Preface

This book is intended for an advanced undergraduate or a first-year graduate course in numerical linear algebra, a very important topic for engineers and scientists. Many of the numerical methods used to solve engineering and science problems have linear algebra as an important component. Examples include spline interpolation, estimation using least squares, and the solution of ordinary and partial differential equations. It has been said that, next to calculus, linear algebra is the most important component in engineering problem solving. In computer science, linear algebra is a critical as well. The Google matrix is an example, as is computer graphics where matrices are used for rotation, projection, rescaling, and translation. Applications to engineering and science are provided throughout the book.

Two important problems in a customary applied linear algebra course are the solution of general linear algebraic systems $Ax = b$, where A is an $m \times n$ matrix, and the computation of eigenvalues and their associated eigenvectors. If the system is square ($m = n$) and nonsingular, the student is taught how to find a solution to $Ax = b$ using Cramer's Rule, Gaussian elimination and multiplication by the inverse. In many areas of application, such as statistics and signal processing, A is square and singular or $m \neq n$. In these situations, the transformation to reduced row echelon form produces no solution or infinitely many, and this is just fine in a theoretical sense, but is not helpful for obtaining a useful solution. Eigenvalues are often discussed late in the course, and the student learns to compute eigenvalues by finding the roots of the characteristic polynomial, never done in practice.

A study of numerical linear algebra is different from a study of linear algebra. The problem is that many of the theoretical linear algebra methods are not practical for use with a computer. To be used on a computer, an algorithm, a method for solving a problem step by step in a finite amount of time, must be developed that deals with the advantages and problems of using a computer. Any such algorithm must be efficient and not use too much computer memory. For instance, Cramer's Rule is not practical for matrices of size 4×4 or greater, since it performs far to many operations. Since a digital computer performs arithmetic in binary with a fixed number of digits, errors occur when entering data and performing computations. For instance, 1/3 cannot be represented exactly in binary, and its binary representation must be approximated. In addition, computation using the operations of addition, subtraction, multiplication, and division rarely can be done exactly, resulting in errors. An algorithm must behave properly in the presence of these inevitable errors; in other words, small errors during computation should produce small errors in the output. For example, the use of Gaussian elimination to solve an $n \times n$ linear system should use a method known as partial pivoting to control errors. When the matrix A is $m \times n$, $m \neq n$, a solution must be obtained in the sense of least-squares, and the efficient and implementation of least-squares presents challenges. The eigenvalue problem is of primary importance in engineering and science. In practice, eigenvalues are not found by finding the roots of a polynomial, since polynomial root finding is very prone to error. Algorithms have been developed for accurate solution of the eigenvalue problem on a computer.

In the book, algorithms are stated using pseudocode, and MATLAB is the vehicle used for algorithm implementation. MATLAB does a superb job of dealing with numeric computation and is used in most engineering programs. Accompanying the text is a library of MATLAB functions and programs, named NLALIB, that implements most of the algorithms discussed in the book. Many examples in the book include computations using MATLAB, as do many exercises. In some cases, a problem will require the student to write a function or program using the MATLAB programming language. If the student is not familiar with MATLAB or needs a refresher, the MathWorks Web site www.mathworks.com provides access to tutorials. There are also many free online tutorials.

If the reader does not have access to MATLAB, it is possible to use GNU Octave, a system primarily intended for numerical computations. The Octave language is quite similar to MATLAB so that most programs are easily portable.

This book is novel, in that there is no assumption the student has had a course in linear algebra. Engineering students who have completed the usual mathematics sequence, including ordinary differential equations, are well prepared. The prerequisites for a computer science student should include at least two semesters of calculus and a course in discrete mathematics. Chapters 1-6 supply an introduction to the the basics of linear algebra. A thorough knowledge of these chapters prepares the student very well for the remainder of the book. If the student has had a course in applied or theoretical linear algebra, these chapters can be used for a quick review.

Throughout the book, proofs are provided for most of the major results. In proofs, the author has made an effort to be clear, to the point of including more detail than normally provided in similar books.It is left to the instructor to determine how much emphasis should be given to the proofs.

The exercises include routine pencil and paper computations. Exercises of this type force the student to better understand the workings of an algorithm. There are some exercises involving proofs. Hints are provided if a proof will be challenging for most students. In the problems for each chapter, there are exercises to be done using MATLAB.

TOPICS

Chapters 1-6 provide coverage of applied linear algebra sufficient for reading the remainder of the book.

Chapter 1: Matrices

The chapter introduces matrix arithmetic and the very important topic of linear transformations. Rotation matrices provide an interesting and useful example of linear transformations. After discussing matrix powers, the concept of the matrix inverse and transpose concludes the chapter.

Chapter 2: Linear Equations

This chapter introduces Gaussian elimination for the solution of linear systems $Ax = b$ and for the computation of the matrix inverse. The chapter also introduces the relationship between the matrix inverse and the solution to a linear homogeneous equation. Two applications involving a truss and an electrical circuit conclude the chapter.

Chapter 3: Subspaces

This chapter is, by its very nature, somewhat abstract. It introduces the concepts of subspaces, linear independence, basis, matrix rank, range, and null space. Although the chapter may challenge some readers, the concepts are essential for understanding many topics in the book, and it should be covered thoroughly.

Chapter 4: Determinants

Although the determinant is rarely computed in practice, it is often used in proofs of important results. The chapter introduces the determinant and its computation using expansion by minors and by row elimination. The chapter ends with an interesting application of the determinant to text encryption.

Chapter 5: Eigenvalues and Eigenvectors

This is a very important chapter, and its results are used throughout the book. After defining the eigenvalue and an associated eigenvector, the chapter develops some of their most important properties, including their use in matrix diagonalization. The chapter concludes with an application to the solution of systems of ordinary differential equations and the problem of ranking items using eigenvectors.

Chapter 6: Orthogonal Vectors and Matrices

This chapter introduces the inner product and its association with orthogonal matrices. Orthogonal matrices play an extremely important role in matrix factorization. The L^2 inner product of functions is briefly introduced to emphasize the general concept of an inner product.

Chapter 7: Vector and Matrix Norms

The study of numerical linear algebra begins with this chapter. The analysis of methods in numerical linear algebra relies heavily on the concept of vector and matrix norms. This chapter develops the 2-norm, the 1-norm, and the infinity norm for vectors. A development of matrix norms follows, the most important being matrix norms associated with a vector norm, called subordinate norms. The infinity and 1-norms are easy to compute, but the connection between their computation and the mathematical definition of the a matrix norm is somewhat complex. A MATLAB program motivates the process for the computation of the infinity norm, and the chapter contains a complete proof verifying the algorithm for computing the infinity norm. The 2-norm is the most useful matrix norm and by far the most difficult to compute. After motivating the computation process with a MATLAB program, the chapter provides a proof that the 2-norm is the square root of the largest singular value of the matrix and develops properties of the matrix 2-norm.

Chapter 8: Floating Point Arithmetic

The chapter presents the representation of integer and floating point data in a computer, discusses the concepts of overflow and underflow, and explains why roundoff errors occur that cannot be avoided. There is a careful discussion concerning the concepts of absolute and relative error measurement and why relative error is normally used. The chapter presents a mathematical analysis of floating point errors for addition and states results for other operations. The chapter concludes with a discussion of situations where a careful choice of algorithm can minimize errors. This chapter is critical for understanding the remaining chapters. The only content that can be reasonably omitted is the mathematical discussion of floating point errors.

Chapter 9: Algorithms

The algorithms in the book are presented using pseudocode, and the pseudocode is quite complete. It is intended that in most cases the conversion between pseudocode and MATLAB should not be difficult. The chapter introduces the concept of algorithm efficiency by computing the the number of floating point operations, called the flop count, or representing it using big-O notation. The presentation of algorithms for matrix multiplication, the solution to upper and lower triangular systems, and the Thomas algorithm for the solution of a tridiagonal system are the primary examples. Included is a brief discussion of block matrices and basic block matrix operations.

Chapter 10: Conditioning of Problems and the Stability of Algorithms

The chapter introduces the concept of stability and the conditioning. An algorithm is unstable if small changes in the data can cause large changes in the result of the computation. An algorithm may be stable, but the data supplied to the algorithm can be ill-conditioned. For instance, some matrices are very sensitive to errors during Gaussian elimination. After discussing examples and introducing some elementary perturbation analysis using backward and forward error, the chapter develops the condition number of a matrix and its properties. The condition number of a matrix plays an important role as we develop algorithms in the remainder of the book. This material is at the heart of numerical linear algebra and should be covered at least intuitively. There are a number of problems involving numerical experiments, and some of these should be done in order to appreciate the issues involved.

Chapter 11: Gaussian Elimination and the LU Factorization

This chapter introduces the LU decomposition of a square matrix. The LU decomposition uses Gaussian elimination, but is not a satisfactory algorithm without using partial pivoting to minimize errors. The LU decomposition properly computed can be used to solve systems of the form $Ax_i = b_i$, $1 \le i \le k$. The somewhat expensive Gaussian elimination algorithm need be used only once. After its computation, many solutions $\{x_i\}$ are quickly found using forward and back substitution.

Chapter 12: Linear Systems Applications

Four applications that involve linear systems comprise this chapter. A discussion of Fourier series introduces the concept of an infinite dimensional vector space and provides an application for the L^2 inner product introduced in Chapter 6. A second application involves finite difference approximations for the heat equation. Finite difference techniques are important when approximating the solution to boundary value problems for ordinary and partial differential equations. Chapter 16 discusses least-squares problems. As a tune-up for this chapter, the third application develops approximation by polynomial least-squares. The last application is a discussion of cubic spline interpolation. Using this process, a series of cubic polynomials are fitted between each pair of data points over an interval $a \le x \le b$, with the requirement that the curve obtained be twice differentiable. These cubic splines can then be used to very accurately estimate the data at other points in the interval. The computation of cubic splines involves the solution of a tridiagonal system of equations, and the Thomas algorithm presented in Chapter 9 works very well.

Chapter 13: Important Special Systems

Numerical linear algebra is all about computing solutions to problems accurately and efficiently. As a result, algorithms must be developed that take advantage of a special structure or properties of a matrix. This chapter discusses the factorization of a tridiagonal matrix and the Cholesky factorization of a symmetric positive definite matrix. In both cases, the matrix factorization leads to more efficient means of solving a linear system having a coefficient matrix of one of these types.

Chapter 14: Gram-Schmidt Orthonormalization

The Gram-Schmidt algorithm for computing an orthonormal basis is time-honored and important. It becomes critical in the development of algorithms such as the singular value and Arnoldi decompositions. The chapter carefully develops the QR decomposition using Gram-Schmidt. Although the decomposition is not normally done this way, it serves to demonstrate that this extremely important tool exists. As a result, the MATLAB algorithm qr can be used with some understanding until efficient methods for the QR decomposition are explained.

Chapter 15: The Singular Value Decomposition

The singular value decomposition (SVD) is perhaps the most important result in numerical linear algebra. Its uses are many, including providing a method for estimating matrix rank and the solution of least-squares problems. This chapter proves the SVD theorem and provides applications. Perhaps the most interesting application is the use of the SVD in image compression. Practical algorithms for the computation of the SVD are complex, and are left to Chapter 23.

Chapter 16: Least Squares Problems

Approximation using least-squares has important applications in statistics and many other areas. For instance, data collected by sensor networks is often analyzed using least-squares in order to approximate events taking place. Least-squares problems arise when the data requires the solution to an $m \times n$ system $Ax = b$, where $m \neq n$. Normally, there is no solution $\bar{x}$ such that $A\bar{x} = b$, or there are infinitely many solutions, so we seek a solution that minimizes the Euclidean norm of $Ax - b$. Least-squares provides an excellent application for the QR factorization and the SVD.

Chapter 17: Implementing the QR Factorization

The QR factorization using the Gram-Schmidt process was developed in Chapter 14. This chapter presents two other approaches to the factorization, the use of Givens rotations and Householder reflections. In each case, the algorithm is more stable than Gram-Schmidt. Also, we will have occasion to use Givens rotations and Householder reflections for other purposes, such as the computation of eigenvalues. If a detailed presentation is not required, these ideas have a nice geometrical interpretation.

Chapter 18: The Algebraic Eigenvalue Problem

The applications of the eigenvalue problem are vast. The chapter begins by presenting three applications, a problem in vibration and resonance, the Leslie model in population biology, and the buckling of a column. The accurate computation of eigenvalues and their associated eigenvectors is difficult. The power and inverse power methods are developed for computing the largest and smallest eigenvalues of a matrix. These methods are important but have limited use. The chapter discusses the QR iteration for the computation of all the eigenvalues and their associated eigenvectors of a real matrix whose eigenvalues are distinct. The development is detailed and includes the use of the shifted Hessenberg QR iteration. The chapter also develops the computation of eigenvectors using the Hessenberg inverse iteration. The method used in most professional implementations is the implicit QR iteration, also known as the Francis iteration. The chapter develops the algorithm for the computation of both the real and complex eigenvalues of a real matrix.

Chapter 19: The Symmetric Eigenvalue Problem

If a matrix is symmetric, an algorithm can exploit its symmetry and compute eigenvalues faster and more accurately. Fortunately, many very important problems in engineering and science involve symmetric matrices. The chapter develops five methods for the computation of eigenvalues and their associated eigenvectors, the Jacobi method, the symmetric QR iteration method, the Francis algorithm, the bisection method, and the divide and conquer method.

Chapter 20: Basic Iterative Methods

Iterative methods are used for the solution of large, sparse, systems, since ordinary Gaussian elimination operations will destroy the sparse structure of the matrix. This chapter presents the classical Jacobi, Gauss-Seidel, and SOR methods, along with discussion of convergence. The chapter concludes with the application of iterative methods to the solution of the two-dimensional Poisson equation.

Chapter 21: Krylov Subspace Methods

This is a capstone chapter, and should be covered, at least in part, in any numerical linear algebra course. The conjugate gradient method (CG) for the solution of large, sparse symmetric positive definite systems is presented. This method is one of the jewels of numerical linear algebra and has revolutionized the solution of many very large problems. The presentation motivates the algorithm and provides mathematical details that explain why it works. The conjugate gradient method is a Krylov subspace method, although the book does not develop it using this approach. However, the next algorithm presented is the general minimum residual method (GMRES) for the iterative solution of large, sparse, general matrices, and it is approached as a Krylov subspace method. The Krylov subspace-based minimum residual (MINRES) method for the solution of large, sparse, symmetric, non-positive definite matrices is the last method presented. If a matrix is ill-conditioned, CG, GMRES, and MINRES do not perform well. The solution is to precondition the system before applying an iterative method. The chapter presents preconditioning techniques for CG and GMRES. After presenting a chart detailing approaches to large, sparse problems, the chapter concludes with another approach to the Poisson equation and a discussion of the biharmonic equation that is one of the most important equations in applied mechanics.

Chapter 22: Large Sparse Eigenvalue Problems

The chapter discusses the use of the Arnoldi and Lanczos processes to find a few eigenvalues of large, sparse matrices. Two approaches are discussed, explicit and implicit restarting. The mathematics behind the performance of these methods is beyond the scope of the text, but the algorithms are presented and MATLAB implementations provided. Various exercises test the methods and clearly demonstrate the challenge of this problem.

Chapter 23: Computing the Singular Value Decomposition

The chapter develops two methods for computing the SVD, the one-sided Jacobi method, and the Demmel and Kahan zero-shift QR downward sweep algorithm**.** Developing the two methods requires a knowledge of many results from earlier chapters.

Appendices A, B, and C

Appendix A provides a discussion of complex numbers so that a reader unfamiliar with the topic will be able to acquire the knowledge necessary when the book uses basic results from the theory of complex numbers. Appendix B presents a brief discussion of mathematical induction, and Appendix C presents an overview of Chebyshev polynomials. Although these polynomials are not used within any proof in the book, they are referenced in theorems whose proofs are provided by other sources.

INTENDED AUDIENCE

Numerical linear algebra is often a final chapter in a standard linear algebra text, and yet is of paramount importance for engineers and scientists. The book covers many of the most important topics in numerical linear algebra, but is not intended to be encyclopedic. However, there are many references to material not covered in the book. Also, it is the author's hope that the material is more accessible as a first course than existing books, and that the first six chapters provide material sufficient for the book to be used without a previous course in applied linear algebra. The book is also is very useful for self-study and can serve as a reference for engineers and scientists. It can also serve as an entry point to more advanced books, such as James Demmel's book [1] or the exhaustive presentation of the topic by Golub and Van Loan [2].

WAYS TO USE THE BOOK

The instructor will need to decide how much theory should be covered; namely, how much emphasis will be placed on understanding the proofs and doing problems involving proofs. If the students are not experienced with proofs, one approach is to explain methods and theorems as intuitively as possible, supporting the discussion with numerical examples in class, and having the students do numerous numerical exercises both in class and in assignments. For instance, using Jacobi rotations to compute the eigenvalues of a real symmetric matrix is easily explained using simple diagrams and running a MATLAB program included with the software distribution graphically demonstrates how the method performs a reduction to a diagonal matrix. This approach works well with engineering students who have little or no experience with theorems and proofs. They will learn how to solve problems, large and small, using the appropriate methods.

If the audience consists of students who are mathematics majors or who have significant mathematical training, then some proofs should be covered and assignments should include proofs. Some of these exercises include hints to get the student started. The author believes that for a student to stare at the hypothesis and conclusion only to give up in frustration makes no sense, when a simple hint will kick start the process.

Of course, the amount of material that the instructor can cover depends on the background of the students. Mathematics majors will likely have taken a theoretical or applied linear algebra course. After optionally reviewing the material in Chapters 1-6 the study of numerical linear algebra can begin. The following is a list of suggestions for various chapters that outlines material that can be omitted, covered lightly, or must be covered.

- In Chapter 7, proofs that justify methods for computing matrix norms can be omitted, but MATLAB programs that motivate the methods should be discussed.
- Chapter 8 is essential to an understanding of numerical linear algebra. It presents storage formats for integers and floating point numbers and shows why the finite precision arithmetic used by a computer leads to roundoff error. Some examples are provided that show how rearranging the order of computation can help to reduce error.
- Chapter 10 that discusses the stability and conditioning of algorithms should be covered at least intuititely. There are numerous examples and problems in the book that illustrate the problems that can occur with floating point arithmetic.
- In Chapter 11, the LU decomposition must be presented, and the student should use it to solve a number of problems. If desired, the use of elementary row matrices to prove why the LU decomposition works can be omitted. It is very important the student understand that multiple systems can be solved with only one LU decomposition. The efficiency of many algorithms depends on it.
- The instructor can choose among the applications in Chapter 12, rather than covering the entire chapter.
- In Chapter 13, factoring tridiagonal matrices can be safely omitted, but positive definite matrices and the Cholesky decomposition must be covered.
- The Gram-Schmidt orthogonalization method and its use in forming the QR decomposition is important and not particularly difficult, so it should be covered.
- Except for the proof of the SVD theorem, all of Chapter 15 should be presented. The use of the SVD for image compression excites students and is just plain fun.
- In Chapter 16, rank-deficient and underdetermined least-squares can be omitted, since the majority of applications involve full rank overdetermined systems.
- It is recommended that Chapter 17 concerning the computation of the QR decomposition using Givens rotations and Householder reflections be covered. These tools are needed later in the book when discussing the eigenvalue problem. Both of these methods can be explained intuitively, supported by MATLAB programs from NLALIB, so the instructor can omit many of the details if desired.
- Chapter 18 discusses the general algebraic eigenvalue problem, and should be covered in part. Certainly it is important to discuss the power and inverse power methods and the QR iteration with and without shifts and deflation. The Francis,

or implicit QR iteration, is used in practice with both single and double implicit shifts. The details are complex, but an overview can be presented, followed by numerical experiments.

- The Spectral Theorem is used throughout the book, and its proof in Chapter 19 can be omitted with no harm. The Jacobi method for computing the eigenvalues and eigenvectors of a symmetric matrix can be covered thoroughly or intuitively. There are a number of programming and mathematical issues involved, but the idea is quite simple, and an intuitive explanation will suffice. Certainly the symmetric QR iteration method should be covered. If the Francis algorithm was covered in Chapter 18, it makes sense to present the single shift Francis algorithm. The bisection method is interesting and not difficult, so covering it is a good option. The chapter concludes with the complex divide-and-conquer method, and it is optional. NLALIB contains a C implementation of the algorithm using the MATLAB MEX interface, and it might be interesting demonstrate the algorithm's performance on a large, dense, symmetric matrix.
- The author feels that some coverage of iterative methods is very important since many engineering and science students will deal with projects that involve large, sparse matrices. The classical material on the Jacobi, Gauss-Seidel, and SOR iterations in Chapter 20 can be covered quickly by not presenting convergence theorems.
- The conjugate gradient method (CG) in Chapter 21 should be introduced and the student should gain experience using it and the preconditioned CG to solve large systems. The approach to its development is through the method of steepest descent. That algorithm is simple and can be supported by geometrical arguments. CG is an improvement of steepest descent, and the mathematical details can be skipped if desired. The application of Krylov subspace methods to develop the Arnoldi and Lanczos decompositions is somewhat technical, but the results are very important. At a minimum, the student should work some exercises that involve using NLALIB to execute some decompositions. It is then easy to see how these decompositions lead to the GMRES and MINRES methods. The software distribution contains a number of large, sparse matrices used in actual applications. These are used for examples and exercises in the book.
- Chapter 22 is very interesting both from a practical and theoretical standpoint. However, the material is challenging and can be left to more advanced courses. A possibility is using the chapter as an introduction to such books as Refs. [3–6].
- The SVD is used from Chapter 15 on, so the student is very familiar with its applications. Chapter 23 contains two methods for computing the SVD, and this material can be left to a subsequent course.

MATLAB LIBRARY

NLALIB is an essential supplement to the book. Figure 0.1 shows the structure of the library, in which almost all major algorithms are implemented as functions. As is customary, directory names are abbreviations; for instance, the subdirectory geneigs contains demonstration software for methods to compute the eigenvalues of a general, non-sparse, matrix. The book provides many examples of matrices from actual applications or matrices designed for testing purposes, and these matrices are included in NLALIB in MATLAB matrix format.

SUPPLEMENTS

At http://textbooks.elsevier.com/web/Manuals.aspx?isbn=9780123944351, the instructor will find supplements that include the solution to every problem in the book, laboratory exercises that can be used after lectures for more interactive

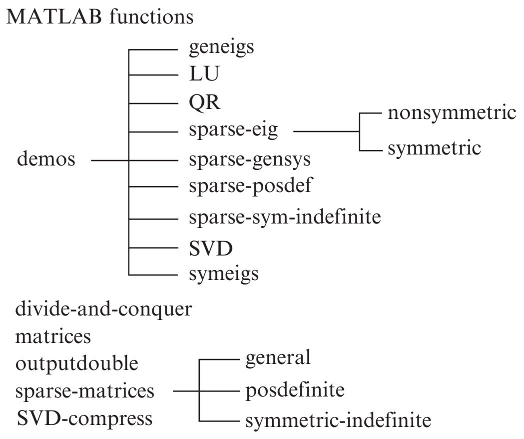

FIGURE 0.1 NLALIB hierarchy.

learning, and a complete set of PowerPoint slides. For students, Elsevier provides the Web site http://booksite.elsevier.com/9780123944435 that provides students with review questions and solutions. The author also provides the Web site http://ford-book.info that provides a summary of the book and links to associated Web sites.

ACKNOWLEDGMENTS

Although I have co-authored a number of books in computer science, this is my first mathematics text. I am indebted to Patrcia Osborn, the acquisitions editor at Elsevier who had faith in the book and ushered it through the acceptance phase. I would also like to thank the reviewers of the book, Jakub Kurzak, Ph.D., Research Director, Innovative Computing Laboratory, University of Tennessee, Rajan Bhatt, Ph.D., research engineer at the University of Iowa Center for Computer Aided Design, and Zhongshan Li, Ph.D., Professor and Graduate Director of Mathematics, Georgia State University. Their comments were helpful and encouraging. I also appreciate the efforts of the editorial project managers, Jill Cetel, Jessica Vaughn, and Paula Callaghan, who were invaluable in helping me through what is a difficult process. I also wish to thank Anusha Sambamoorthy, Project Manager, Book Production, Chennai, Elsevier, for her help with getting the book into print.

Dr Keith Matthews, Honorary Research Consultant, University of Queensland, gave me permission to use his online textbook, *Elementary Linear Algebra*, as a starting point for the material in the first six chapters. While I have made many changes to suit the needs of the book, his generosity saved me much time.

I thank the University of the Pacific, School of Engineering and Computer Science, Stockton, California, for providing resources and a sabbatical leave during the development of the book. I also appreciate comments made by engineering graduate students as I used the manuscript in an advanced computation course. Lastly, I am indebted to William Topp, Ph.D., with whom I have coauthored a number of books. He provided encouragement and consolation during the lengthy process of developing the text.