

Contents

Preface	xi
1 Foundations	1
1.1 Introductory Examples	4
1.1.1 Nonlinear Equations	4
1.1.2 Linear Systems	6
1.1.3 Numerical Integration	8
1.2 Useful Background	10
1.2.1 Results from Calculus	10
1.2.2 Results from Linear Algebra	11
1.2.3 A Little Information about Computers	13
1.3 Some Basic Issues	16
1.3.1 Error	16
1.3.2 Convergence	22
1.3.3 Getting Better Results	26
1.4 Using MATLAB	31
1.4.1 Command Window Computations	31
1.4.2 M-Files	35
1.4.3 Programming in MATLAB	37
1.4.4 Matrix Multiplication	39
1.5 Chapter Wrap-Up	41
2 Functions of One Variable	47
2.1 Bisection Method	50
2.2 Secant-Type Methods	54
2.2.1 Regula Falsi	55
2.2.2 Secant Method	58
2.2.3 Analysis	61
2.3 Newton's Method	64
2.4 Muller's Method	71
2.5 Minimization	76
2.5.1 Golden-Section Search	76
2.5.2 Brent's Method	79
2.6 Beyond the Basics	80
2.6.1 Using MATLAB's Functions	80
2.6.2 Laguerre's Method	82
2.6.3 Zeros of a Nonlinear Function	85
2.7 Chapter Wrap-Up	88

3	Solving Linear Systems: Direct Methods	95
3.1	Gaussian Elimination	98
3.1.1	Basic Method	98
3.1.2	Row Pivoting	107
3.2	Gauss-Jordan	112
3.2.1	Inverse of a Matrix	113
3.3	Tridiagonal Systems	114
3.4	Further Topics	119
3.4.1	MATLAB's Methods	119
3.4.2	Condition of a Matrix	121
3.4.3	Iterative Refinement	123
3.5	Chapter Wrap-Up	125
4	LU and QR Factorization	135
4.1	LU Factorization	138
4.1.1	Using Gaussian Elimination	138
4.1.2	Direct LU Factorization	146
4.1.3	Applications	150
4.2	Matrix Transformations	154
4.2.1	Householder Transformation	155
4.2.2	Givens Rotations	162
4.3	QR Factorization	164
4.3.1	Using Householder Transformations	164
4.3.2	Using Givens Rotations	166
4.4	Beyond the Basics	168
4.4.1	LU Factorization with Implicit Row Pivoting	168
4.4.2	Efficient Conversion to Hessenberg Form	170
4.4.3	Using MATLAB's Functions	171
4.5	Chapter Wrap-Up	172
5	Eigenvalues and Eigenvectors	179
5.1	Power Method	182
5.1.1	Basic Power Method	183
5.1.2	Rayleigh Quotient	186
5.1.3	Shifted Power Method	188
5.1.4	Accelerating Convergence	189
5.2	Inverse Power Method	190
5.2.1	General Inverse Power Method	192
5.2.2	Convergence	193
5.3	QR Method	194
5.3.1	Basic QR Method	194
5.3.2	Better QR Method	196
5.3.3	Finding Eigenvectors	198
5.3.4	Accelerating Convergence	199
5.4	Further Topics	202
5.4.1	Singular Value Decomposition	202
5.4.2	MATLAB's Methods	203
5.5	Chapter Wrap-Up	204

6	Solving Linear Systems: Iterative Methods	213
6.1	Jacobi Method	217
6.2	Gauss-Seidel Method	224
6.3	Successive Over-Relaxation	228
6.4	Beyond the Basics	232
6.4.1	MATLAB's Built-In Functions	232
6.4.2	Conjugate Gradient Methods	234
6.4.3	GMRES	238
6.4.4	Simplex Method	240
6.5	Chapter Wrap-Up	243
7	Nonlinear Functions of Several Variables	251
7.1	Nonlinear Systems	254
7.1.1	Newton's Method	254
7.1.2	Secant Methods	260
7.1.3	Fixed-Point Iteration	262
7.2	Minimization	264
7.2.1	Descent Methods	264
7.2.2	Quasi-Newton Methods	266
7.3	Further Topics	268
7.3.1	Levenberg-Marquardt Method	268
7.3.2	Nelder-Mead Simplex Search	269
7.4	Chapter Wrap-Up	270
8	Interpolation	275
8.1	Polynomial Interpolation	278
8.1.1	Lagrange Form	278
8.1.2	Newton Form	284
8.1.3	Difficulties	290
8.2	Hermite Interpolation	294
8.3	Piecewise Polynomial Interpolation	299
8.3.1	Piecewise Linear Interpolation	300
8.3.2	Piecewise Quadratic Interpolation	301
8.3.3	Piecewise Cubic Hermite Interpolation	304
8.3.4	Cubic Spline Interpolation	305
8.4	Beyond the Basics	312
8.4.1	Rational-Function Interpolation	312
8.4.2	Using MATLAB's Functions	316
8.5	Chapter Wrap-Up	323
9	Approximation	333
9.1	Least-Squares Approximation	336
9.1.1	Approximation by a Straight Line	336
9.1.2	Approximation by a Parabola	342
9.1.3	General Least-Squares Approximation	346
9.1.4	Approximation for Other Functional Forms	348

9.2	Continuous Least-Squares Approximation	350
9.2.1	Approximation Using Powers of x	350
9.2.2	Orthogonal Polynomials	352
9.2.3	Legendre Polynomials	354
9.2.4	Chebyshev Polynomials	356
9.3	Function Approximation at a Point	358
9.3.1	Padé Approximation	358
9.3.2	Taylor Approximation	361
9.4	Further Topics	362
9.4.1	Bezier Curves	362
9.4.2	Using MATLAB's Functions	364
9.5	Chapter Wrap-Up	366
10	Fourier Methods	373
10.1	Fourier Approximation and Interpolation	376
10.1.1	Derivation	381
10.1.2	Data on Other Intervals	384
10.2	Radix-2 Fourier Transforms	386
10.2.1	Discrete Fourier Transform	386
10.2.2	Fast Fourier Transform	387
10.2.3	Matrix Form of FFT	388
10.2.4	Algebraic Form of FFT	389
10.3	Mixed-Radix FFT	392
10.4	Using MATLAB's Functions	396
10.5	Chapter Wrap-Up	400
11	Numerical Differentiation and Integration	405
11.1	Differentiation	408
11.1.1	First Derivatives	408
11.1.2	Higher Derivatives	412
11.1.3	Partial Derivatives	413
11.1.4	Richardson Extrapolation	414
11.2	Numerical Integration	416
11.2.1	Trapezoid Rule	417
11.2.2	Simpson's Rule	420
11.2.3	Newton-Cotes Open Formulas	426
11.2.4	Extrapolation Methods	428
11.3	Quadrature	431
11.3.1	Gaussian Quadrature	431
11.3.2	Other Gauss-Type Quadratures	435
11.4	MATLAB's Methods	437
11.4.1	Differentiation	437
11.4.2	Integration	437
11.5	Chapter Wrap-Up	438

12 Ordinary Differential Equations: Fundamentals	445
12.1 Euler's Method	447
12.1.1 Geometric Introduction	447
12.1.2 Approximating the Derivative	448
12.1.3 Approximating the Integral	449
12.1.4 Using Taylor Series	451
12.2 Runge-Kutta Methods	452
12.2.1 Second-Order Runge-Kutta Methods	452
12.2.2 Third-Order Runge-Kutta Methods	457
12.2.3 Classic Runge-Kutta Method	459
12.2.4 Fourth-Order Runge-Kutta Methods	462
12.2.5 Fifth-Order Runge-Kutta Methods	464
12.2.6 Runge-Kutta-Fehlberg Methods	465
12.3 Multistep Methods	474
12.3.1 Adams-Bashforth Methods	476
12.3.2 Adams-Moulton Methods	479
12.3.3 Adams Predictor-Corrector Methods	480
12.3.4 Other Predictor-Corrector Methods	485
12.4 Further Topics	487
12.4.1 MATLAB's Methods	487
12.4.2 Consistency and Convergence	488
12.5 Chapter Wrap-Up	490
13 ODE: Systems, Stiffness, Stability	499
13.1 Systems	502
13.1.1 Systems of Two ODE	504
13.1.2 Euler's Method for Systems	510
13.1.3 Runge-Kutta Methods for Systems	512
13.1.4 Multistep Methods for Systems	516
13.1.5 Second-Order ODE	522
13.2 Stiff ODE	526
13.2.1 BDF Methods	527
13.2.2 Implicit Runge-Kutta Methods	529
13.3 Stability	530
13.3.1 A-Stable and Stiffly Stable Methods	532
13.3.2 Stability in the Limit	533
13.4 Further Topics	536
13.4.1 MATLAB's Methods for Stiff ODE	536
13.4.2 Extrapolation Methods	537
13.4.3 Rosenbrock Methods	539
13.4.4 Multivalued Methods	539
13.5 Chapter Wrap-Up	552

14 ODE: Boundary-Value Problems	561
14.1 Shooting Method	565
14.1.1 Linear ODE	565
14.1.2 Nonlinear ODE	570
14.2 Finite-Difference Method	576
14.2.1 Linear ODE	576
14.2.2 Nonlinear ODE	580
14.3 Function Space Methods	582
14.3.1 Collocation	582
14.3.2 Rayleigh-Ritz	586
14.4 Chapter Wrap-Up	588
15 Partial Differential Equations	593
15.1 Heat Equation: Parabolic PDE	598
15.1.1 Explicit Method	599
15.1.2 Implicit Method	604
15.1.3 Crank-Nicolson Method	608
15.1.4 Insulated Boundary	611
15.2 Wave Equation: Hyperbolic PDE	612
15.2.1 Explicit Method	614
15.2.2 Implicit Method	616
15.3 Poisson Equation: Elliptic PDE	618
15.4 Finite-Element Method for Elliptic PDE	622
15.4.1 Defining the Subregions	623
15.4.2 Defining the Basis Functions	624
15.4.3 Computing the Coefficients	626
15.4.4 Using MATLAB	629
15.5 Chapter Wrap-Up	634
 Bibliography	 643
 Answers	 653
 Index	 667

Preface

Of the many facets of mathematics, numerical computation is one of the most ancient. It is also one of the areas with the broadest relevance in the modern world. Ancient astronomers computed the seasons and the solstices; modern mathematical models allow highly accurate weather predictions. The Babylonians used a simple iterative method to compute the approximate value of square roots; today there are many powerful methods for solving nonlinear equations of great complexity. The practicality of various numerical methods depends on the available computational tools; simple methods are useful for quick solutions to small problems while more sophisticated methods are appropriate for large-scale scientific computing.

Applied Numerical Analysis Using MATLAB provides a clear explanation of the classic methods of numerical analysis; these form the basis for the more recent and more advanced methods, which are introduced in the latter part of each chapter. The title of the text conveys several key points about the perspective of the presentation.

First, the title clearly indicates the emphasis on the applied nature of the material. Each chapter (after the first) follows the same basic plan. The chapter is introduced with descriptions of applications in which the methods presented in the chapter are useful. The presentation of each method starts with some motivation for how the method works, i.e., how it can be applied. The presentation then proceeds to consider a numerical procedure (simple computer program) for carrying out the computations, before considering the theory behind the method. Examples throughout the chapter focus on applying the methods both to simple hypothetical situations (clarifying how the method works) and to simple applications (showing the method in a somewhat broader context). Finally, the exercises at the end of the chapter include problems for practicing the techniques, using the methods to solve applications from a variety of areas of science and engineering, investigating the computational aspects of the methods further, and extending one's understanding (perhaps in a more theoretical direction).

Next, the term “numerical analysis” implies that the goal is to obtain an accurate, but in most cases approximate, numerical solution to the problem under consideration. The terms “numerical analysis”, “numerical methods”, “numerical techniques”, and “scientific computing” each suggest a slightly different emphasis on the scale of the problem, the type of computing resources available, and the extent to which one is interested in “how it works” as compared to “why it works” and “when it works”. In this text, the analysis of each method follows the more applied introduction of the method, and the suggestions for further reading give references for more in-depth treatment of the theoretical aspects of the subject matter. “Theory guides practice; practice inspires theory”. There are many instances in which scientists and engineers developed computational methods for solving problems that were later refined and put on a firm theoretical foundation