

Contents

Preface — VII

Acknowledgments — IX

1 Introduction — 1

- 1.1 Why study numerical methods? — 1
- 1.2 Terminology — 2
- 1.3 Convergence terminology — 3
- 1.4 Exercises — 5

2 Computer representation of numbers and roundoff error — 7

- 2.1 Examples of the effects of roundoff error — 7
- 2.2 Binary numbers — 10
- 2.3 64-bit floating-point numbers — 12
- 2.4 Floating-point arithmetic — 14
 - 2.4.1 Avoid adding large and small numbers — 14
 - 2.4.2 Subtracting two nearly equal numbers is bad — 14
- 2.5 Visualizing a floating-point number system — 16
- 2.6 Exercises — 17

3 Solving linear systems of equations — 20

- 3.1 Linear systems of equations and solvability — 20
- 3.2 Solving triangular systems — 22
- 3.3 Gaussian elimination — 24
- 3.4 The backslash operator — 28
- 3.5 LU decomposition — 28
- 3.6 Exercises — 30

4 Finite difference methods — 33

- 4.1 Approximating the first derivative — 34
 - 4.1.1 Forward and backward differences — 34
 - 4.1.2 Centered difference — 37
 - 4.1.3 Three-point difference formulas — 40
 - 4.1.4 Further notes — 41
- 4.2 Approximating the second derivative — 41
- 4.3 Application: initial value ODE's using the forward Euler method — 42
- 4.4 Application: boundary value ODE's — 45
- 4.5 Exercises — 50

5 Solving nonlinear equations — 53

- 5.1 The bisection method — 54
- 5.2 Newton's method — 58
- 5.3 Secant method — 60
- 5.4 Comparing bisection, Newton, and secant methods — 61
- 5.5 Combining methods, inverse interpolation, and the *fzero* command — 62
- 5.6 Newton's method in higher dimensions — 64
- 5.7 Fixed point theory and algorithms — 67
 - 5.7.1 Nonlinear Helmholtz — 70
 - 5.7.2 Navier–Stokes — 71
 - 5.7.3 Anderson acceleration — 74
- 5.8 Exercises — 77

6 Accuracy in solving linear systems — 80

- 6.1 Gauss–Jordan elimination and finding matrix inverses — 80
- 6.2 Matrix and vector norms and condition number — 83
- 6.3 Sensitivity in linear system solving — 85
- 6.4 Exercises — 87

7 Eigenvalues and eigenvectors — 90

- 7.1 Mathematical definition — 90
- 7.2 Power method — 92
- 7.3 Application: population dynamics — 95
- 7.4 Exercises — 96

8 Fitting curves to data — 98

- 8.1 Interpolation — 98
 - 8.1.1 Interpolation by a single polynomial — 99
 - 8.1.2 Piecewise polynomial interpolation — 101
- 8.2 Curve fitting — 104
 - 8.2.1 Line of best fit — 104
 - 8.2.2 Curve of best fit — 107
- 8.3 Exercises — 110

9 Numerical integration — 113

- 9.1 Newton–Cotes methods — 113
- 9.2 Composite rules — 117
- 9.3 MATLAB's integral function — 122
- 9.4 Gauss quadrature — 122
- 9.5 Exercises — 125

10 Initial value ODEs — 128

- 10.1 Reduction of higher-order ODEs to first-order ODEs — **128**
- 10.2 Common methods and derivation from integration rules — **130**
- 10.2.1 Backward Euler — **131**
- 10.2.2 Crank–Nicolson — **132**
- 10.2.3 Runge–Kutta 4 — **133**
- 10.3 Comparison of speed of implicit versus explicit solvers — **134**
- 10.4 Stability of ODE solvers — **135**
- 10.4.1 Stability of forward Euler — **136**
- 10.4.2 Stability of backward Euler — **137**
- 10.4.3 Stability of Crank–Nicolson — **138**
- 10.4.4 Stability of Runge–Kutta 4 — **139**
- 10.5 Accuracy of ODE solvers — **139**
- 10.5.1 Forward Euler — **140**
- 10.5.2 Backward Euler — **140**
- 10.5.3 Crank–Nicolson — **141**
- 10.5.4 Runge–Kutta 4 — **142**
- 10.6 Summary, general strategy, and MATLAB ODE solvers — **143**
- 10.7 The 1D heat equation — **145**
- 10.8 Exercises — **152**

A Getting started with Octave and MATLAB — 155

- A.1 Basic operations — **155**
- A.2 Arrays — **158**
- A.3 Operating on arrays — **160**
- A.4 Script files — **161**
- A.5 Function files — **162**
- A.5.1 Inline functions — **162**
- A.5.2 Passing functions to other functions — **163**
- A.6 Outputting information — **163**
- A.7 Programming in MATLAB — **164**
- A.8 Plotting — **165**
- A.9 Exercises — **166**

Index — 169

