

Preface to the Third Edition (2007)

“I was just going to say, when I was interrupted...” begins Oliver Wendell Holmes in the second series of his famous essays, *The Autocrat of the Breakfast Table*. The interruption referred to was a gap of 25 years. In our case, as the autocrats of *Numerical Recipes*, the gap between our second and third editions has been “only” 15 years. Scientific computing has changed enormously in that time.

The first edition of *Numerical Recipes* was roughly coincident with the first commercial success of the personal computer. The second edition came at about the time that the Internet, as we know it today, was created. Now, as we launch the third edition, the practice of science and engineering, and thus scientific computing, has been profoundly altered by the mature Internet and Web. It is no longer difficult to find *somebody’s* algorithm, and usually free code, for almost any conceivable scientific application. The critical questions have instead become, “How does it work?” and “Is it any good?” Correspondingly, the second edition of *Numerical Recipes* has come to be valued more and more for its text explanations, concise mathematical derivations, critical judgments, and advice, and less for its code implementations per se.

Recognizing the change, we have expanded and improved the text in many places in this edition and added many completely new sections. We seriously considered leaving the code out entirely, or making it available only on the Web. However, in the end, we decided that without code, it wouldn’t be *Numerical Recipes*. That is, without code you, the reader, could never know whether our advice was in fact honest, implementable, and practical. Many discussions of algorithms in the literature and on the Web omit crucial details that can only be uncovered by actually coding (our job) or reading compilable code (your job). Also, we needed actual code to teach and illustrate the large number of lessons about object-oriented programming that are implicit and explicit in this edition.

Our wholehearted embrace of a style of object-oriented computing for scientific applications should be evident throughout this book. We say “*a* style,” because, contrary to the claims of various self-appointed experts, there can be no one rigid style of programming that serves all purposes, not even all scientific purposes. Our style is ecumenical. If a simple, global, C-style function will fill the need, then we use it. On the other hand, you will find us building some fairly complicated structures for something as complicated as, e.g., integrating ordinary differential equations. For more on the approach taken in this book, see §1.3 – §1.5.

In bringing the text up to date, we have luckily not had to bridge a full 15-year gap. Significant modernizations were incorporated into the second edition versions in Fortran 90 (1996)* and C++ (2002), in which, notably, the last vestiges of unit-based arrays were expunged in favor of C-style zero-based indexing. Only with this third edition, however, have we incorporated a substantial amount (several hundred pages!) of completely new material. Highlights include:

- a new chapter on classification and inference, including such topics as Gaussian mixture models, hidden Markov modeling, hierarchical clustering (phylogenetic trees), and support vector machines

*“Alas, poor Fortran 90! We knew him, Horatio: a programming language of infinite jest, of most excellent fancy: he hath borne us on his back a thousand times.”

- a new chapter on computational geometry, including topics like KD trees, quad- and octrees, Delaunay triangulation and applications, and many useful algorithms for lines, polygons, triangles, spheres, etc.
- many new statistical distributions, with pdfs, cdfs, and inverse cdfs
- an expanded treatment of ODEs, emphasizing recent advances, and with completely new routines
- much expanded sections on uniform random deviates and on deviates from many other statistical distributions
- an introduction to spectral and pseudospectral methods for PDEs
- interior point methods for linear programming
- more on sparse matrices
- interpolation on scattered data in multidimensions
- curve interpolation in multidimensions
- quadrature by variable transformation and adaptive quadrature
- more on Gaussian quadratures and orthogonal polynomials
- more on accelerating the convergence of series
- improved incomplete gamma and beta functions and new inverse functions
- improved spherical harmonics and fast spherical harmonic transforms
- generalized Fermi-Dirac integrals
- multivariate Gaussian deviates
- algorithms and implementations for hash memory functions
- incremental quantile estimation
- chi-square with small numbers of counts
- dynamic programming
- hard and soft error correction and Viterbi decoding
- eigensystem routines for real, nonsymmetric matrices
- multitaper methods for power spectral estimation
- wavelets on the interval
- information-theoretic properties of distributions
- Markov chain Monte Carlo
- Gaussian process regression and kriging
- stochastic simulation of chemical reaction networks
- code for plotting simple graphs from within programs

The *Numerical Recipes* Web site, www.nr.com, is one of the oldest active sites on the Internet, as evidenced by its two-letter domain name. We will continue to make the Web site useful to readers of this edition. Go there to find the latest bug reports, to purchase the machine-readable source code, or to participate in our readers' forum. With this third edition, we also plan to offer, by subscription, a completely electronic version of *Numerical Recipes* — accessible via the Web, downloadable, printable, and, unlike any paper version, always up to date with the latest corrections. Since the electronic version does not share the page limits of the print version, it will grow over time by the addition of completely new sections, available only electronically. This, we think, is the future of *Numerical Recipes* and perhaps of technical reference books generally. If it sounds interesting to you, look at <http://www.nr.com/electronic>.

This edition also incorporates some “user-friendly” typographical and stylistic improvements: Color is used for headings and to highlight executable code. For code, a label in the margin gives the name of the source file in the machine-readable distribution. Instead of printing repetitive `#include` statements, we provide a con-

venient Web tool at <http://www.nr.com/dependencies> that will generate exactly the statements you need for any combination of routines. Subsections are now numbered and referred to by number. References to journal articles now include, in most cases, the article title, as an aid to easy Web searching. Many references have been updated; but we have kept references to the grand old literature of classical numerical analysis when we think that books and articles deserve to be remembered.

Acknowledgments

Regrettably, over 15 years, we were not able to maintain a systematic record of the many dozens of colleagues and readers who have made important suggestions, pointed us to new material, corrected errors, and otherwise improved the *Numerical Recipes* enterprise. It is a tired cliché to say that “you know who you are.” Actually, in most cases, *we* know who you are, and we are grateful. But a list of names would be incomplete, and therefore offensive to those whose contributions are no less important than those listed. We apologize to both groups, those we might have listed and those we might have missed.

We prepared this book for publication on Windows and Linux machines, generally with Intel Pentium processors, using LaTeX in the TeTeX and MiKTeX implementations. Packages used include amsmath, amsfonts, txfonts, and graphicx, among others. Our principal development environments were Microsoft Visual Studio / Microsoft Visual C++ and GNU C++. We used the SourceJammer cross-platform source control system. Many tasks were automated with Perl scripts. We could not live without GNU Emacs. To all the developers: “You know who you are,” and we thank you.

Research by the authors on computational methods was supported in part by the U.S. National Science Foundation and the U.S. Department of Energy.