Fundamentals of Matrix Computations

Second Edition

David S. Watkins

A Wiley-Interscience Publication
JOHN WILEY & SONS, INC.

New York / Chichester / Weinheim / Brisbane / Singapore / Toronto

Contents

Preface

This book was written for advanced undergraduates, graduate students, and mature scientists in mathematics, computer science, engineering, and all disciplines in which numerical methods are used. At the heart of most scientific computer codes lie matrix computations, so it is important to understand how to perform such computations efficiently and accurately. This book meets that need by providing a detailed introduction to the fundamental ideas of numerical linear algebra.

The prerequisites are a first course in linear algebra and some experience with computer programming. For the understanding of some of the examples, especially in the second half of the book, the student will find it helpful to have had a first course in differential equations.

There are several other excellent books on this subject, including those by Demmel [15], Golub and Van Loan [33], and Trefethen and Bau [71]. Students who are new to this material often find those books quite difficult to read. The purpose of this book is to provide a gentler, more gradual introduction to the subject that is nevertheless mathematically solid. The strong positive student response to the first edition has assured me that my first attempt was successful and encouraged me to produce this updated and extended edition.

The first edition was aimed mainly at the undergraduate level. As it turned out, the book also found a great deal of use as a graduate text. I have therefore added new material to make the book more attractive at the graduate level. These additions are detailed below. However, the text remains suitable for undergraduate use, as the elementary material has been kept largely intact, and more elementary exercises have been added. The instructor can control the level of difficulty by deciding which sections to cover and how far to push into each section. Numerous advanced topics are developed in exercises at the ends of the sections.

The book contains many exercises, ranging from easy to moderately difficult. Some are interspersed with the textual material and others are collected at the end of each section. Those that are interspersed with the text are meant to be worked immediately by the reader. This is my way of getting students actively involved in the learning process. In order to get something out, you have to put something in. Many of the exercises at the ends of sections are lengthy and may appear intimidating at first. However, the persistent student will find that s/he can make it through them with the help of the ample hints and advice that are given. I encourage every student to work as many of the exercises as possible.

Numbering Scheme

Nearly all numbered items in this book, including theorems, lemmas, numbered equations, examples, and exercises, share a single numbering scheme. For example, the first numbered item in Section 1.3 is Theorem 1.3.1. The next two numbered items are displayed equations, which are numbered (1.3.2) and (1.3.3), respectively. These are followed by the first exercise of the section, which bears the number 1.3.4. Thus each item has a unique number: the only item in the book that has the number 1.3.4 is Exercise 1.3.4. Although this scheme is unusual, I believe that most readers will find it perfectly natural, once they have gotten used to it. Its big advantage is that it makes things easy to find: The reader who has located Exercises 1.4.15 and 1.4.25 but is looking for Example 1.4.20, knows for sure that this example lies somewhere between the two exercises.

There are a couple of exceptions to the scheme. For technical reasons related to the type setting, tables and figures (the so-called *floating bodies*) are numbered separately by chapter. For example, the third figure of Chapter 1 is Figure 1.3.

New Features of the Second Edition

Use of MATLAB

By now MATLAB[1] is firmly established as the most widely used vehicle for teaching matrix computations. MATLAB is an easy to use, very high-level language that allows the student to perform much more elaborate computational experiments than before. MATLAB is also widely used in industry. I have therefore added many examples and exercises that make use of MATLAB. This book is not, however, an introduction to MATLAB, nor is it a MATLAB manual. For those purposes there are other books available, for example, the *MATLAB Guide* by Higham and Higham [40].

[1] MATLAB is a registered trademark of the MathWorks, Inc. (`http://www.mathworks.com`)

However, MATLAB's extensive help facilities are good enough that the reader may feel no need for a supplementary text. In an effort to make it easier for the student to use MATLAB with this book, I have included an index of MATLAB terms, separate from the ordinary index.

I used to make my students write and debug their own Fortran programs. I have left the Fortran exercises from the first edition largely intact. I hope a few students will choose to work through some of these worthwhile projects.

More Applications

In order to help the student better understand the importance of the subject matter of this book, I have included more examples and exercises on applications (solved using MATLAB), mostly at the beginnings of chapters. I have chosen very simple applications: electrical circuits, mass-spring systems, simple partial differential equations. In my opinion the simplest examples are the ones from which we can learn the most.

Earlier Introduction of the Singular Value Decomposition (SVD)

The SVD is one of the most important tools in numerical linear algebra. In the first edition it was placed in the final chapter of the book, because it is impossible to discuss methods for computing the SVD until after eigenvalue problems have been discussed. I have since decided that the SVD needs to be introduced sooner, so that the student can find out earlier about its properties and uses. With the help of MATLAB, the student can experiment with the SVD without knowing anything about how it is computed. Therefore I have added a brief chapter on the SVD in the middle of the book.

New Material on Iterative Methods

The biggest addition to the book is a chapter on iterative methods for solving large, sparse systems of linear equations. The main focus of the chapter is the powerful conjugate-gradient method for solving symmetric, positive definite systems. However, the classical iterations are also discussed, and so are preconditioners. Krylov subspace methods for solving indefinite and nonsymmetric problems are surveyed briefly.

There are also two new sections on methods for solving large, sparse eigenvalue problems. The discussion includes the popular implicitly-restarted Arnoldi and Jacobi-Davidson methods.

I hope that these additions in particular will make the book more attractive as a graduate text.

Other New Features

To make the book more versatile, a number of other topics have been added, including:

- a backward error analysis of Gaussian elimination, including a discussion of the modern componentwise error analysis.
- a discussion of reorthogonalization, a practical means of obtaining numerically orthonormal vectors.
- a discussion of how to update the QR decomposition when a row or column is added to or deleted from the data matrix, as happens in signal processing and data analysis applications.
- a section introducing new methods for the symmetric eigenvalue problem that have been developed since the first edition was published.

A few topics have been deleted on the grounds that they are either obsolete or too specialized. I have also taken the opportunity to correct several vexing errors from the first edition. I can only hope that I have not introduced too many new ones.

DAVID S. WATKINS

Pullman, Washington
January, 2002

Acknowledgments

I am greatly indebted to the authors of some of the early works in this field. These include A. S. Householder [43], J. H. Wilkinson [81], G. E. Forsythe and C. B. Moler [24], G. W. Stewart [67], C. L. Lawson and R. J. Hanson [48], B. N. Parlett [54], A. George and J. W. Liu [30], as well as the authors of the Handbook [83], the EISPACK Guide [64], and the LINPACK Users' Guide [18]. All of them influenced me profoundly. By the way, every one of these books is still worth reading today. Special thanks go to Cleve Moler for inventing MATLAB, which has changed everything.

Most of the first edition was written while I was on leave, at the University of Bielefeld, Germany. I am pleased to thank once again my host and longtime friend, Ludwig Elsner. During that stay I received financial support from the Fulbright commission. A big chunk of the second edition was also written in Germany, at the Technical University of Chemnitz. I thank my host (and another longtime friend), Volker Mehrmann. On that visit I received financial support from Sonderforschungsbereich 393, TU Chemnitz. I am also indebted to my home institution, Washington State University, for its support of my work on both editions.

I thank once again professors Dale Olesky, Kemble Yates, and Tjalling Ypma, who class-tested a preliminary version of the first edition. Since publication of the first edition, numerous people have sent me corrections, feedback, and comments. These include A. Cline, L. Dieci, E. Jessup, D. Koya, D. D. Olesky, B. N. Parlett, A. C. Raines, A. Witt, and K. Wright. Finally, I thank the many students who have helped me learn how to teach this material over the years.

D. S. W.