

Mark H. Holmes  
Department of Mathematical Sciences  
Rensselaer Polytechnic Institute  
Troy, NY 12180-3590

Introduction to  
Scientific Computing and  
Data Analysis

July 7, 2023

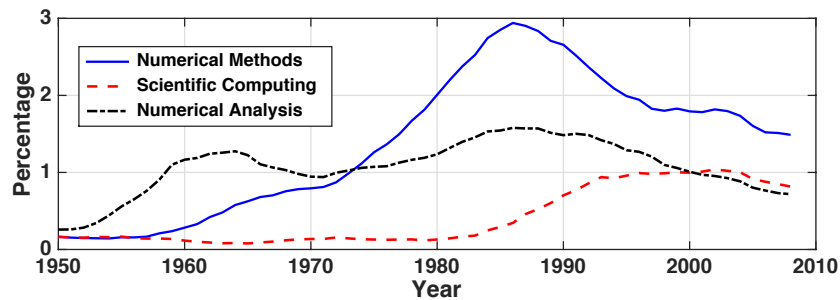
Springer



# Preface

The objective of this text is easy to state, and it is to investigate ways to use a computer to solve various mathematical problems. One of the challenges for those learning this material is that it involves a nonlinear combination of mathematical analysis and nitty-gritty computer programming. Texts vary considerably in how they balance these two aspects of the subject. You can see this in the brief history of the subject given in Figure 0.1 (which is an example of what is called an ngram plot). According to this plot, the earlier books concentrated more on the analysis (theory). In the early 1970's this changed and there was more of an emphasis on methods (which generally means much less theory), and these continue to dominant the area today. However, the 1980's saw the advent of scientific computing books, which combine theory and programming, and you can see a subsequent decline in the other two types of books when this occurred. This text falls within this latter group.

There are two important threads running through the text. One concerns understanding the mathematical problem that is being solved. As an example,



**Figure 0.1** Historical record according to Google. The values are the number of instances that the expression appeared in a published book in the respective year, expressed as a percentage for that year, times  $10^5$  [Michel et al., 2011].

when using Newton's method to solve  $f(x) = 0$ , the usual statement is that it will work if you guess a starting value close to the solution. It is important to know how to determine good starting points, and, perhaps even more importantly, knowing whether the problem being solved even has a solution. Consequently, when deriving Newton's method, and others like it, an effort is made to explain how to fairly easily answer these questions.

The second theme is the importance in scientific computing of having a solid grasp of the theory underlying the methods being used. A computer has the unfortunate ability to produce answers even if the methods used to find the solution are completely wrong. Consequently, it is essential to have an understanding of how the method works, and how the error in the computation depends on the method being used.

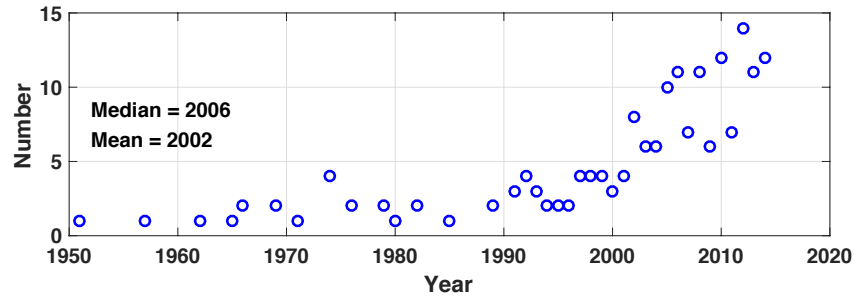
Needless to say, is also important to be able to code these methods, and in the process be able to adapt them to the particular problem being solved. There is considerable room for interpretation on what this means. To explain, in terms of computing languages, the current favorites are MATLAB and Python. Using the commands they provide, a text such as this one becomes more of a user's manual, reducing the entire book down to a few commands. For example, with MATLAB, this book (as well as most others in this area) can be replaced with the following commands:

Chapter 1: `eps`  
Chapter 2: `fzero(@f,x0)`  
Chapter 3: `A\b`  
Chapter 4: `eig(A)`  
Chapter 5: `polyfit(x,y,n)`  
Chapter 6: `integral(@f,a,b)`  
Chapter 7: `ode45(@f,tspan,y0)`  
Chapter 8: `fminsearch(@fun,x0)`  
Chapter 9: `svd(A)`

Certainly this statement qualifies as hyperbole, and as an example, Chapters 4 and 5 should probably have two commands listed. The other extreme is to write all of the methods from scratch, something that was expected of students in the early days of computing. In the end, the level of coding depends on what the learning outcomes are for the course, and the background and computing prerequisites required for the course.

Many of the topics included are typical of what are found in an upper division scientific computing course. There are also notable additions. This includes material related to data analysis, as well as variational methods and derivative free minimization methods. Moreover, there are differences related to emphasis. An example here concerns the preeminent role matrix factorizations play in numerical linear algebra, and this is made evident in the development of the material.

The coverage of any particular topic is not exhaustive, but intended to introduce the basic ideas. For this reason, numerous references are provided



**Figure 0.2** Number of references in this book, after 1950, as a function of the year they were published.

for those who might be interested in further study, and many of these are from the current research literature. To quantify this statement, a code was written that reads the *tex.bbl* file containing the references for this text, and then uses MATLAB to plot the number as a function of the year published. The result is Figure 0.2, and it shows that approximately half of the references were published in the last ten years. By the way, in terms of data generation and plotting, Figure 0.1 was produced by writing a code which reads the html source code for the ngram web-page and then uses MATLAB to produce the plot.

The MATLAB codes used to produce almost every figure, and table with numerical output, in this text are available from the author's web-site as well as from SpringerLink. In other words, the MATLAB codes for all of the methods considered, and the examples used, are available. These can be used as a learning tool. This also goes to the importance in computational based research, and education, of providing open source to guarantee the correctness and reproducibility of the work. Some interesting comments on this can be found in Morin et al. [2012] and Peng [2011].

The prerequisites depend on which chapters are covered, but the typical two year lower division mathematics program (consisting of calculus, matrix algebra and differential equations) should be sufficient for the entire text. However, one topic plays an oversized role in this subject, and this is Taylor's theorem. This also tends to be the topic that students had the most trouble with in calculus. For this reason, an appendix is included that reviews some of the more pertinent aspects of Taylor's theorem. It should also be pointed out that there are numerous theorems in the text, as well as an outline of the proof for many of them. These should be read with care because they contain information that is useful when testing the code that implements the respective method (i.e., they provide one of the essential ways we will have to make sure the computed results are actually correct).

I would like to thank the reviewers of an early draft of the book, who made several very constructive suggestions to improve the text. Also, as usual, I

would like to thank those who developed and have maintained TeXShop, a free and very good TeX previewer.

Troy, New York  
January, 2016

*Mark H. Holmes*

## Preface to Second Edition

One of the principal reasons for this edition is to include material necessary for an upper division course in computational linear algebra. So, least squares is covered more extensively, along with new material covering Householder QR, sparse matrix methods, preconditioning, and Markov chains. The computational implications of the Gershgorin, Perron-Frobenius, and Eckart-Young theorems have also been expanded. It is assumed in the development that a previous course in linear algebra has been taken, but as a refresher some of the more pertinent facts from such a course are given in an appendix.

Other significant changes include a reorganization and expansion of the exercises. Also, the material on cubic splines, particularly related to data analysis, has been expanded (e.g., Section 6.4), and the presentation of Gaussian quadrature has been modified. There is also a new section providing an introduction to data based modeling and dynamic modes.

As with the first edition, the material is developed, and presented, independent of the language used for computing. So, there are no computer codes in the text. Rather, the procedures are written in a generic format, such as Newton's method on page 41. There are, however, a few exercises where example MATLAB commands are given indicating how the problem can be done (e.g., the commands on page 198 for inputting, and displaying, a grayscale image). All of the codes used in the computational examples in the text are available from the author's GitHub repository (<https://github.com/HolmesRPI/IntroSciComp2nd>). Other files for the book, such as an errata page and the answers for selected exercises, are also there.

Troy, New York  
August, 2022

*Mark H. Holmes*



# Contents

<b>Preface</b> .....	i
<b>Preface to Second Edition</b> .....	v
<b>1 Introduction to Scientific Computing</b> .....	1
1.1 Unexpected Results .....	1
1.2 Floating-Point Number System .....	5
1.2.1 Normal Floats .....	5
1.2.2 Rounding .....	8
1.2.3 Non-Normal Floats .....	9
1.2.4 Significance .....	10
1.2.5 Flops .....	10
1.2.6 Functions .....	11
1.3 Arbitrary-Precision Arithmetic .....	12
1.4 Explaining, and Possibly Fixing, the Unexpected Results . . . .	12
1.5 Error and Accuracy .....	18
1.5.1 Over-computing? .....	20
1.6 Multicore Computing .....	21
<b>2 Solving A Nonlinear Equation</b> .....	31
2.1 The Problem to Solve .....	32
2.2 Bisection Method .....	33
2.2.1 Convergence Theory .....	36
2.3 Newton's Method .....	39
2.3.1 Picking $x_0$ .....	42
2.3.2 Order of Convergence .....	43
2.3.3 Failure .....	45
2.3.4 Examples .....	46
2.3.5 Convergence Theorem .....	48
2.4 Secant Method .....	50
2.4.1 Convergence Theorem .....	53

2.5	Other Ideas . . . . .	54
2.5.1	Is Newton's Method Really Newton's Method? . . . . .	54
<b>3</b>	<b>Matrix Equations . . . . .</b>	<b>71</b>
3.1	An Example . . . . .	71
3.2	Finding L and U . . . . .	74
3.2.1	What Matrices Have a LU Factorization? . . . . .	75
3.3	Implementing a LU Solver . . . . .	76
3.3.1	Pivoting Strategies . . . . .	78
3.3.2	LU and Gaussian Elimination . . . . .	78
3.4	LU Method: Summary . . . . .	80
3.4.1	Flops and Multicore Processors . . . . .	85
3.5	Vector and Matrix Norms . . . . .	86
3.5.1	Matrix Norm . . . . .	88
3.6	Error and Residual . . . . .	91
3.6.1	Correct Significant Digits . . . . .	92
3.6.2	The Condition Number . . . . .	92
3.6.3	A Heuristic for the Error . . . . .	97
3.7	Positive Definite Matrices . . . . .	98
3.7.1	Cholesky Factorization . . . . .	101
3.8	Tri-Diagonal Matrices . . . . .	103
3.9	Sparse Matrices . . . . .	105
3.10	Nonlinear Systems . . . . .	106
3.11	Some Additional Ideas . . . . .	109
3.11.1	Yogi Berra and Perturbation Theory . . . . .	109
3.11.2	Fixing an Ill-Conditioned Matrix . . . . .	110
3.11.3	Insightful Observations about the Condition Number . . . . .	110
3.11.4	Faster than LU? . . . . .	111
3.11.5	Historical Comparisons . . . . .	112
<b>4</b>	<b>Eigenvalue Problems . . . . .</b>	<b>127</b>
4.1	Review of Eigenvalue Problems . . . . .	127
4.2	Power Method . . . . .	132
4.2.1	General Formulation . . . . .	136
4.3	Extensions of the Power Method . . . . .	139
4.3.1	Inverse Iteration . . . . .	141
4.3.2	Rayleigh Quotient Iteration . . . . .	143
4.4	Calculating Multiple Eigenvalues . . . . .	145
4.4.1	Orthogonal Iteration . . . . .	146
4.4.2	Regular and Modified Gram-Schmidt . . . . .	149
4.4.3	QR Factorization . . . . .	152
4.4.4	The QR Method . . . . .	153
4.4.5	QR Method versus Orthogonal Iteration . . . . .	155
4.4.6	Are the Computed Values Correct? . . . . .	156
4.5	Applications . . . . .	159

4.5.1	Natural Frequencies	159
4.5.2	Graphs and Adjacency Matrices	161
4.5.3	Markov Chain	163
4.6	Singular Value Decomposition	167
4.6.1	Derivation of the SVD	168
4.6.2	Interpretations of the SVD	171
4.6.3	Summary of the SVD	173
4.6.4	Consequences of a SVD	176
4.6.5	Computing a SVD	177
4.6.6	Low-Rank Approximations	177
4.6.7	Application: Image Compression	178
<b>5</b>	<b>Interpolation</b>	<b>203</b>
5.1	Information from Data	203
5.2	Global Polynomial Interpolation	205
5.2.1	Direct Approach	205
5.2.2	Lagrange Approach	207
5.2.3	Runge's Function	210
5.3	Piecewise Linear Interpolation	211
5.4	Piecewise Cubic Interpolation	215
5.4.1	Cubic B-splines	218
5.5	Function Approximation	224
5.5.1	Global Polynomial Interpolation	224
5.5.2	Piecewise Linear Interpolation	226
5.5.3	Cubic Splines	228
5.5.4	Chebyshev Interpolation	230
5.5.5	Chebyshev versus Cubic Splines	235
5.5.6	Other Ideas	237
5.6	Questions and Additional Comments	237
<b>6</b>	<b>Numerical Integration</b>	<b>253</b>
6.1	The Definition from Calculus	254
6.1.1	Midpoint Rule	256
6.2	Methods Based on Polynomial Interpolation	259
6.2.1	Trapezoidal Rule	259
6.2.2	Simpson's Rule	263
6.2.3	Cubic Splines	266
6.2.4	Other Interpolation Ideas	269
6.3	Romberg Integration	270
6.3.1	Computing Using Romberg	271
6.4	Methods for Discrete Data	272
6.5	Methods Based on Precision	275
6.5.1	1-Point Gaussian Rule	276
6.5.2	2-Point Gaussian Rule	277
6.5.3	$m$ -Point Gaussian Rule	279

6.5.4	Error	279
6.5.5	Parting Comments	280
6.6	Adaptive Quadrature	281
6.7	Other Ideas	285
6.8	Epilogue	285
<b>7</b>	<b>Initial Value Problems</b>	<b>301</b>
7.1	Solution of an IVP	301
7.2	Numerical Differentiation	303
7.2.1	Higher Derivatives	307
7.2.2	Interpolation	308
7.3	IVP Methods Using Numerical Differentiation	309
7.3.1	The Five Steps	309
7.3.2	Additional Difference Methods	316
7.3.3	Error and Convergence	319
7.3.4	Computing the Solution	320
7.4	IVP Methods Using Numerical Integration	322
7.5	Runge–Kutta Methods	325
7.5.1	RK2	325
7.5.2	RK4	328
7.5.3	Stability	331
7.5.4	RK-n	331
7.6	Solving Systems of IVPs	333
7.6.1	Examples	333
7.6.2	Simple Approach	334
7.6.3	Component Approach and Symplectic Methods	337
7.7	Some Additional Questions and Ideas	340
7.7.1	RK4 Order Conditions	343
<b>8</b>	<b>Optimization: Regression</b>	<b>357</b>
8.1	Model Function	358
8.2	Error Function	359
8.2.1	Vertical Distance	359
8.3	Linear Least Squares	360
8.3.1	Two Parameters	361
8.3.2	General Case	363
8.3.3	QR Approach	366
8.3.4	Moore–Penrose Pseudo-Inverse	368
8.3.5	Overdetermined System	369
8.3.6	Regularization	371
8.3.7	Over-fitting the Data	372
8.4	QR Factorization	374
8.4.1	Finding $\mathbf{Q}$ and $\mathbf{R}$	375
8.4.2	Accuracy	381
8.4.3	Parting Comments	382

8.5	Other Error Functions .....	382
8.6	Nonlinear Regression.....	385
8.7	Fitting IVPs to Data .....	389
8.7.1	Logistic Equation .....	389
8.7.2	FitzHugh-Nagumo Equations .....	392
8.7.3	Parting Comments .....	393
<b>9</b>	<b>Optimization: Descent Methods .....</b>	<b>409</b>
9.1	Introduction .....	409
9.2	Descent Methods .....	411
9.2.1	Descent Directions.....	412
9.3	Solving Linear Systems.....	413
9.3.1	Basic Descent Algorithm for $\mathbf{Ax} = \mathbf{b}$ .....	414
9.3.2	Method of Steepest Descents for $\mathbf{Ax} = \mathbf{b}$ .....	416
9.3.3	Conjugate Gradient Method for $\mathbf{Ax} = \mathbf{b}$ .....	418
9.3.4	Large Sparse Matrices .....	423
9.3.5	Rate of Convergence .....	426
9.3.6	Preconditioning .....	428
9.3.7	Parting Comments .....	433
9.4	General Nonlinear Problem .....	434
9.4.1	Descent Direction .....	434
9.4.2	Line Search Problem.....	436
9.4.3	Examples .....	439
9.5	Levenberg-Marquardt Method.....	441
9.5.1	Parting Comments .....	447
9.6	Minimization Without Differentiation .....	447
9.7	Variational Problems.....	451
9.7.1	Example: Minimum Potential Energy .....	451
9.7.2	Example: Brachistochrone Problem .....	453
9.7.3	Parting Comments .....	456
<b>10</b>	<b>Data Analysis .....</b>	<b>473</b>
10.1	Introduction .....	473
10.2	Principal Component Analysis .....	473
10.2.1	Example: Word Length .....	473
10.2.2	Derivation of Principal Component Approximation ...	478
10.2.3	Summary of the PCA .....	482
10.2.4	Scaling Factors .....	485
10.2.5	Geometry and Data Approximation.....	486
10.2.6	Application: Crime Data .....	487
10.2.7	Practical Issues .....	491
10.2.8	Parting Comments .....	493
10.3	Independent Component Analysis .....	493
10.3.1	Derivation of the Method.....	495
10.3.2	Reduced Problem .....	497

10.3.3 Contrast Function .....	498
10.3.4 Summary of ICA .....	503
10.3.5 Application: Image Steganography .....	504
10.4 Data Based Modeling .....	506
10.4.1 Dynamic Modes .....	510
10.4.2 Example: COVID-19 .....	513
10.4.3 Propagation Paths .....	515
10.4.4 Parting Comments .....	516
Exercises .....	517
<b>A Taylor's Theorem .....</b>	<b>527</b>
A.1 Useful Examples for $x$ Near Zero .....	529
A.2 Order Symbol and Truncation Error .....	530
<b>B Vector and Matrix Summary .....</b>	<b>533</b>
<b>C Answers .....</b>	<b>537</b>
<b>Solutions .....</b>	<b>545</b>
<b>References .....</b>	<b>547</b>
<b>Index .....</b>	<b>559</b>