanyingPrograms.zip`