# Contents

## PREFACE

## CHAPTER 0  Fundamentals

- **0.1 Evaluating a Polynomial**
- **0.2 Binary Numbers**
  - 0.2.1 Decimal to binary
  - 0.2.2 Binary to decimal
- **0.3 Floating Point Representation of Real Numbers**
  - 0.3.1 Floating point formats
  - 0.3.2 Machine representation
  - 0.3.3 Addition of floating point numbers
- **0.4 Loss of Significance**
- **0.5 Review of Calculus**
  - Software and Further Reading

## CHAPTER 1  Solving Equations

- **1.1 The Bisection Method**
  - 1.1.1 Bracketing a root
  - 1.1.2 How accurate and how fast?
- **1.2 Fixed-Point Iteration**
  - 1.2.1 Fixed points of a function
  - 1.2.2 Geometry of Fixed-Point Iteration
  - 1.2.3 Linear convergence of Fixed-Point Iteration
  - 1.2.4 Stopping criteria
- **1.3 Limits of Accuracy**
  - 1.3.1 Forward and backward error
  - 1.3.2 The Wilkinson polynomial
  - 1.3.3 Sensitivity of root-finding
- **1.4 Newton's Method**
  - 1.4.1 Quadratic convergence of Newton’s Method
  - 1.4.2 Linear convergence of Newton’s Method
- **1.5 Root-Finding without Derivatives**
  - 1.5.1 Secant Method and variants
  - 1.5.2 Brent's Method

### Reality Check 1: Kinematics of the Stewart platform

- Software and Further Reading
CHAPTER 2  Systems of Equations 74
   2.1 Gaussian Elimination 74
      2.1.1 Naive Gaussian elimination 75
      2.1.2 Operation counts 77
   2.2 The LU Factorization 82
      2.2.1 Matrix form of Gaussian elimination 82
      2.2.2 Back substitution with the LU factorization 85
      2.2.3 Complexity of the LU factorization 86
   2.3 Sources of Error 89
      2.3.1 Error magnification and condition number 89
      2.3.2 Swamping 95
   2.4 The PA = LU Factorization 99
      2.4.1 Partial pivoting 99
      2.4.2 Permutation matrices 101
      2.4.3 PA = LU factorization 102

Reality Check 2: The Euler–Bernoulli Beam 107

CHAPTER 3  Interpolation 144
   3.1 Data and Interpolating Functions 145
      3.1.1 Lagrange interpolation 146
      3.1.2 Newton’s divided differences 147
      3.1.3 How many degree \( d \) polynomials pass through \( n \) points? 150
      3.1.4 Code for interpolation 151
      3.1.5 Representing functions by approximating polynomials 153
   3.2 Interpolation Error 157
      3.2.1 Interpolation error formula 158
      3.2.2 Proof of Newton form and error formula 159
      3.2.3 Runge phenomenon 162
   3.3 Chebyshev Interpolation 164
      3.3.1 Chebyshev’s theorem 165
      3.3.2 Chebyshev polynomials 167
      3.3.3 Change of interval 169
3.4 Cubic Splines
  3.4.1 Properties of splines 174
  3.4.2 Endpoint conditions 180
3.5 Bézier Curves 185

Reality Check 3: Fonts from Bézier curves 190
Software and Further Reading 194

CHAPTER 4 Least Squares 196
4.1 Least Squares and the Normal Equations 196
  4.1.1 Inconsistent systems of equations 197
  4.1.2 Fitting models to data 201
  4.1.3 Conditioning of least squares 205
4.2 A Survey of Models 208
  4.2.1 Periodic data 208
  4.2.2 Data linearization 211
4.3 QR Factorization 220
  4.3.1 Gram–Schmidt orthogonalization and least squares 220
  4.3.2 Modified Gram–Schmidt orthogonalization 227
  4.3.3 Householder reflectors 228
4.4 Generalized Minimum Residual (GMRES) Method 235
  4.4.1 Krylov methods 235
  4.4.2 Preconditioned GMRES 237
4.5 Nonlinear Least Squares 240
  4.5.1 Gauss–Newton Method 240
  4.5.2 Models with nonlinear parameters 243
   4.5.3 The Levenberg–Marquardt Method. 245
Reality Check 4: GPS, Conditioning, and Nonlinear Least Squares 248
Software and Further Reading 251

CHAPTER 5 Numerical Differentiation and Integration 253
5.1 Numerical Differentiation 254
  5.1.1 Finite difference formulas 254
  5.1.2 Rounding error 257
  5.1.3 Extrapolation 259
  5.1.4 Symbolic differentiation and integration 261
5.2 Newton–Cotes Formulas for Numerical Integration 264
  5.2.1 Trapezoid Rule 265
  5.2.2 Simpson's Rule 267
  5.2.3 Composite Newton–Cotes formulas 269
  5.2.4 Open Newton–Cotes Methods 272
5.3 Romberg Integration 276
5.4 Adaptive Quadrature 279
5.5 Gaussian Quadrature 284
Reality Check 5: Motion Control in Computer-Aided Modeling 289
Software and Further Reading 291
**CHAPTER 6  Ordinary Differential Equations**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Initial Value Problems</td>
<td>294</td>
</tr>
<tr>
<td>6.1.1 Euler’s Method</td>
<td>295</td>
</tr>
<tr>
<td>6.1.2 Existence, uniqueness, and continuity for solutions</td>
<td>300</td>
</tr>
<tr>
<td>6.1.3 First-order linear equations</td>
<td>303</td>
</tr>
<tr>
<td>6.2 Analysis of IVP Solvers</td>
<td>306</td>
</tr>
<tr>
<td>6.2.1 Local and global truncation error</td>
<td>306</td>
</tr>
<tr>
<td>6.2.2 The explicit Trapezoid Method</td>
<td>310</td>
</tr>
<tr>
<td>6.2.3 Taylor Methods</td>
<td>313</td>
</tr>
<tr>
<td>6.3 Systems of Ordinary Differential Equations</td>
<td>316</td>
</tr>
<tr>
<td>6.3.1 Higher order equations</td>
<td>317</td>
</tr>
<tr>
<td>6.3.2 Computer simulation: the pendulum</td>
<td>318</td>
</tr>
<tr>
<td>6.3.3 Computer simulation: orbital mechanics</td>
<td>322</td>
</tr>
<tr>
<td>6.4 Runge–Kutta Methods and Applications</td>
<td>328</td>
</tr>
<tr>
<td>6.4.1 The Runge–Kutta family</td>
<td>328</td>
</tr>
<tr>
<td>6.4.2 Computer simulation: the Hodgkin–Huxley neuron</td>
<td>331</td>
</tr>
<tr>
<td>6.4.3 Computer simulation: the Lorenz equations</td>
<td>333</td>
</tr>
<tr>
<td>Reality Check 6: The Tacoma Narrows Bridge</td>
<td>337</td>
</tr>
<tr>
<td>6.5 Variable Step-Size Methods</td>
<td>340</td>
</tr>
<tr>
<td>6.5.1 Embedded Runge–Kutta pairs</td>
<td>340</td>
</tr>
<tr>
<td>6.5.2 Order 4/5 methods</td>
<td>342</td>
</tr>
<tr>
<td>6.6 Implicit Methods and Stiff Equations</td>
<td>347</td>
</tr>
<tr>
<td>6.7 Multistep Methods</td>
<td>351</td>
</tr>
<tr>
<td>6.7.1 Generating multistep methods</td>
<td>352</td>
</tr>
<tr>
<td>6.7.2 Explicit multistep methods</td>
<td>354</td>
</tr>
<tr>
<td>6.7.3 Implicit multistep methods</td>
<td>359</td>
</tr>
<tr>
<td>Software and Further Reading</td>
<td>365</td>
</tr>
</tbody>
</table>

**CHAPTER 7  Boundary Value Problems**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Shooting Method</td>
<td>367</td>
</tr>
<tr>
<td>7.1.1 Solutions of boundary value problems</td>
<td>367</td>
</tr>
<tr>
<td>7.1.2 Shooting Method implementation</td>
<td>370</td>
</tr>
<tr>
<td>Reality Check 7: Buckling of a Circular Ring</td>
<td>374</td>
</tr>
<tr>
<td>7.2 Finite Difference Methods</td>
<td>376</td>
</tr>
<tr>
<td>7.2.1 Linear boundary value problems</td>
<td>376</td>
</tr>
<tr>
<td>7.2.2 Nonlinear boundary value problems</td>
<td>378</td>
</tr>
<tr>
<td>7.3 Collocation and the Finite Element Method</td>
<td>384</td>
</tr>
<tr>
<td>7.3.1 Collocation</td>
<td>384</td>
</tr>
<tr>
<td>7.3.2 Finite elements and the Galerkin Method</td>
<td>387</td>
</tr>
<tr>
<td>Software and Further Reading</td>
<td>392</td>
</tr>
</tbody>
</table>
## CHAPTER 8 Partial Differential Equations 394

### 8.1 Parabolic Equations 395
- 8.1.1 Forward Difference Method 395
- 8.1.2 Stability analysis of Forward Difference Method 399
- 8.1.3 Backward Difference Method 400
- 8.1.4 Crank–Nicolson Method 405

### 8.2 Hyperbolic Equations 413
- 8.2.1 The wave equation 413
- 8.2.2 The CFL condition 415

### 8.3 Elliptic Equations 419
- 8.3.1 Finite Difference Method for elliptic equations 420
- 8.3.2 Finite Element Method for elliptic equations 427

**Reality Check 8:** Heat distribution on a cooling fin 424

### 8.4 Nonlinear partial differential equations 438
- 8.4.1 Implicit Newton solver 438
- 8.4.2 Nonlinear equations in two space dimensions 444

Software and Further Reading 451

## CHAPTER 9 Random Numbers and Applications 453

### 9.1 Random Numbers 454
- 9.1.1 Pseudo-random numbers 454
- 9.1.2 Exponential and normal random numbers 459

### 9.2 Monte Carlo Simulation 462
- 9.2.1 Power laws for Monte Carlo estimation 462
- 9.2.2 Quasi-random numbers 464

### 9.3 Discrete and Continuous Brownian Motion 469
- 9.3.1 Random walks 469
- 9.3.2 Continuous Brownian motion 472

### 9.4 Stochastic Differential Equations 474
- 9.4.1 Adding noise to differential equations 475
- 9.4.2 Numerical methods for SDEs 478

**Reality Check 9:** The Black–Scholes Formula 486

Software and Further Reading 488

## CHAPTER 10 Trigonometric Interpolation and the FFT 489

### 10.1 The Fourier Transform 490
- 10.1.1 Complex arithmetic 490
- 10.1.2 Discrete Fourier Transform 493
- 10.1.3 The Fast Fourier Transform 495

### 10.2 Trigonometric Interpolation 498
- 10.2.1 The DFT Interpolation Theorem 498
- 10.2.2 Efficient evaluation of trigonometric functions 502

### 10.3 The FFT and Signal Processing 505
- 10.3.1 Orthogonality and interpolation 506
- 10.3.2 Least squares fitting with trigonometric functions 508
- 10.3.3 Sound, noise, and filtering 512

**Reality Check 10:** The Wiener Filter 515

Software and Further Reading 517
CHAPTER 11 Compression 518

11.1 The Discrete Cosine Transform 519
11.1.1 One-dimensional DCT 519
11.1.2 The DCT and least squares approximation 521

11.2 Two-Dimensional DCT and Image Compression 524
11.2.1 Two-dimensional DCT 524
11.2.2 Image compression 528
11.2.3 Quantization 531

11.3 Huffman Coding 538
11.3.1 Information theory and coding 538
11.3.2 Huffman coding for the JPEG format 541

11.4 Modified DCT and Audio Compression 544
11.4.1 Modified Discrete Cosine Transform 544
11.4.2 Bit quantization 550

Reality Check 11: A Simple Audio Codec 552

Software and Further Reading 555

CHAPTER 12 Eigenvalues and Singular Values 556

12.1 Power Iteration Methods 556
12.1.1 Power Iteration 557
12.1.2 Convergence of Power Iteration 559
12.1.3 Inverse Power Iteration 560
12.1.4 Rayleigh Quotient Iteration 562

12.2 QR Algorithm 564
12.2.1 Simultaneous iteration 565
12.2.2 Real Schur form and the QR algorithm 567
12.2.3 Upper Hessenberg form 570

Reality Check 12: How Search Engines Rate Page Quality 575

12.3 Singular Value Decomposition 578
12.3.1 Geometry of the SVD 578
12.3.2 Finding the SVD in general 581

12.4 Applications of the SVD 585
12.4.1 Properties of the SVD 585
12.4.2 Dimension reduction 587
12.4.3 Compression 588
12.4.4 Calculating the SVD 590

Software and Further Reading 592
## Contents

### CHAPTER 13 Optimization 593

**13.1 Unconstrained Optimization without Derivatives** 594
  - 13.1.1 Golden Section Search 594
  - 13.1.2 Successive parabolic interpolation 597
  - 13.1.3 Nelder–Mead search 600

**13.2 Unconstrained Optimization with Derivatives** 604
  - 13.2.1 Newton's Method 604
  - 13.2.2 Steepest Descent 605
  - 13.2.3 Conjugate Gradient Search 606

**Reality Check 13:** Molecular Conformation and Numerical Optimization 609
  - Software and Further Reading 610

### Appendix A: Matrix Algebra 612

**A.1 Matrix Fundamentals** 612

**A.2 Systems of linear equations** 614

**A.3 Block Multiplication** 615

**A.4 Eigenvalues and Eigenvectors** 616

**A.5 Symmetric Matrices** 617

**A.6 Vector Calculus** 618

### Appendix B: Introduction to Matlab 620

**B.1 Starting MATLAB** 620

**B.2 Graphics** 620

**B.3 Programming in MATLAB** 621

**B.4 Flow Control** 623

**B.5 Functions** 625

**B.6 Matrix Operations** 627

**B.7 Animation and Movies** 628

### ANSWERS TO SELECTED EXERCISES 630

### BIBLIOGRAPHY 646

### INDEX 652


**Preface**

*Numerical Analysis* is a text for students of engineering, science, mathematics, and computer science who have completed elementary calculus and matrix algebra. The primary goal is to construct and explore algorithms for solving science and engineering problems. The not-so-secret secondary mission is to help the reader locate these algorithms in a landscape of some potent and far-reaching principles. These unifying principles, taken together, constitute a dynamic field of current research and development in modern numerical and computational science.

The discipline of numerical analysis is jam-packed with useful ideas. Textbooks run the risk of presenting the subject as a bag of neat but unrelated tricks. For a deep understanding, readers need to learn much more than how to code Newton’s Method, Runge-Kutta, and the Fast Fourier Transform. They must absorb the big principles, the ones that permeate numerical analysis and integrate its competing concerns of accuracy and efficiency.

The notions of convergence, complexity, conditioning, compression, and orthogonality are among the most important of the big ideas. Any approximation method worth its salt must converge to the correct answer as more computational resources are devoted to it, and the complexity of a method is a measure of its use of these resources. The conditioning of a problem, or susceptibility to error magnification, is fundamental to knowing how it can be attacked. Many of the newest applications of numerical analysis strive to realize data in a shorter or compressed way. Finally, orthogonality is crucial for efficiency in many algorithms, and is irreplaceable where conditioning is an issue or compression is a goal.

In this book, the roles of these five concepts in modern numerical analysis are emphasized in short thematic elements labeled *Spotlight*. They comment on the topic at hand and make informal connections to other expressions of the same concept elsewhere in the book. We hope that highlighting the five concepts in such an explicit way functions as a Greek chorus, accentuating what is really crucial about the theory on the page.

Although it is common knowledge that the ideas of numerical analysis are vital to the practice of modern science and engineering, it never hurts to be obvious. The feature entitled *Reality Check* provide concrete examples of the way numerical methods lead to solutions of important scientific and technological problems. These extended applications were chosen to be timely and close to everyday experience. Although it is impossible (and probably undesirable) to present the full details of the problems, the Reality Checks attempt to go deeply enough to show how a technique or algorithm can leverage a small amount of mathematics into a great payoff in technological design and function. The Reality Checks were popular as a source of student projects in previous editions, and they have been extended and amplified in this edition.

**NEW TO THIS EDITION**

Features of the third edition include:

- Short URLs in the side margin of the text (235 of them in all) take students directly to relevant content that supports their use of the textbook. Specifically:
  - MATLAB Code: Longer instances of MATLAB code are available for students in *.m format. The homepage for all of the instances of MATLAB code is goo.gl/VxzXyw.
xii | Preface

- **Solutions to Selected Exercises**: This text used to be supported by a Student Solutions Manual that was available for purchase separately. In this edition we are providing students with access solutions to selected exercises online at no extra charge. The homepage for the selected solutions is goo.gl/2j5gI7.

- **Additional Examples**: Each section of the third edition is enhanced with extra new examples, designed to reinforce the text exposition and to ease the reader's transition to active solution of exercises and computer problems. The full worked-out details of these examples, more than one hundred in total, are available online. Some of the solutions are in video format (created by the author). The homepage for the solutions to Additional Examples is goo.gl/lFQb0B.

- **NOTE**: The homepage for all web content supporting the text is goo.gl/zQNJeP.

- More detailed discussion of several key concepts has been added in this edition, including theory of polynomial interpolation, multi-step differential equation solvers, boundary value problems, and the singular value decomposition, among others.

- The Reality Check on audio compression in Chapter 11 has been refurbished and simplified, and other MATLAB codes have been added and updated throughout the text.

- Several dozen new exercises and computer problems have been added to the third edition.

### TECHNOLOGY

The software package MATLAB is used both for exposition of algorithms and as a suggested platform for student assignments and projects. The amount of MATLAB code provided in the text is carefully modulated, due to the fact that too much tends to be counterproductive. More MATLAB code is found in the early chapters, allowing the reader to gain proficiency in a gradual manner. Where more elaborate code is provided (in the study of interpolation, and ordinary and partial differential equations, for example), the expectation is for the reader to use what is given as a jumping-off point to exploit and extend.

It is not essential that any particular computational platform be used with this textbook, but the growing presence of MATLAB in engineering and science departments shows that a common language can smooth over many potholes. With MATLAB, all of the interface problems—data input/output, plotting, and so on—are solved in one fell swoop. Data structure issues (for example those that arise when studying sparse matrix methods) are standardized by relying on appropriate commands. MATLAB has facilities for audio and image file input and output. Differential equations simulations are simple to realize due to the animation commands built into MATLAB. These goals can all be achieved in other ways. But it is helpful to have one package that will run on almost all operating systems and simplify the details so that students can focus on the real mathematical issues. Appendix B is a MATLAB tutorial that can be used as a first introduction to students, or as a reference for those already familiar.

### SUPPLEMENTS

The Instructor’s Solutions Manual contains detailed solutions to the odd-numbered exercises, and answers to the even-numbered exercises. The manual also shows how to
use MATLAB software as an aid to solving the types of problems that are presented in the Exercises and Computer Problems.

**DESIGNING THE COURSE**

*Numerical Analysis* is structured to move from foundational, elementary ideas at the outset to more sophisticated concepts later in the presentation. Chapter 0 provides fundamental building blocks for later use. Some instructors like to start at the beginning; others (including the author) prefer to start at Chapter 1 and fold in topics from Chapter 0 when required. Chapters 1 and 2 cover equation-solving in its various forms. Chapters 3 and 4 primarily treat the fitting of data, interpolation and least squares methods. In chapters 5–8, we return to the classical numerical analysis areas of continuous mathematics: numerical differentiation and integration, and the solution of ordinary and partial differential equations with initial and boundary conditions.

Chapter 9 develops random numbers in order to provide complementary methods to Chapters 5–8: the Monte-Carlo alternative to the standard numerical integration schemes and the counterpoint of stochastic differential equations are necessary when uncertainty is present in the model.

Compression is a core topic of numerical analysis, even though it often hides in plain sight in interpolation, least squares, and Fourier analysis. Modern compression techniques are featured in Chapters 10 and 11. In the former, the Fast Fourier Transform is treated as a device to carry out trigonometric interpolation, both in the exact and least squares sense. Links to audio compression are emphasized, and fully carried out in Chapter 11 on the Discrete Cosine Transform, the standard workhorse for modern audio and image compression. Chapter 12 on eigenvalues and singular values is also written to emphasize its connections to data compression, which are growing in importance in contemporary applications. Chapter 13 provides a short introduction to optimization techniques.

*Numerical Analysis* can also be used for a one-semester course with judicious choice of topics. Chapters 0–3 are fundamental for any course in the area. Separate one-semester tracks can be designed as follows:

**ACKNOWLEDGMENTS**

The third edition owes a debt to many people, including the students of many classes who have read and commented on earlier versions. In addition, Paul Lorczak was
essential in helping me avoid embarrassing blunders. The resourceful staff at Pearson, including Jeff Weidenaar, Jenn Snyder, Yvonne Vannatta, and Tara Corpuz, made the production of the third edition almost enjoyable. Finally, thanks are due to the helpful readers from other universities for their encouragement of this project and indispensable advice for improvement of earlier versions:

- Eugene Allgower, Colorado State University
- Constantin Bacuta, University of Delaware
- Michele Benzi, Emory University
- Jerry Bona, University of Illinois at Chicago
- George Davis, Georgia State University
- Chris Danforth, University of Vermont
- Alberto Delgado, Illinois State University
- Robert Dillon, Washington State University
- Qiang Du, Columbia University
- Ahmet Duran, University of Michigan
- Gregory Goeckel, Presbyterian College
- Herman Gollwitzer, Drexel University
- Weimin Han, University of Iowa *
- Don Hardcastle, Baylor University
- David R. Hill, Temple University
- Alberto Jimenez, California Polytechnic State University *
- Hideaki Kaneko, Old Dominion University
- Ashwani Kapila, Rensselaer Polytechnic Institute *
- Daniel Kaplan, Macalester College
- Fritz Keinert, Iowa State University
- Akhtar A. Khan, Rochester Institute of Technology
- Lucia M. Kimball, Bentley College
- Colleen M. Kirk, California Polytechnic State University
- Seppo Korpela, Ohio State University
- William Layton, University of Pittsburgh
- Brenton LeMesurier, College of Charleston
- Melvin Leok, University of California, San Diego
- Doron Levy, University of Maryland
- Bo Li, University of California, San Diego *
- Jiaxi Liu, University of North Texas *
- Mark Lyon, University of New Hampshire *
- Shankar Mahalingam, University of Alabama, Huntsville
- Amnon Meir, Southern Methodist University
- Peter Monk, University of Delaware
- Joseph E. Pasciak, Texas A&M University
- Jeff Parker, Harvard University
- Jacek Polewczak, California State University
- Jorge Rebaza, Missouri State University
- Jeffrey Scroggs, North Carolina State University
- David Stewart, University of Iowa *
- David Stowell, Brigham Young University *
- Sergei Suslov, Arizona State University
- Daniel Szyld, Temple University
- Ahlam Tannouri, Morgan State University
• Janos Turi, University of Texas, Dallas *
• Jin Wang, Old Dominion University
• Bruno Welfert, Arizona State University
• Nathaniel Whitaker, University of Massachusetts

* Contributed to the current edition