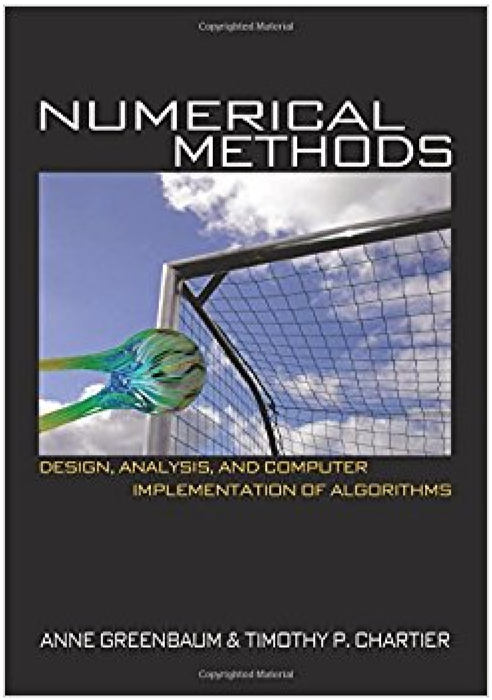
Numerical Methods: Design, Analysis, And Computer Implementation Of Algorithms



Synopsis

Numerical Methods provides a clear and concise exploration of standard numerical analysis topics, as well as nontraditional ones, including mathematical modeling, Monte Carlo methods, Markov chains, and fractals. Filled with appealing examples that will motivate students, the textbook considers modern application areas, such as information retrieval and animation, and classical topics from physics and engineering. Exercises use MATLAB and promote understanding of computational results. The book gives instructors the flexibility to emphasize different aspects--design, analysis, or computer implementation--of numerical algorithms, depending on the background and interests of students. Designed for upper-division undergraduates in mathematics or computer science classes, the textbook assumes that students have prior knowledge of linear algebra and calculus, although these topics are reviewed in the text. Short discussions of the history of numerical methods are interspersed throughout the chapters. The book also includes polynomial interpolation at Chebyshev points, use of the MATLAB package Chebfun, and a section on the fast Fourier transform. Supplementary materials are available online. Clear and concise exposition of standard numerical analysis topics Explores nontraditional topics, such as mathematical modeling and Monte Carlo methods Covers modern applications, including information retrieval and animation, and classical applications from physics and engineering Promotes understanding of computational results through MATLAB exercises Provides flexibility so instructors can emphasize mathematical or applied/computational aspects of numerical methods or a combination Includes recent results on polynomial interpolation at Chebyshev points and use of the MATLAB package Chebfun Short discussions of the history of numerical methods interspersed throughout Supplementary materials available online

Book Information

Hardcover: 464 pages

Publisher: Princeton University Press; 3.2.2012 edition (April 1, 2012)

Language: English

ISBN-10: 0691151229

ISBN-13: 978-0691151229

Product Dimensions: 7.2 x 1.2 x 10 inches

Shipping Weight: 2.5 pounds (View shipping rates and policies)

Average Customer Review: 3.9 out of 5 stars   5 customer reviews

Best Sellers Rank: #226,167 in Books (See Top 100 in Books)   #14 inÂ Â Books > Science & Math

> Mathematics > Matrices   #117 inÃ Â Books > Science & Math > Mathematics > Pure Mathematics > Algebra > Linear   #147 inÃ Â Books > Science & Math > Mathematics > Mathematical Analysis

Customer Reviews

"Distinguishing features are the inclusion of many recent applications of numerical methods and the extensive discussion of methods based on Chebyshev interpolation. This book would be suitable for use in courses aimed at advanced undergraduate students in mathematics, the sciences, and engineering."--Choice"An instructor could assemble several different one-semester courses using this book--numerical linear algebra and interpolation, or numerical solutions of differential equations--or perhaps a two-semester sequence. This is a charming book, well worth consideration for the next numerical analysis course."--William J. Satzer, MAA Focus"[Numerical Methods] is a very pleasant book, where the concepts involved are clearly explained. All chapters begin with motivating examples that give a precise idea of the methods developed. In addition, every chapter ends with an extensive collection of exercises, useful to understand the importance of the results. These are complemented by a series of exercises, designed to be performed with Matlab, useful to appreciate the behavior of the methods studied."--European Mathematical Society

"This is an excellent introduction to the exciting world of numerical analysis. Fulfilling the need for a modern textbook on numerical methods, this volume has a wealth of examples that will keep students interested in the material. The mathematics is completely rigorous and I applaud the authors for doing such a marvelous job."--Michele Benzi, Emory University"Filled with polished details and a plethora of examples and illustrations, this ambitious and substantial text touches every standard topic of numerical analysis. The authors have done a huge amount of work and produced a major textbook for this subject."--Lloyd N. Trefethen, University of Oxford

I am an undergraduate student in Applied Mathematics who just used this text in a course on Numerical Analysis in one of the last courses I'm taking before moving on to grad school for Computer Science.As a previous reviewer noted, this text really goes out of its way to motivate the reader with a bit of a firehose approach to introducing all the different ways in which this material can be applied to computing problems, from graphics processing to machining to airfoil simulation to web search. This field of study may have been developed centuries ago largely by astrophysicists, but the application goes way beyond that with the widespread availability of computers and their ability to implement algorithms that involve too many steps to complete by hand.The complete chapter list is1) Mathematical Modeling2) Basic Operations with MATLAB3) Monte Carlo Methods4) Solution of a Single Nonlinear Equation in One Unknown5) Floating-Point Arithmetic6) Conditioning of Problems; Stability of Algorithms7) Direct Methods for Solving Linear Systems and Least Squares Problems8) Polynomial and Piecewise Polynomial Interpolation9) Numerical Differentiation and Richardson Extrapolation10) Numerical Integration11) Numerical Solution of the Initial Value Problem for Ordinary Differential Equations12) More Numerical Linear Algebra: Eigenvalues and Iterative Methods for Solving Linear Systems13) Numerical Solution of Two-Point Boundary Value Problems14) Numerical Solution of Partial Differential EquationsAppendices:A) Review of Linear AlgebraB) Taylor's Theorem in MultidimensionsFirst off, the prerequisites review is not going to help you unless you already learned linear algebra and calculus (at least up to power series) in detail in the past. If you're purchasing this for a university course, they should be enforcing prerequisites anyway. If you're purchasing this for self-study, be aware that it's not teaching you much math. Finding polynomial roots, solving linear systems, solving least squares problems, differentiation, integration, and solving differential equations should be something you already know how to do. This text is simply showing you how to do it with a computer when an analytic solution is either not available or otherwise too difficult. Additionally, you'll probably be a little lost without a background in probability in the Monte Carlo chapter, but that chapter really does not tie in with the rest of the text and can be easily omitted (we covered it, but the professor didn't expect much and was clear in stating this is a non-traditional topic for a text in numerical analysis).The upside of this text is that it is extremely readable. Derivations, theorems, and proofs are integrated with narrative and are given a clear explanation. It came across a bit dense at first, but as the error analysis for each algorithm relies upon the order of the remainder term in a Taylor series expansion, the proofs should eventually click given the similarity from one to the next. For the most part, the algorithms themselves are not very complicated. Figuring out how to code some of the solvers for linear systems took me a few hours, though I think that had more to do with learning specific language features than anything about the algorithm itself. The harder part is the error analysis, and you really need to slow down, write it all down and figure out what's happening before you move on. There isn't much complexity analysis, and what little there is should not be challenging if you have any background in more general algorithmic analysis/discrete math.As a quick note for students who are not using MATLAB, beware some of the specific low-level features of the language you're using. Undergrad math departments like to use computer algebra systems, and these are very nice for symbolic computation, but they often don't work the same. We used Maple in my course, and by default, Maple uses software floating-point numbers and performs much of the computation in memory, not in the processor ALU, and this changes the error analysis as software floats don't adhere to the IEEE double-precision standard the text discusses and are not limited by machine precision. You can get identical results to those in the text, with identical error analysis, but only if you explicitly tell the software to use hardware floating-point representations. You may need to dig into the documentation to figure out how to do this.Downsides: This may or may not be a downside depending upon what you're looking for, but this is very much a university textbook. It's teaching you how to think like a numerical analyst. The code snippets are nice, but this is not a reference or a cookbook. Much of the code is fragments. You can copy and paste it straight into a MATLAB script and it won't run. You're not going to pick up this book and instantly learn how to do something. You're going to struggle to learn something. But you will learn. There's a fair amount (arguably too much) of MATLAB-specific discussion: for instance, how to use chebfun to find Chebyshev nodes when interpolating high-degree polynomials. You're on your own figuring out how to do this if you don't have MATLAB or some comparable numerical computing software. Note that Octave and Numpy have much of the same functionality and are free, so if you can't get student MATLAB, you can still follow along.Note that there is also no solution manual available yet and no answers to any of the exercises. This may or may not make a difference to you. If you're in a course being taught by a professor, the professor will grade and provide feedback and answer homework questions. If you're purchasing this for self-study, be aware. It's really easy if you have MATLAB or Maple or know how to use Numpy to use built-in functions to check for the true solution of a linear system or a least squares problem and then see if your code worked by comparison. If you don't have those tools or don't know how to use them, however, the text itself will not help you.The bottom line is I would not recommend this for self-study unless you already have a math, cs, or engineering background and simply never covered numerical analysis. You need to have the prerequisite background, which is at least a full year of calculus and a full term of linear algebra. Ideally, you've had calc III and some background in programming and algorithmic complexity analysis, but these are not completely necessary. The coverage of each topic is brief and you will not find it difficult if you have the background. You're not going to get very deep into how to find eigenvalues of large matrices. This book is simply the next step when you've already learned calculus and linear algebra to begin considering the engineering problems faced when you implement techniques on a computer and now have to worry about conditioning and stability and floating-point arithmetic. It gives you the tools to understand the sources of error and to properly weigh trade-offs using rigorous quantification.

I teach computers to do math, so-- disclaimer-- I'm on the applied, not pure math side of NA. I came across this text while compiling a Body of Knowledge entry on Spectral, Fourier and Chebyshev methods for IEEE and the International Association of Bodies of Knowledge (The 9bok dot org people who certify math BoKs). The usual "track" for advanced undergrads is Calc up to PDE's, some linear algebra, a little computer arithmetic (and maybe some of my field, Computer Algebra), then on to Engineering or Physics.Along the way, most of us will touch Numerical Analysis. There are two distinct sides to NA-- pure, as a way of defining formal proofs with "results" as much as methods, and applied-- solving problems, especially using algorithms, via close approximation, guessing, brute force, iteration, and other "cheats." The problem with many of the classic NA texts is that "applied" usually means, you guessed it, physics and engineering. Today, however, NA is as much at home with digital artists, game programmers creating physics engines, animators, Maya programmers, etc. as with physicists!You'd think with that going on, there would be some rocking texts that are also fun. Not the case. Sadly, most of the "better" (read understandable) texts in NA date back to the late 1980s, when there was no internet (there were 50 websites in 1992 when Clinton took office). In fact, this author's book on Iterative Linear methods dates back to 1987, and John Boyd's classic on Fourier Spectrals to 1989.This text changes a lot of that! The authors use a LOT more current examples you're likely to find in many other fields, from protein folding to NASCAR. Who uses computers to "guess" at difficult PDE solutions other than astrophysicists? Try Neurologists modeling cognition as Dynamic Systems! Yes, the applications today are way beyond what they were in 1987, and we finally have an NA text that covers not only the basics, but MANY cutting edge areas-- like fractals-- that weren't even taken seriously back then.To be fair, some of the examples just give a "taste" of the field, and were filled in by experts, but not really used in the text, and apparently not really understood by the authors. For example, Dan Goldman was tapped to give some fun examples of collision detection for Yoda in Star Wars, but if you look up Inverse Kinematics or Kinematics in the Index, there is no mention. Lorentz transforms and dynamic analysis are not really covered, and when their NA engines are mentioned (Newton's Method, for example), they are in the context of Julia sets and fractals, not Kinematics.If you haven't taken linear algebra and aren't very familiar with matrices, there is "some" review here, but not enough to make this text fun and painless. You really need to brush up on LA before tackling this. Look at it this way: you can think of many NA techniques as you would visual basic behind Excel. Your computer is using a lot of "spreadsheet" type crunching, only of functions, to "guesstimate" things like zeros/roots, parameters, etc. So what are these "spreadsheets" called by mathematicians? Right, Linear Algebra (on roids). Back in ancient times (the 90's) NA was considered the playground of mathematicians, engineers and physicists. Today, game programmers, animators, digital artists and even programmers like yours truly need to understand it to know what's going on underneath the calculations-- namely, crunching, right down to the stacks and registers.Another group that will like this text are the embedded circuit folks-- instead of nail biting about on or off chip memory limits, many of the newer memory limit "work arounds" are in NA functions, algorithms and shortcuts. Sure, we'll eventually have to SOLVE the memory issues, but for now, the real world IS about working around with "close enough" solutions. You won't find any of these applications in most NA texts-- the present work is a gem, and unique in being up to date on many NOW applications, including several beyond the traditional physics and engineering examples. Oh, and yes, you do learn analysis here too, including the proofs and pure math sides if your track is math. I'm not qualified to opine in that "pure proof" track, but if you're on the applied side of NA, and want to go deeper, you'll love this text.History note for a few emailers: Thanks for reminding me that 1987 is "recent" compared to many NA techniques that adapt Euler to algorithmic form-- by "recent" I also mean that these authors use examples like web surfing and Google's (secret sauce) analytics. I'm talking examples too, not just the fact --which I'll gladly grant-- that much of NA stands on the shoulders of giants going back to the 1700s. I challenge anyone to show me an NA text that is this relevant to today's applications, however! If you're a student, you also won't feel like you're being forced to study stuff that will have no relevance to your future. If you're a prof-- don't you want to orient your students via examples that are being used right now? They WILL thank you!

Excellent book! It's full of very useful and practical information.

Smooth, fast transaction. Thanks!

Speaking as an undergraduate, I simply hated this book. It was very confusing to read because the material would jump all over the place. I had to use notes that were uploaded by other schools to get through my class.

Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics Biological Modeling and Simulation: A Survey of Practical Models, Algorithms, and Numerical Methods (Computational Molecular Biology) Computer Algorithms: Introduction to Design and Analysis (3rd Edition) Analog

Methods for Computer-Aided Circuit Analysis and Diagnosis (Electrical and Computer Engineering) Computer Vision: Algorithms and Applications (Texts in Computer Science) Analytics: Business Intelligence, Algorithms and Statistical Analysis (Predictive Analytics, Data Visualization, Data Analytics, Business Analytics, Decision Analysis, Big Data, Statistical Analysis) Analytics: Data Science, Data Analysis and Predictive Analytics for Business (Algorithms, Business Intelligence, Statistical Analysis, Decision Analysis, Business Analytics, Data Mining, Big Data) Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods (Applied and Numerical Harmonic Analysis) Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications (Applied and Numerical Harmonic Analysis) Computer Organization and Design MIPS Edition, Fifth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) Computer Organization and Design, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) 1st Grade Computer Basics : The Computer and Its Parts: Computers for Kids First Grade (Children's Computer Hardware Books) Accuracy and Stability of Numerical Algorithms Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms Bundle of Algorithms in C++, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms (3rd Edition) (Pts. 1-5) Practical Algorithms in Pediatric Hematology and Oncology: (Practical Algorithms in Pediatrics. Series Editor: Z. Hochberg) Practical Algorithms in Pediatric Nephrology: (Practical Algorithms in Pediatrics. Series Editor: Z. Hochberg) Practical Algorithms in Pediatric Gastroenterology: (Practical Algorithms in Pediatrics. Series Editor: Z. Hochberg) Practical Algorithms in Pediatric Endocrinology: (Practical Algorithms in Pediatrics. Series Editor: Z. Hochberg)