Matrix Analysis and Applied Linear Algebra

Carl D. Meyer

siam

Contents

You are today where your knowledge brought you;
you will be tomorrow where your knowledge takes you.
— Anonymous

Preface

Scaffolding

Reacting to criticism concerning the lack of motivation in his writings, Gauss remarked that architects of great cathedrals do not obscure the beauty of their work by leaving the scaffolding in place after the construction has been completed. His philosophy epitomized the formal presentation and teaching of mathematics throughout the nineteenth and twentieth centuries, and it is still commonly found in mid-to-upper-level mathematics textbooks. The inherent efficiency and natural beauty of mathematics are compromised by straying too far from Gauss's viewpoint. But, as with most things in life, appreciation is generally preceded by some understanding seasoned with a bit of maturity, and in mathematics this comes from seeing some of the scaffolding.

Purpose, Gap, and Challenge

The purpose of this text is to present the contemporary theory and applications of linear algebra to university students studying mathematics, engineering, or applied science at the postcalculus level. Because linear algebra is usually encountered between basic problem solving courses such as calculus or differential equations and more advanced courses that require students to cope with mathematical rigors, the challenge in teaching applied linear algebra is to expose some of the scaffolding while conditioning students to appreciate the utility and beauty of the subject. Effectively meeting this challenge and bridging the inherent gaps between basic and more advanced mathematics are primary goals of this book.

Rigor and Formalism

To reveal portions of the scaffolding, narratives, examples, and summaries are used in place of the formal definition–theorem–proof development. But while well-chosen examples can be more effective in promoting understanding than rigorous proofs, and while precious classroom minutes cannot be squandered on theoretical details, I believe that all scientifically oriented students should be exposed to some degree of mathematical thought, logic, and rigor. And if logic and rigor are to reside anywhere, they have to be in the textbook. So even when logic and rigor are not the primary thrust, they are always available. Formal definition–theorem–proof designations are not used, but definitions, theorems, and proofs nevertheless exist, and they become evident as a student's maturity increases. A significant effort is made to present a linear development that avoids forward references, circular arguments, and dependence on prior knowledge of the subject. This results in some inefficiencies—e.g., the matrix 2-norm is presented

before eigenvalues or singular values are thoroughly discussed. To compensate, I try to provide enough "wiggle room" so that an instructor can temper the inefficiencies by tailoring the approach to the students' prior background.

Comprehensiveness and Flexibility

A rather comprehensive treatment of linear algebra and its applications is presented and, consequently, the book is not meant to be devoured cover-to-cover in a typical one-semester course. However, the presentation is structured to provide flexibility in topic selection so that the text can be easily adapted to meet the demands of different course outlines without suffering breaks in continuity. Each section contains basic material paired with straightforward explanations, examples, and exercises. But every section also contains a degree of depth coupled with thought-provoking examples and exercises that can take interested students to a higher level. The exercises are formulated not only to make a student think about material from a current section, but they are designed also to pave the way for ideas in future sections in a smooth and often transparent manner. The text accommodates a variety of presentation levels by allowing instructors to select sections, discussions, examples, and exercises of appropriate sophistication. For example, traditional one-semester undergraduate courses can be taught from the basic material in Chapter 1 (Linear Equations); Chapter 2 (Rectangular Systems and Echelon Forms); Chapter 3 (Matrix Algebra); Chapter 4 (Vector Spaces); Chapter 5 (Norms, Inner Products, and Orthogonality); Chapter 6 (Determinants); and Chapter 7 (Eigenvalues and Eigenvectors). The level of the course and the degree of rigor are controlled by the selection and depth of coverage in the latter sections of Chapters 4, 5, and 7. An upper-level course might consist of a quick review of Chapters 1, 2, and 3 followed by a more in-depth treatment of Chapters 4, 5, and 7. For courses containing advanced undergraduate or graduate students, the focus can be on material in the latter sections of Chapters 4, 5, 7, and Chapter 8 (Perron–Frobenius Theory of Nonnegative Matrices). A rich two-semester course can be taught by using the text in its entirety.

What Does "Applied" Mean?

Most people agree that linear algebra is at the heart of applied science, but there are divergent views concerning what "applied linear algebra" really means; the academician's perspective is not always the same as that of the practitioner. In a poll conducted by SIAM in preparation for one of the triannual SIAM conferences on applied linear algebra, a diverse group of internationally recognized scientific corporations and government laboratories was asked how linear algebra finds application in their missions. The overwhelming response was that the primary use of linear algebra in applied industrial and laboratory work involves the development, analysis, and implementation of numerical algorithms along with some discrete and statistical modeling. The applications in this book tend to reflect this realization. While most of the popular "academic" applications are included, and "applications" to other areas of mathematics are honestly treated,

there is an emphasis on numerical issues designed to prepare students to use linear algebra in scientific environments outside the classroom.

Computing Projects

Computing projects help solidify concepts, and I include many exercises that can be incorporated into a laboratory setting. But my goal is to write a mathematics text that can last, so I don't muddy the development by marrying the material to a particular computer package or language. I am old enough to remember what happened to the FORTRAN- and APL-based calculus and linear algebra texts that came to market in the 1970s. I provide instructors with a flexible environment that allows for an ancillary computing laboratory in which any number of popular packages and lab manuals can be used in conjunction with the material in the text.

History

Finally, I believe that revealing only the scaffolding without teaching something about the scientific architects who erected it deprives students of an important part of their mathematical heritage. It also tends to dehumanize mathematics, which is the epitome of human endeavor. Consequently, I make an effort to say things (sometimes very human things that are not always complimentary) about the lives of the people who contributed to the development and applications of linear algebra. But, as I came to realize, this is a perilous task because writing history is frequently an interpretation of facts rather than a statement of facts. I considered documenting the sources of the historical remarks to help mitigate the inevitable challenges, but it soon became apparent that the sheer volume required to do so would skew the direction and flavor of the text. I can only assure the reader that I made an effort to be as honest as possible, and I tried to corroborate "facts." Nevertheless, there were times when interpretations had to be made, and these were no doubt influenced by my own views and experiences.

Supplements

Included with this text is a solutions manual and a CD-ROM. The solutions manual contains the solutions for each exercise given in the book. The solutions are constructed to be an integral part of the learning process. Rather than just providing answers, the solutions often contain details and discussions that are intended to stimulate thought and motivate material in the following sections. The CD, produced by Vickie Kearn and the people at SIAM, contains the entire book along with the solutions manual in PDF format. This electronic version of the text is completely searchable and linked. With a click of the mouse a student can jump to a referenced page, equation, theorem, definition, or proof, and then jump back to the sentence containing the reference, thereby making learning quite efficient. In addition, the CD contains material that extends historical remarks in the book and brings them to life with a large selection of

portraits, pictures, attractive graphics, and additional anecdotes. The supporting Internet site at `MatrixAnalysis.com` contains updates, errata, new material, and additional supplements as they become available.

SIAM

I thank the SIAM organization and the people who constitute it (the infrastructure as well as the general membership) for allowing me the honor of publishing my book under their name. I am dedicated to the goals, philosophy, and ideals of SIAM, and there is no other company or organization in the world that I would rather have publish this book. In particular, I am most thankful to Vickie Kearn, publisher at SIAM, for the confidence, vision, and dedication she has continually provided, and I am grateful for her patience that allowed me to write the book that I wanted to write. The talented people on the SIAM staff went far above and beyond the call of ordinary duty to make this project special. This group includes Lois Sellers (art and cover design), Michelle Montgomery and Kathleen LeBlanc (promotion and marketing), Marianne Will and Deborah Poulson (copy for CD-ROM biographies), Laura Helfrich and David Comdico (design and layout of the CD-ROM), Kelly Cuomo (linking the CD-ROM), and Kelly Thomas (managing editor for the book). Special thanks goes to Jean Anderson for her eagle-sharp editor's eye.

Acknowledgments

This book evolved over a period of several years through many different courses populated by hundreds of undergraduate and graduate students. To all my students and colleagues who have offered suggestions, corrections, criticisms, or just moral support, I offer my heartfelt thanks, and I hope to see as many of you as possible at some point in the future so that I can convey my feelings to you in person. I am particularly indebted to Michele Benzi for conversations and suggestions that led to several improvements. All writers are influenced by people who have written before them, and for me these writers include (in no particular order) Gil Strang, Jim Ortega, Charlie Van Loan, Leonid Mirsky, Ben Noble, Pete Stewart, Gene Golub, Charlie Johnson, Roger Horn, Peter Lancaster, Paul Halmos, Franz Hohn, Nick Rose, and Richard Bellman—thanks for lighting the path. I want to offer particular thanks to Richard J. Painter and Franklin A. Graybill, two exceptionally fine teachers, for giving a rough Colorado farm boy a chance to pursue his dreams. Finally, neither this book nor anything else I have done in my career would have been possible without the love, help, and unwavering support from Bethany, my friend, partner, and wife. Her multiple readings of the manuscript and suggestions were invaluable. I dedicate this book to Bethany and our children, Martin and Holly, to our granddaughter, Margaret, and to the memory of my parents, Carl and Louise Meyer.

Carl D. Meyer
April 19, 2000

CHAPTER 1

Linear Equations

1.1 INTRODUCTION

A fundamental problem that surfaces in all mathematical sciences is that of analyzing and solving m algebraic equations in n unknowns. The study of a system of simultaneous linear equations is in a natural and indivisible alliance with the study of the rectangular array of numbers defined by the coefficients of the equations. This link seems to have been made at the outset.

The earliest recorded analysis of simultaneous equations is found in the ancient Chinese book *Chiu-chang Suan-shu* (*Nine Chapters on Arithmetic*), estimated to have been written some time around 200 B.C. In the beginning of Chapter VIII, there appears a problem of the following form.

> *Three sheafs of a good crop, two sheafs of a mediocre crop, and one sheaf of a bad crop are sold for* 39 *dou. Two sheafs of good, three mediocre, and one bad are sold for* 34 *dou; and one good, two mediocre, and three bad are sold for* 26 *dou. What is the price received for each sheaf of a good crop, each sheaf of a mediocre crop, and each sheaf of a bad crop?*

Today, this problem would be formulated as three equations in three unknowns by writing

$$3x + 2y + z = 39,$$
$$2x + 3y + z = 34,$$
$$x + 2y + 3z = 26,$$

where x, y, and z represent the price for one sheaf of a good, mediocre, and bad crop, respectively. The Chinese saw right to the heart of the matter. They placed the coefficients (represented by colored bamboo rods) of this system in

a square array on a "counting board" and then manipulated the lines of the array according to prescribed rules of thumb. Their counting board techniques and rules of thumb found their way to Japan and eventually appeared in Europe with the colored rods having been replaced by numerals and the counting board replaced by pen and paper. In Europe, the technique became known as *Gaussian elimination* in honor of the German mathematician Carl Gauss,[1] whose extensive use of it popularized the method.

Because this elimination technique is fundamental, we begin the study of our subject by learning how to apply this method in order to compute solutions for linear equations. After the computational aspects have been mastered, we will turn to the more theoretical facets surrounding linear systems.

[1] Carl Friedrich Gauss (1777–1855) is considered by many to have been the greatest mathematician who has ever lived, and his astounding career requires several volumes to document. He was referred to by his peers as the "prince of mathematicians." Upon Gauss's death one of them wrote that "His mind penetrated into the deepest secrets of numbers, space, and nature; He measured the course of the stars, the form and forces of the Earth; He carried within himself the evolution of mathematical sciences of a coming century." History has proven this remark to be true.

1.2 GAUSSIAN ELIMINATION AND MATRICES

The problem is to calculate, if possible, a common solution for a system of m linear algebraic equations in n unknowns

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 &+ a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ &\vdots \\ a_{m1}x_1 &+ a_{m2}x_2 + \cdots + a_{mn}x_n = b_m, \end{aligned}$$

where the x_i's are the unknowns and the a_{ij}'s and the b_i's are known constants. The a_{ij}'s are called the ***coefficients*** of the system, and the set of b_i's is referred to as the ***right-hand side*** of the system. For any such system, there are exactly three possibilities for the set of solutions.

Three Possibilities

- **UNIQUE SOLUTION:** There is one and only one set of values for the x_i's that satisfies all equations simultaneously.
- **NO SOLUTION:** There is no set of values for the x_i's that satisfies all equations simultaneously—the solution set is empty.
- **INFINITELY MANY SOLUTIONS:** There are infinitely many different sets of values for the x_i's that satisfy all equations simultaneously. It is not difficult to prove that if a system has more than one solution, then it has infinitely many solutions. For example, it is impossible for a system to have exactly two different solutions.

Part of the job in dealing with a linear system is to decide which one of these three possibilities is true. The other part of the task is to compute the solution if it is unique or to describe the set of all solutions if there are many solutions. Gaussian elimination is a tool that can be used to accomplish all of these goals.

Gaussian elimination is a methodical process of systematically transforming one system into another simpler, but equivalent, system (two systems are called ***equivalent*** if they possess equal solution sets) by successively eliminating unknowns and eventually arriving at a system that is easily solvable. The elimination process relies on three simple operations by which to transform one system to another equivalent system. To describe these operations, let E_k denote the k^{th} equation

$$E_k: \quad a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n = b_k$$

and write the system as

$$\mathcal{S} = \left\{ \begin{array}{c} E_1 \\ E_2 \\ \vdots \\ E_m \end{array} \right\}.$$

For a linear system $\mathcal{S}$, each of the following three ***elementary operations*** results in an equivalent system $\mathcal{S}'$.

(1) Interchange the i^{th} and j^{th} equations. That is, if

$$\mathcal{S} = \left\{ \begin{array}{c} E_1 \\ \vdots \\ E_i \\ \vdots \\ E_j \\ \vdots \\ E_m \end{array} \right\}, \quad \text{then} \quad \mathcal{S}' = \left\{ \begin{array}{c} E_1 \\ \vdots \\ E_j \\ \vdots \\ E_i \\ \vdots \\ E_m \end{array} \right\}. \tag{1.2.1}$$

(2) Replace the i^{th} equation by a nonzero multiple of itself. That is,

$$\mathcal{S}' = \left\{ \begin{array}{c} E_1 \\ \vdots \\ \alpha E_i \\ \vdots \\ E_m \end{array} \right\}, \quad \text{where} \quad \alpha \neq 0. \tag{1.2.2}$$

(3) Replace the j^{th} equation by a combination of itself plus a multiple of the i^{th} equation. That is,

$$\mathcal{S}' = \left\{ \begin{array}{c} E_1 \\ \vdots \\ E_i \\ \vdots \\ E_j + \alpha E_i \\ \vdots \\ E_m \end{array} \right\}. \tag{1.2.3}$$

Providing explanations for why each of these operations cannot change the solution set is left as an exercise.

The most common problem encountered in practice is the one in which there are n equations as well as n unknowns—called a ***square system***—for which there is a unique solution. Since Gaussian elimination is straightforward for this case, we begin here and later discuss the other possibilities. What follows is a detailed description of Gaussian elimination as applied to the following simple (but typical) square system:

$$\begin{aligned} 2x &+ y + z = 1, \\ 6x &+ 2y + z = -1, \\ -2x &+ 2y + z = 7. \end{aligned} \tag{1.2.4}$$

At each step, the strategy is to focus on one position, called the ***pivot position,*** and to eliminate all terms below this position using the three elementary operations. The coefficient in the pivot position is called a ***pivotal element*** (or simply a ***pivot***), while the equation in which the pivot lies is referred to as the ***pivotal equation.*** Only nonzero numbers are allowed to be pivots. If a coefficient in a pivot position is ever 0, then the pivotal equation is interchanged with an equation *below* the pivotal equation to produce a nonzero pivot. (This is always possible for square systems possessing a unique solution.) Unless it is 0, the first coefficient of the first equation is taken as the first pivot. For example, the circled $\textcircled{2}$ in the system below is the pivot for the first step:

$$\begin{aligned} \textcircled{2}x &+ y + z = 1, \\ 6x &+ 2y + z = -1, \\ -2x &+ 2y + z = 7. \end{aligned}$$

Step 1. Eliminate all terms below the first pivot.

- Subtract three times the first equation from the second so as to produce the equivalent system:

$$\begin{aligned} \textcircled{2}x &+ y + z = 1, \\ - y &- 2z = -4 \quad (E_2 - 3E_1), \\ -2x &+ 2y + z = 7. \end{aligned}$$

- Add the first equation to the third equation to produce the equivalent system:

$$\begin{aligned} \textcircled{2}x &+ y + z = 1, \\ - y &- 2z = -4, \\ 3y &+ 2z = 8 \quad (E_3 + E_1). \end{aligned}$$

Step 2. Select a new pivot.

- For the time being, select a new pivot by moving down and to the right.[2] If this coefficient is not 0, then it is the next pivot. Otherwise, interchange with an equation *below* this position so as to bring a nonzero number into this pivotal position. In our example, -1 is the second pivot as identified below:

$$\begin{aligned} 2x &+ y &+ z &= 1, \\ &\textcircled{-1}y &- 2z &= -4, \\ &3y &+ 2z &= 8. \end{aligned}$$

Step 3. Eliminate all terms below the second pivot.

- Add three times the second equation to the third equation so as to produce the equivalent system:

$$\begin{aligned} 2x &+ y &+ z &= 1, \\ &\textcircled{-1}y &- 2z &= -4, \\ &&- 4z &= -4 \quad (E_3 + 3E_2). \end{aligned} \tag{1.2.5}$$

- In general, at each step you move down and to the right to select the next pivot, then eliminate all terms below the pivot until you can no longer proceed. In this example, the third pivot is -4, but since there is nothing below the third pivot to eliminate, the process is complete.

At this point, we say that the system has been ***triangularized.*** A triangular system is easily solved by a simple method known as ***back substitution*** in which the last equation is solved for the value of the last unknown and then substituted back into the penultimate equation, which is in turn solved for the penultimate unknown, etc., until each unknown has been determined. For our example, solve the last equation in (1.2.5) to obtain

$$z = 1.$$

Substitute $z = 1$ back into the second equation in (1.2.5) and determine

$$y = 4 - 2z = 4 - 2(1) = 2.$$

[2] The strategy of selecting pivots in numerical computation is usually a bit more complicated than simply using the next coefficient that is down and to the right. Use the down-and-right strategy for now, and later more practical strategies will be discussed.

Finally, substitute $z = 1$ and $y = 2$ back into the first equation in (1.2.5) to get

$$x = \frac{1}{2}(1 - y - z) = \frac{1}{2}(1 - 2 - 1) = -1,$$

which completes the solution.

It should be clear that there is no reason to write down the symbols such as "x," "y," "z," and "$=$" at each step since we are only manipulating the coefficients. If such symbols are discarded, then a system of linear equations reduces to a rectangular array of numbers in which each horizontal line represents one equation. For example, the system in (1.2.4) reduces to the following array:

$$\left(\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 6 & 2 & 1 & -1 \\ -2 & 2 & 1 & 7 \end{array}\right). \qquad \text{(The line emphasizes where = appeared.)}$$

The array of coefficients—the numbers on the left-hand side of the vertical line—is called the ***coefficient matrix*** for the system. The entire array—the coefficient matrix augmented by the numbers from the right-hand side of the system—is called the ***augmented matrix*** associated with the system. If the coefficient matrix is denoted by $\mathbf{A}$ and the right-hand side is denoted by $\mathbf{b}$, then the augmented matrix associated with the system is denoted by $[\mathbf{A}|\mathbf{b}]$.

Formally, a ***scalar*** is either a real number or a complex number, and a ***matrix*** is a rectangular array of scalars. It is common practice to use uppercase boldface letters to denote matrices and to use the corresponding lowercase letters with two subscripts to denote individual entries in a matrix. For example,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}.$$

The first subscript on an individual entry in a matrix designates the ***row*** (the horizontal line), and the second subscript denotes the ***column*** (the vertical line) that the entry occupies. For example, if

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 8 & 6 & 5 & -9 \\ -3 & 8 & 3 & 7 \end{pmatrix}, \quad \text{then} \quad a_{11} = 2,\ a_{12} = 1,\ \ldots,\ a_{34} = 7. \qquad (1.2.6)$$

A ***submatrix*** of a given matrix $\mathbf{A}$ is an array obtained by deleting any combination of rows and columns from $\mathbf{A}$. For example, $\mathbf{B} = \begin{pmatrix} 2 & 4 \\ -3 & 7 \end{pmatrix}$ is a submatrix of the matrix $\mathbf{A}$ in (1.2.6) because $\mathbf{B}$ is the result of deleting the second row and the second and third columns of $\mathbf{A}$.

Matrix $\mathbf{A}$ is said to have ***shape*** or ***size*** $m \times n$—pronounced "m by n"—whenever $\mathbf{A}$ has exactly m rows and n columns. For example, the matrix in (1.2.6) is a 3×4 matrix. By agreement, 1×1 matrices are identified with scalars and vice versa. To emphasize that matrix $\mathbf{A}$ has shape $m \times n$, subscripts are sometimes placed on $\mathbf{A}$ as $\mathbf{A}_{m\times n}$. Whenever $m = n$ (i.e., when $\mathbf{A}$ has the same number of rows as columns), $\mathbf{A}$ is called a ***square matrix.*** Otherwise, $\mathbf{A}$ is said to be ***rectangular.*** Matrices consisting of a single row or a single column are often called ***row vectors*** or ***column vectors***, respectively.

The symbol $\mathbf{A}_{i*}$ is used to denote the i^{th} row, while $\mathbf{A}_{*j}$ denotes the j^{th} column of matrix $\mathbf{A}$. For example, if $\mathbf{A}$ is the matrix in (1.2.6), then

$$\mathbf{A}_{2*} = \begin{pmatrix} 8 & 6 & 5 & -9 \end{pmatrix} \quad \text{and} \quad \mathbf{A}_{*2} = \begin{pmatrix} 1 \\ 6 \\ 8 \end{pmatrix}.$$

For a linear system of equations

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 + \cdots + a_{1n}x_n &&= b_1, \\ a_{21}x_1 &+ a_{22}x_2 + \cdots + a_{2n}x_n &&= b_2, \\ &\vdots \\ a_{m1}x_1 &+ a_{m2}x_2 + \cdots + a_{mn}x_n &&= b_m, \end{aligned}$$

Gaussian elimination can be executed on the associated augmented matrix $[\mathbf{A}|\mathbf{b}]$ by performing elementary operations to the rows of $[\mathbf{A}|\mathbf{b}]$. These row operations correspond to the three elementary operations (1.2.1), (1.2.2), and (1.2.3) used to manipulate linear systems. For an $m \times n$ matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{1*} \\ \vdots \\ \mathbf{M}_{i*} \\ \vdots \\ \mathbf{M}_{j*} \\ \vdots \\ \mathbf{M}_{m*} \end{pmatrix},$$

the three types of ***elementary row operations*** on $\mathbf{M}$ are as follows.

- Type I: Interchange rows i and j to produce $\begin{pmatrix} \mathbf{M}_{1*} \\ \vdots \\ \mathbf{M}_{j*} \\ \vdots \\ \mathbf{M}_{i*} \\ \vdots \\ \mathbf{M}_{m*} \end{pmatrix}$. (1.2.7)

- Type II: Replace row i by a nonzero multiple of itself to produce

$$\begin{pmatrix} \mathbf{M}_{1*} \\ \vdots \\ \alpha\mathbf{M}_{i*} \\ \vdots \\ \mathbf{M}_{m*} \end{pmatrix}, \quad \text{where} \quad \alpha \neq 0. \tag{1.2.8}$$

- Type III: Replace row j by a combination of itself plus a multiple of row i to produce

$$\begin{pmatrix} \mathbf{M}_{1*} \\ \vdots \\ \mathbf{M}_{i*} \\ \vdots \\ \mathbf{M}_{j*} + \alpha\mathbf{M}_{i*} \\ \vdots \\ \mathbf{M}_{m*} \end{pmatrix}. \tag{1.2.9}$$

To solve the system (1.2.4) by using elementary row operations, start with the associated augmented matrix $[\mathbf{A}|\mathbf{b}]$ and triangularize the coefficient matrix $\mathbf{A}$ by performing exactly the same sequence of row operations that corresponds to the elementary operations executed on the equations themselves:

$$\left(\begin{array}{ccc|c} \textcircled{2} & 1 & 1 & 1 \\ 6 & 2 & 1 & -1 \\ -2 & 2 & 1 & 7 \end{array}\right) \begin{matrix} \\ R_2 - 3R_1 \\ R_3 + R_1 \end{matrix} \longrightarrow \left(\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & \textcircled{-1} & -2 & -4 \\ 0 & 3 & 2 & 8 \end{array}\right) R_3 + 3R_2$$

$$\longrightarrow \left(\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & -4 & -4 \end{array}\right).$$

The final array represents the triangular system

$$\begin{aligned} 2x + y + z &= 1, \\ -y - 2z &= -4, \\ -4z &= -4 \end{aligned}$$

that is solved by back substitution as described earlier. In general, if an $n \times n$ system has been triangularized to the form

$$\left(\begin{array}{cccc|c} t_{11} & t_{12} & \cdots & t_{1n} & c_1 \\ 0 & t_{22} & \cdots & t_{2n} & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & t_{nn} & c_n \end{array}\right) \tag{1.2.10}$$

in which each $t_{ii} \neq 0$ (i.e., there are no zero pivots), then the general algorithm for back substitution is as follows.

Algorithm for Back Substitution

Determine the x_i's from (1.2.10) by first setting $x_n = c_n/t_{nn}$ and then recursively computing

$$x_i = \frac{1}{t_{ii}}\left(c_i - t_{i,i+1}x_{i+1} - t_{i,i+2}x_{i+2} - \cdots - t_{in}x_n\right)$$

for $i = n-1,\ n-2, \ldots, 2,\ 1$.

One way to gauge the efficiency of an algorithm is to count the number of arithmetical operations required.[3] For a variety of reasons, no distinction is made between additions and subtractions, and no distinction is made between multiplications and divisions. Furthermore, multiplications/divisions are usually counted separately from additions/subtractions. Even if you do not work through the details, it is important that you be aware of the operational counts for Gaussian elimination with back substitution so that you will have a basis for comparison when other algorithms are encountered.

Gaussian Elimination Operation Counts

Gaussian elimination with back substitution applied to an $n \times n$ system requires

$$\frac{n^3}{3} + n^2 - \frac{n}{3} \quad \text{multiplications/divisions}$$

and

$$\frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} \quad \text{additions/subtractions.}$$

As n grows, the $n^3/3$ term dominates each of these expressions. Therefore, the important thing to remember is that Gaussian elimination with back substitution on an $n \times n$ system requires about $n^3/3$ multiplications/divisions and about the same number of additions/subtractions.

[3] Operation counts alone may no longer be as important as they once were in gauging the efficiency of an algorithm. Older computers executed instructions sequentially, whereas some contemporary machines are capable of executing instructions in parallel so that different numerical tasks can be performed simultaneously. An algorithm that lends itself to parallelism may have a higher operational count but might nevertheless run faster on a parallel machine than an algorithm with a lesser operational count that cannot take advantage of parallelism.

Example 1.2.1

Problem: Solve the following system using Gaussian elimination with back substitution:

$$\begin{aligned} v - w &= 3, \\ -2u + 4v - w &= 1, \\ -2u + 5v - 4w &= -2. \end{aligned}$$

Solution: The associated augmented matrix is

$$\left(\begin{array}{ccc|c} 0 & 1 & -1 & 3 \\ -2 & 4 & -1 & 1 \\ -2 & 5 & -4 & -2 \end{array}\right).$$

Since the first pivotal position contains 0, interchange rows one and two before eliminating below the first pivot:

$$\left(\begin{array}{ccc|c} \textcircled{0} & 1 & -1 & 3 \\ -2 & 4 & -1 & 1 \\ -2 & 5 & -4 & -2 \end{array}\right) \xrightarrow{\textit{Interchange } R_1 \textit{ and } R_2} \left(\begin{array}{ccc|c} \textcircled{-2} & 4 & -1 & 1 \\ 0 & 1 & -1 & 3 \\ -2 & 5 & -4 & -2 \end{array}\right) R_3 - R_1$$

$$\longrightarrow \left(\begin{array}{ccc|c} -2 & 4 & -1 & 1 \\ 0 & \textcircled{1} & -1 & 3 \\ 0 & 1 & -3 & -3 \end{array}\right) R_3 - R_2 \longrightarrow \left(\begin{array}{ccc|c} -2 & 4 & -1 & 1 \\ 0 & 1 & -1 & 3 \\ 0 & 0 & -2 & -6 \end{array}\right).$$

Back substitution yields

$$\begin{aligned} w &= \frac{-6}{-2} = 3, \\ v &= 3 + w = 3 + 3 = 6, \\ u &= \frac{1}{-2}(1 - 4v + w) = \frac{1}{-2}(1 - 24 + 3) = 10. \end{aligned}$$

Exercises for section 1.2

1.2.1. Use Gaussian elimination with back substitution to solve the following system:

$$\begin{aligned} x_1 + x_2 + x_3 &= 1, \\ x_1 + 2x_2 + 2x_3 &= 1, \\ x_1 + 2x_2 + 3x_3 &= 1. \end{aligned}$$

1.2.2. Apply Gaussian elimination with back substitution to the following system:

$$\begin{aligned} 2x_1 - x_2 &= 0, \\ -x_1 + 2x_2 - x_3 &= 0, \\ -x_2 + x_3 &= 1. \end{aligned}$$

1.2.3. Use Gaussian elimination with back substitution to solve the following system:

$$\begin{aligned} 4x_2 - 3x_3 &= 3, \\ -x_1 + 7x_2 - 5x_3 &= 4, \\ -x_1 + 8x_2 - 6x_3 &= 5. \end{aligned}$$

1.2.4. Solve the following system:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1, \\ x_1 + x_2 + 3x_3 + 3x_4 &= 3, \\ x_1 + x_2 + 2x_3 + 3x_4 &= 3, \\ x_1 + 3x_2 + 3x_3 + 3x_4 &= 4. \end{aligned}$$

1.2.5. Consider the following three systems where the coefficients are the same for each system, but the right-hand sides are different (this situation occurs frequently):

$$\left.\begin{aligned} 4x - 8y + 5z &= 1 \\ 4x - 7y + 4z &= 0 \\ 3x - 4y + 2z &= 0 \end{aligned}\right| \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \left| \begin{matrix} 0, \\ 0, \\ 1. \end{matrix}\right.$$

Solve all three systems at one time by performing Gaussian elimination on an augmented matrix of the form

$$\left[\mathbf{A} \mid \mathbf{b_1} \mid \mathbf{b_2} \mid \mathbf{b_3}\right].$$

1.2.6. Suppose that matrix $\mathbf{B}$ is obtained by performing a sequence of row operations on matrix $\mathbf{A}$. Explain why $\mathbf{A}$ can be obtained by performing row operations on $\mathbf{B}$.

1.2.7. Find angles α, β, and γ such that

$$\begin{aligned} 2\sin\alpha - \cos\beta + 3\tan\gamma &= 3, \\ 4\sin\alpha + 2\cos\beta - 2\tan\gamma &= 2, \\ 6\sin\alpha - 3\cos\beta + \tan\gamma &= 9, \end{aligned}$$

where $0 \le \alpha \le 2\pi$, $0 \le \beta \le 2\pi$, and $0 \le \gamma < \pi$.

1.2.8. The following system has no solution:

$$\begin{aligned} -x_1 + 3x_2 - 2x_3 &= 1, \\ -x_1 + 4x_2 - 3x_3 &= 0, \\ -x_1 + 5x_2 - 4x_3 &= 0. \end{aligned}$$

Attempt to solve this system using Gaussian elimination and explain what occurs to indicate that the system is impossible to solve.

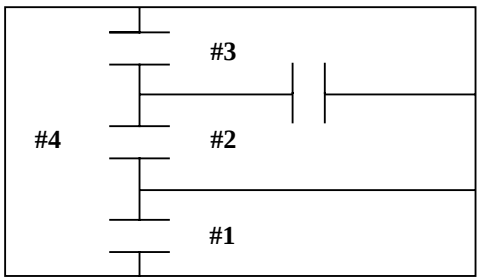
1.2.9. Attempt to solve the system

$$\begin{aligned} -x_1 + 3x_2 - 2x_3 &= 4, \\ -x_1 + 4x_2 - 3x_3 &= 5, \\ -x_1 + 5x_2 - 4x_3 &= 6, \end{aligned}$$

using Gaussian elimination and explain why this system must have infinitely many solutions.

1.2.10. By solving a 3×3 system, find the coefficients in the equation of the parabola $y = \alpha + \beta x + \gamma x^2$ that passes through the points $(1, 1)$, $(2, 2)$, and $(3, 0)$.

1.2.11. Suppose that 100 insects are distributed in an enclosure consisting of four chambers with passageways between them as shown below.



At the end of one minute, the insects have redistributed themselves. Assume that a minute is not enough time for an insect to visit more than one chamber and that at the end of a minute 40% of the insects in each chamber have not left the chamber they occupied at the beginning of the minute. The insects that leave a chamber disperse uniformly among the chambers that are directly accessible from the one they initially occupied—e.g., from #3, half move to #2 and half move to #4.

(a) If at the end of one minute there are 12, 25, 26, and 37 insects in chambers #1, #2, #3, and #4, respectively, determine what the initial distribution had to be.
(b) If the initial distribution is 20, 20, 20, 40, what is the distribution at the end of one minute?

1.2.12. Show that the three types of elementary row operations discussed on p. 8 are not independent by showing that the interchange operation (1.2.7) can be accomplished by a sequence of the other two types of row operations given in (1.2.8) and (1.2.9).

1.2.13. Suppose that $[\mathbf{A}|\mathbf{b}]$ is the augmented matrix associated with a linear system. You know that performing row operations on $[\mathbf{A}|\mathbf{b}]$ does not change the solution of the system. However, no mention of *column operations* was ever made because column operations can alter the solution.
(a) Describe the effect on the solution of a linear system when columns $\mathbf{A}_{*j}$ and $\mathbf{A}_{*k}$ are interchanged.
(b) Describe the effect when column $\mathbf{A}_{*j}$ is replaced by $\alpha\mathbf{A}_{*j}$ for $\alpha \neq 0$.
(c) Describe the effect when $\mathbf{A}_{*j}$ is replaced by $\mathbf{A}_{*j} + \alpha\mathbf{A}_{*k}$.
Hint: Experiment with a 2×2 or 3×3 system.

1.2.14. Consider the $n \times n$ ***Hilbert matrix*** defined by

$$\mathbf{H} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{pmatrix}.$$

Express the individual entries h_{ij} in terms of i and j.

1.2.15. Verify that the operation counts given in the text for Gaussian elimination with back substitution are correct for a general 3×3 system. If you are up to the challenge, try to verify these counts for a general $n \times n$ system.

1.2.16. Explain why a linear system can never have exactly two different solutions. Extend your argument to explain the fact that if a system has more than one solution, then it must have infinitely many different solutions.

1.3 GAUSS–JORDAN METHOD

The purpose of this section is to introduce a variation of Gaussian elimination that is known as the ***Gauss–Jordan method.*** [4] The two features that distinguish the Gauss–Jordan method from standard Gaussian elimination are as follows.

- At each step, the pivot element is forced to be 1.
- At each step, all terms *above* the pivot as well as all terms below the pivot are eliminated.

In other words, if

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right)$$

is the augmented matrix associated with a linear system, then elementary row operations are used to reduce this matrix to

$$\left(\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & s_1 \\ 0 & 1 & \cdots & 0 & s_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & s_n \end{array}\right).$$

The solution then appears in the last column (i.e., $x_i = s_i$) so that this procedure circumvents the need to perform back substitution.

Example 1.3.1

Problem: Apply the Gauss–Jordan method to solve the following system:

$$\begin{aligned} 2x_1 + 2x_2 + 6x_3 &= 4, \\ 2x_1 + x_2 + 7x_3 &= 6, \\ -2x_1 - 6x_2 - 7x_3 &= -1. \end{aligned}$$

[4] Although there has been some confusion as to which Jordan should receive credit for this algorithm, it now seems clear that the method was in fact introduced by a geodesist named Wilhelm Jordan (1842–1899) and not by the more well known mathematician Marie Ennemond Camille Jordan (1838–1922), whose name is often mistakenly associated with the technique, but who is otherwise correctly credited with other important topics in matrix analysis, the "Jordan canonical form" being the most notable. Wilhelm Jordan was born in southern Germany, educated in Stuttgart, and was a professor of geodesy at the technical college in Karlsruhe. He was a prolific writer, and he introduced his elimination scheme in the 1888 publication *Handbuch der Vermessungskunde.* Interestingly, a method similar to W. Jordan's variation of Gaussian elimination seems to have been discovered and described independently by an obscure Frenchman named Clasen, who appears to have published only one scientific article, which appeared in 1888—the same year as W. Jordan's *Handbuch* appeared.

Solution: The sequence of operations is indicated in parentheses and the pivots are circled.

$$\left(\begin{array}{ccc|c} \textcircled{2} & 2 & 6 & 4 \\ 2 & 1 & 7 & 6 \\ -2 & -6 & -7 & -1 \end{array}\right) R_1/2 \longrightarrow \left(\begin{array}{ccc|c} \textcircled{1} & 1 & 3 & 2 \\ 2 & 1 & 7 & 6 \\ -2 & -6 & -7 & -1 \end{array}\right) \begin{array}{l} \\ R_2 - 2R_1 \\ R_3 + 2R_1 \end{array}$$

$$\longrightarrow \left(\begin{array}{ccc|c} \textcircled{1} & 1 & 3 & 2 \\ 0 & -1 & 1 & 2 \\ 0 & -4 & -1 & 3 \end{array}\right) (-R_2) \longrightarrow \left(\begin{array}{ccc|c} 1 & 1 & 3 & 2 \\ 0 & \textcircled{1} & -1 & -2 \\ 0 & -4 & -1 & 3 \end{array}\right) \begin{array}{l} R_1 - R_2 \\ \\ R_3 + 4R_2 \end{array}$$

$$\longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & 4 & 4 \\ 0 & \textcircled{1} & -1 & -2 \\ 0 & 0 & -5 & -5 \end{array}\right) -R_3/5 \longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & 4 & 4 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & \textcircled{1} & 1 \end{array}\right) \begin{array}{l} R_1 - 4R_3 \\ R_2 + R_3 \\ \\ \end{array}$$

$$\longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & \textcircled{1} & 1 \end{array}\right).$$

Therefore, the solution is $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$.

On the surface it may seem that there is little difference between the Gauss–Jordan method and Gaussian elimination with back substitution because eliminating terms above the pivot with Gauss–Jordan seems equivalent to performing back substitution. But this is not correct. Gauss–Jordan requires more arithmetic than Gaussian elimination with back substitution.

Gauss–Jordan Operation Counts

For an $n \times n$ system, the Gauss–Jordan procedure requires

$$\frac{n^3}{2} + \frac{n^2}{2} \quad \text{multiplications/divisions}$$

and

$$\frac{n^3}{2} - \frac{n}{2} \quad \text{additions/subtractions.}$$

In other words, the Gauss–Jordan method requires about $n^3/2$ multiplications/divisions and about the same number of additions/subtractions.

Recall from the previous section that Gaussian elimination with back substitution requires only about $n^3/3$ multiplications/divisions and about the same

number of additions/subtractions. Compare this with the $n^3/2$ factor required by the Gauss–Jordan method, and you can see that Gauss–Jordan requires about 50% *more* effort than Gaussian elimination with back substitution. For small systems of the textbook variety (e.g., $n = 3$), these comparisons do not show a great deal of difference. However, in practical work, the systems that are encountered can be quite large, and the difference between Gauss–Jordan and Gaussian elimination with back substitution can be significant. For example, if $n = 100$, then $n^3/3$ is about 333,333, while $n^3/2$ is 500,000, which is a difference of 166,667 multiplications/divisions as well as that many additions/subtractions.

Although the Gauss–Jordan method is not recommended for solving linear systems that arise in practical applications, it does have some theoretical advantages. Furthermore, it can be a useful technique for tasks other than computing solutions to linear systems. We will make use of the Gauss–Jordan procedure when matrix inversion is discussed—this is the primary reason for introducing Gauss–Jordan.

Exercises for section 1.3

1.3.1. Use the Gauss–Jordan method to solve the following system:

$$\begin{aligned} 4x_2 - 3x_3 &= 3, \\ -x_1 + 7x_2 - 5x_3 &= 4, \\ -x_1 + 8x_2 - 6x_3 &= 5. \end{aligned}$$

1.3.2. Apply the Gauss–Jordan method to the following system:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1, \\ x_1 + 2x_2 + 2x_3 + 2x_4 &= 0, \\ x_1 + 2x_2 + 3x_3 + 3x_4 &= 0, \\ x_1 + 2x_2 + 3x_3 + 4x_4 &= 0. \end{aligned}$$

1.3.3. Use the Gauss–Jordan method to solve the following three systems at the same time.

$$\left.\begin{aligned} 2x_1 - x_2 &= 1 \\ -x_1 + 2x_2 - x_3 &= 0 \\ -x_2 + x_3 &= 0 \end{aligned}\right| \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \left| \begin{matrix} 0, \\ 0, \\ 1. \end{matrix}\right.$$

1.3.4. Verify that the operation counts given in the text for the Gauss–Jordan method are correct for a general 3×3 system. If you are up to the challenge, try to verify these counts for a general $n \times n$ system.

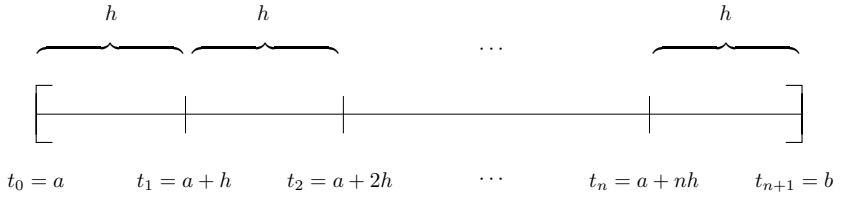
1.4 TWO-POINT BOUNDARY VALUE PROBLEMS

It was stated previously that linear systems that arise in practice can become quite large in size. The purpose of this section is to understand why this often occurs and why there is frequently a special structure to the linear systems that come from practical applications.

Given an interval $[a, b]$ and two numbers α and β, consider the general problem of trying to find a function $y(t)$ that satisfies the differential equation

$$u(t)y''(t)+v(t)y'(t)+w(t)y(t) = f(t), \quad \text{where} \quad y(a) = \alpha \text{ and } y(b) = \beta. \tag{1.4.1}$$

The functions u, v, w, and f are assumed to be known functions on $[a, b]$. Because the unknown function $y(t)$ is specified at the boundary points a and b, problem (1.4.1) is known as a ***two-point boundary value problem.*** Such problems abound in nature and are frequently very hard to handle because it is often not possible to express $y(t)$ in terms of elementary functions. Numerical methods are usually employed to approximate $y(t)$ at discrete points inside $[a, b]$. Approximations are produced by subdividing the interval $[a, b]$ into $n+1$ equal subintervals, each of length $h = (b - a)/(n + 1)$ as shown below.



Derivative approximations at the interior nodes (grid points) $t_i = a + ih$ are made by using Taylor series expansions $y(t) = \sum_{k=0}^{\infty} y^{(k)}(t_i)(t - t_i)^k/k!$ to write

$$\begin{aligned} y(t_i + h) &= y(t_i) + y'(t_i)h + \frac{y''(t_i)h^2}{2!} + \frac{y'''(t_i)h^3}{3!} + \cdots, \\ y(t_i - h) &= y(t_i) - y'(t_i)h + \frac{y''(t_i)h^2}{2!} - \frac{y'''(t_i)h^3}{3!} + \cdots, \end{aligned} \tag{1.4.2}$$

and then subtracting and adding these expressions to produce

$$y'(t_i) = \frac{y(t_i + h) - y(t_i - h)}{2h} + O(h^3)$$

and

$$y''(t_i) = \frac{y(t_i - h) - 2y(t_i) + y(t_i + h)}{h^2} + O(h^4),$$

where $O(h^p)$ denotes[5] terms containing p^{th} and higher powers of h. The

[5] Formally, a function $f(h)$ is $O(h^p)$ if $f(h)/h^p$ remains bounded as $h \to 0$, but $f(h)/h^q$ becomes unbounded if $q > p$. This means that f goes to zero as fast as h^p goes to zero.

resulting approximations

$$y'(t_i) \approx \frac{y(t_i+h) - y(t_i-h)}{2h} \text{ and } y''(t_i) \approx \frac{y(t_i-h) - 2y(t_i) + y(t_i+h)}{h^2} \quad (1.4.3)$$

are called ***centered difference approximations,*** and they are preferred over less accurate one-sided approximations such as

$$y'(t_i) \approx \frac{y(t_i+h) - y(t_i)}{h} \quad \text{or} \quad y'(t_i) \approx \frac{y(t) - y(t-h)}{h}.$$

The value $h = (b-a)/(n+1)$ is called the ***step size.*** Smaller step sizes produce better derivative approximations, so obtaining an accurate solution usually requires a small step size and a large number of grid points. By evaluating the centered difference approximations at each grid point and substituting the result into the original differential equation (1.4.1), a system of n linear equations in n unknowns is produced in which the unknowns are the values $y(t_i)$. A simple example can serve to illustrate this point.

Example 1.4.1

Suppose that $f(t)$ is a known function and consider the two-point boundary value problem

$$y''(t) = f(t) \quad \text{on } [0,1] \text{ with } y(0) = y(1) = 0.$$

The goal is to approximate the values of y at n equally spaced grid points t_i interior to $[0,1]$. The step size is therefore $h = 1/(n+1)$. For the sake of convenience, let $y_i = y(t_i)$ and $f_i = f(t_i)$. Use the approximation

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} \approx y''(t_i) = f_i$$

along with $y_0 = 0$ and $y_{n+1} = 0$ to produce the system of equations

$$-y_{i-1} + 2y_i - y_{i+1} \approx -h^2 f_i \quad \text{for } i = 1, 2, \ldots, n.$$

(The signs are chosen to make the 2's positive to be consistent with later developments.) The augmented matrix associated with this system is shown below:

$$\left(\begin{array}{ccccccc|c} 2 & -1 & 0 & \cdots & 0 & 0 & 0 & -h^2 f_1 \\ -1 & 2 & -1 & \cdots & 0 & 0 & 0 & -h^2 f_2 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 & -h^2 f_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 & -h^2 f_{n-2} \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 & -h^2 f_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 & -h^2 f_n \end{array}\right).$$

By solving this system, approximate values of the unknown function y at the grid points t_i are obtained. Larger values of n produce smaller values of h and hence better approximations to the exact values of the y_i's.

Notice the pattern of the entries in the coefficient matrix in the above example. The nonzero elements occur only on the subdiagonal, main-diagonal, and superdiagonal lines—such a system (or matrix) is said to be ***tridiagonal.*** This is characteristic in the sense that when finite difference approximations are applied to the general two-point boundary value problem, a tridiagonal system is the result.

Tridiagonal systems are particularly nice in that they are inexpensive to solve. When Gaussian elimination is applied, only two multiplications/divisions are needed at each step of the triangularization process because there is at most only one nonzero entry below and to the right of each pivot. Furthermore, Gaussian elimination preserves all of the zero entries that were present in the original tridiagonal system. This makes the back substitution process cheap to execute because there are at most only two multiplications/divisions required at each substitution step. Exercise 3.10.6 contains more details.

Exercises for section 1.4

1.4.1. Divide the interval $[0, 1]$ into five equal subintervals, and apply the finite difference method in order to approximate the solution of the two-point boundary value problem

$$y''(t) = 125t, \quad y(0) = y(1) = 0$$

at the four interior grid points. Compare your approximate values at the grid points with the exact solution at the grid points. **Note:** You should not expect very accurate approximations with only four interior grid points.

1.4.2. Divide $[0, 1]$ into $n+1$ equal subintervals, and apply the finite difference approximation method to derive the linear system associated with the two-point boundary value problem

$$y''(t) - y'(t) = f(t), \quad y(0) = y(1) = 0.$$

1.4.3. Divide $[0, 1]$ into five equal subintervals, and approximate the solution to

$$y''(t) - y'(t) = 125t, \quad y(0) = y(1) = 0$$

at the four interior grid points. Compare the approximations with the exact values at the grid points.

1.5 MAKING GAUSSIAN ELIMINATION WORK

Now that you understand the basic Gaussian elimination technique, it's time to turn it into a practical algorithm that can be used for realistic applications. For pencil and paper computations where you are doing exact arithmetic, the strategy is to keep things as simple as possible (like avoiding messy fractions) in order to minimize those "stupid arithmetic errors" we are all prone to make. But very few problems in the real world are of the textbook variety, and practical applications involving linear systems usually demand the use of a computer. Computers don't care about messy fractions, and they don't introduce errors of the "stupid" variety. Computers produce a more predictable kind of error, called *roundoff error,* and it's important[6] to spend a little time up front to understand this kind of error and its effects on solving linear systems.

Numerical computation in digital computers is performed by approximating the infinite set of real numbers with a finite set of numbers as described below.

Floating-Point Numbers

A t-digit, base-β ***floating-point number*** has the form

$$f = \pm .d_1 d_2 \cdots d_t \times \beta^{\epsilon} \quad \text{with} \quad d_1 \neq 0,$$

where the base β, the exponent ϵ, and the digits $0 \leq d_i \leq \beta - 1$ are integers. For internal machine representation, $\beta = 2$ (binary representation) is standard, but for pencil-and-paper examples it's more convenient to use $\beta = 10$. The value of t, called the ***precision,*** and the exponent ϵ can vary with the choice of hardware and software.

Floating-point numbers are just adaptations of the familiar concept of scientific notation where $\beta = 10$, which will be the value used in our examples. For any fixed set of values for t, β, and ϵ, the corresponding set $\mathcal{F}$ of floating-point numbers is necessarily a finite set, so some real numbers can't be found in $\mathcal{F}$. There is more than one way of approximating real numbers with floating-point numbers. For the remainder of this text, the following common *rounding convention* is adopted. Given a real number x, the floating-point approximation $fl(x)$ is defined to be the nearest element in $\mathcal{F}$ to x, and in case of a tie we round away from 0. This means that for t-digit precision with $\beta = 10$, we need

[6] The computer has been the single most important scientific and technological development of our century and has undoubtedly altered the course of science for all future time. The prospective young scientist or engineer who passes through a contemporary course in linear algebra and matrix theory and fails to learn at least the elementary aspects of what is involved in solving a practical linear system with a computer is missing a fundamental tool of applied mathematics.

to look at digit d_{t+1} in $x = .d_1d_2\cdots d_td_{t+1}\cdots \times 10^{\epsilon}$ (making sure $d_1 \neq 0$) and then set

$$fl(x) = \begin{cases} .d_1d_2\cdots d_t \times 10^{\epsilon} & \text{if } d_{t+1} < 5, \\ ([.d_1d_2\cdots d_t] + 10^{-t}) \times 10^{\epsilon} & \text{if } d_{t+1} \geq 5. \end{cases}$$

For example, in 2-digit, base-10 floating-point arithmetic,

$$fl\,(3/80) = fl(.0375) = fl(.375 \times 10^{-1}) = .38 \times 10^{-1} = .038.$$

By considering $\eta = 1/3$ and $\xi = 3$ with t-digit base-10 arithmetic, it's easy to see that

$$fl(\eta + \xi) \neq fl(\eta) + fl(\xi) \quad \text{and} \quad fl(\eta\xi) \neq fl(\eta)fl(\xi).$$

Furthermore, several familiar rules of real arithmetic do not hold for floating-point arithmetic—associativity is one outstanding example. This, among other reasons, makes the analysis of floating-point computation difficult. It also means that you must be careful when working the examples and exercises in this text because although most calculators and computers can be instructed to display varying numbers of digits, most have a fixed internal precision with which all calculations are made before numbers are displayed, and this internal precision cannot be altered. Almost certainly, the internal precision of your calculator or computer is greater than the precision called for by the examples and exercises in this text. This means that each time you perform a t-digit calculation, you should manually round the result to t *significant* digits and reenter the rounded number before proceeding to the next calculation. In other words, don't "chain" operations in your calculator or computer.

To understand how to execute Gaussian elimination using floating-point arithmetic, let's compare the use of exact arithmetic with the use of 3-digit base-10 arithmetic to solve the following system:

$$\begin{aligned} 47x + 28y &= 19, \\ 89x + 53y &= 36. \end{aligned}$$

Using Gaussian elimination with exact arithmetic, we multiply the first equation by the multiplier $m = 89/47$ and subtract the result from the second equation to produce

$$\left(\begin{array}{cc|c} 47 & 28 & 19 \\ 0 & -1/47 & 1/47 \end{array}\right).$$

Back substitution yields the *exact solution*

$$x = 1 \quad \text{and} \quad y = -1.$$

Using 3-digit arithmetic, the multiplier is

$$fl(m) = fl\left(\frac{89}{47}\right) = .189 \times 10^{1} = 1.89.$$

Since

$$fl\Big(fl(m)fl(47)\Big) = fl(1.89 \times 47) = .888 \times 10^2 = 88.8,$$
$$fl\Big(fl(m)fl(28)\Big) = fl(1.89 \times 28) = .529 \times 10^2 = 52.9,$$
$$fl\Big(fl(m)fl(19)\Big) = fl(1.89 \times 19) = .359 \times 10^2 = 35.9,$$

the first step of 3-digit Gaussian elimination is as shown below:

$$\left(\begin{array}{cc|c} 47 & 28 & 19 \\ fl(89-88.8) & fl(53-52.9) & fl(36-35.9) \end{array}\right)$$
$$= \left(\begin{array}{cc|c} 47 & 28 & 19 \\ \textcircled{.2} & .1 & .1 \end{array}\right).$$

The goal is to triangularize the system—to produce a zero in the circled (2,1)-position—but this cannot be accomplished with 3-digit arithmetic. Unless the circled value $\textcircled{.2}$ is replaced by 0, back substitution cannot be executed. *Henceforth, we will agree simply to enter* 0 *in the position that we are trying to annihilate,* regardless of the value of the floating-point number that might actually appear. The value of the position being annihilated is generally not even computed. For example, don't even bother computing

$$fl\Big[89 - fl\big(fl(m)fl(47)\big)\Big] = fl(89 - 88.8) = .2$$

in the above example. Hence the result of 3-digit Gaussian elimination for this example is

$$\left(\begin{array}{cc|c} 47 & 28 & 19 \\ 0 & .1 & .1 \end{array}\right).$$

Apply 3-digit back substitution to obtain the 3-digit floating-point solution

$$y = fl\left(\frac{.1}{.1}\right) = 1,$$
$$x = fl\left(\frac{19-28}{47}\right) = fl\left(\frac{-9}{47}\right) = -.191.$$

The vast discrepancy between the exact solution $(1, -1)$ and the 3-digit solution $(-.191, 1)$ illustrates some of the problems we can expect to encounter while trying to solve linear systems with floating-point arithmetic. Sometimes using a higher precision may help, but this is not always possible because on all machines there are natural limits that make extended precision arithmetic impractical past a certain point. Even if it is possible to increase the precision, it

may not buy you very much because there are many cases for which an increase in precision does not produce a comparable decrease in the accumulated roundoff error. Given any particular precision (say, t), it is not difficult to provide examples of linear systems for which the computed t-digit solution is just as bad as the one in our 3-digit example above.

Although the effects of rounding can almost never be eliminated, there are some simple techniques that can help to minimize these machine induced errors.

Partial Pivoting

At each step, search the positions on and below the pivotal position for the coefficient of *maximum magnitude.* If necessary perform the appropriate row interchange to bring this maximal coefficient into the pivotal position. Illustrated below is the third step in a typical case:

$$\left(\begin{array}{ccccc|c} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & \textcircled{S} & * & * & * \\ 0 & 0 & S & * & * & * \\ 0 & 0 & S & * & * & * \end{array}\right).$$

Search the positions in the third column marked "S" for the coefficient of maximal magnitude and, if necessary, interchange rows to bring this coefficient into the circled pivotal position. Simply stated, the strategy is to maximize the magnitude of the pivot at each step by using only row interchanges.

On the surface, it is probably not apparent why partial pivoting should make a difference. The following example not only shows that partial pivoting can indeed make a great deal of difference, but it also indicates what makes this strategy effective.

Example 1.5.1

It is easy to verify that the exact solution to the system

$$\begin{aligned} -10^{-4}x + y &= 1, \\ x + y &= 2, \end{aligned}$$

is given by

$$x = \frac{1}{1.0001} \quad \text{and} \quad y = \frac{1.0002}{1.0001}.$$

If 3-digit arithmetic *without* partial pivoting is used, then the result is

$$\left(\begin{array}{cc|c} -10^{-4} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) R_2 + 10^4 R_1 \longrightarrow \left(\begin{array}{cc|c} -10^{-4} & 1 & 1 \\ 0 & 10^4 & 10^4 \end{array}\right)$$

because

$$fl(1 + 10^4) = fl(.10001 \times 10^5) = .100 \times 10^5 = 10^4 \tag{1.5.1}$$

and

$$fl(2 + 10^4) = fl(.10002 \times 10^5) = .100 \times 10^5 = 10^4. \tag{1.5.2}$$

Back substitution now produces

$$x = 0 \quad \text{and} \quad y = 1.$$

Although the computed solution for y is close to the exact solution for y, the computed solution for x is not very close to the exact solution for x—the computed solution for x is certainly not accurate to three significant figures as you might hope. If 3-digit arithmetic *with* partial pivoting is used, then the result is

$$\left(\begin{array}{cc|c} -10^{-4} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ -10^{-4} & 1 & 1 \end{array}\right) R_2 + 10^{-4} R_1$$
$$\longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right)$$

because

$$fl(1 + 10^{-4}) = fl(.10001 \times 10^1) = .100 \times 10^1 = 1 \tag{1.5.3}$$

and

$$fl(1 + 2 \times 10^{-4}) = fl(.10002 \times 10^1) = .100 \times 10^1 = 1. \tag{1.5.4}$$

This time, back substitution produces the computed solution

$$x = 1 \quad \text{and} \quad y = 1,$$

which is as close to the exact solution as one can reasonably expect—the computed solution agrees with the exact solution to three significant digits.

Why did partial pivoting make a difference? The answer lies in comparing (1.5.1) and (1.5.2) with (1.5.3) and (1.5.4).

Without partial pivoting the multiplier is 10^4, and this is so large that it completely swamps the arithmetic involving the relatively smaller numbers 1 and 2 and prevents them from being taken into account. That is, the smaller numbers 1 and 2 are "blown away" as though they were never present so that our 3-digit computer produces the *exact* solution to another system, namely,

$$\left(\begin{array}{cc|c} -10^{-4} & 1 & 1 \\ 1 & 0 & 0 \end{array}\right),$$

which is quite different from the original system. *With* partial pivoting the multiplier is 10^{-4}, and this is small enough so that it does not swamp the numbers 1 and 2. In this case, the 3-digit computer produces the *exact* solution to the system $\left(\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 1 & 2 \end{array}\right)$, which is close to the original system.[7]

In summary, the villain in Example 1.5.1 is the large multiplier that prevents some smaller numbers from being fully accounted for, thereby resulting in the exact solution of another system that is very different from the original system. By maximizing the magnitude of the pivot at each step, we minimize the magnitude of the associated multiplier thus helping to control the growth of numbers that emerge during the elimination process. This in turn helps circumvent some of the effects of roundoff error. The problem of growth in the elimination procedure is more deeply analyzed on p. 348.

When partial pivoting is used, no multiplier ever exceeds 1 in magnitude. To see that this is the case, consider the following two typical steps in an elimination procedure:

$$\left(\begin{array}{ccccc|c} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & \textcircled{p} & * & * & * \\ 0 & 0 & q & * & * & * \\ 0 & 0 & r & * & * & * \end{array}\right) \begin{array}{l} \\ \\ \\ R_4 - (q/p)R_3 \\ R_5 - (r/p)R_3 \end{array} \longrightarrow \left(\begin{array}{ccccc|c} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & \textcircled{p} & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * & * \end{array}\right).$$

The pivot is p, while q/p and r/p are the multipliers. If partial pivoting has been employed, then $|p| \geq |q|$ and $|p| \geq |r|$ so that

$$\left|\frac{q}{p}\right| \leq 1 \quad \text{and} \quad \left|\frac{r}{p}\right| \leq 1.$$

By guaranteeing that no multiplier exceeds 1 in magnitude, the possibility of producing relatively large numbers that can swamp the significance of smaller numbers is much reduced, but not completely eliminated. To see that there is still more to be done, consider the following example.

Example 1.5.2

The exact solution to the system

$$\begin{aligned} -10x + 10^5 y &= 10^5, \\ x + \quad y &= 2, \end{aligned}$$

[7] Answering the question, "What system have I really solved (i.e., obtained the exact solution of), and how close is this system to the original system," is called *backward error analysis,* as opposed to forward analysis in which one tries to answer the question, "How close will a computed solution be to the exact solution?" Backward analysis has proven to be an effective way to analyze the numerical stability of algorithms.

is given by

$$x = \frac{1}{1.0001} \quad \text{and} \quad y = \frac{1.0002}{1.0001}.$$

Suppose that 3-digit arithmetic with partial pivoting is used. Since $|-10| > 1$, no interchange is called for and we obtain

$$\left(\begin{array}{cc|c} -10 & 10^5 & 10^5 \\ 1 & 1 & 2 \end{array}\right) \begin{array}{l} \\ R_2 + 10^{-1}R_1 \end{array} \longrightarrow \left(\begin{array}{cc|c} -10 & 10^5 & 10^5 \\ 0 & 10^4 & 10^4 \end{array}\right)$$

because

$$fl(1 + 10^4) = fl(.10001 \times 10^5) = .100 \times 10^5 = 10^4$$

and

$$fl(2 + 10^4) = fl(.10002 \times 10^5) = .100 \times 10^5 = 10^4.$$

Back substitution yields

$$x = 0 \quad \text{and} \quad y = 1,$$

which must be considered to be very bad—the computed 3-digit solution for y is not too bad, but the computed 3-digit solution for x is terrible!

What is the source of difficulty in Example 1.5.2? This time, the multiplier cannot be blamed. The trouble stems from the fact that the first equation contains coefficients that are much larger than the coefficients in the second equation. That is, there is a problem of *scale* due to the fact that the coefficients are of different orders of magnitude. Therefore, we should somehow rescale the system before attempting to solve it.

If the first equation in the above example is rescaled to insure that the coefficient of maximum magnitude is a 1, which is accomplished by multiplying the first equation by 10^{-5}, then the system given in Example 1.5.1 is obtained, and we know from that example that partial pivoting produces a very good approximation to the exact solution.

This points to the fact that the success of partial pivoting can hinge on maintaining the proper scale among the coefficients. Therefore, the second refinement needed to make Gaussian elimination practical is a reasonable scaling strategy. Unfortunately, there is no known scaling procedure that will produce optimum results for every possible system, so we must settle for a strategy that will work most of the time. The strategy is to combine ***row scaling***—multiplying selected rows by nonzero multipliers—with ***column scaling***—multiplying selected columns of the coefficient matrix $\mathbf{A}$ by nonzero multipliers.

Row scaling doesn't alter the exact solution, but column scaling does—see Exercise 1.2.13(b). Column scaling is equivalent to changing the units of the k^{th} unknown. For example, if the units of the k^{th} unknown x_k in $[\mathbf{A}|\mathbf{b}]$ are millimeters, and if the k^{th} column of $\mathbf{A}$ is multiplied by $.001$, then the k^{th} unknown in the scaled system $[\hat{\mathbf{A}} \mid \mathbf{b}]$ is $\hat{x}_i = 1000x_i$, and thus the units of the scaled unknown $\hat{x}_k$ become meters.

Experience has shown that the following strategy for combining row scaling with column scaling usually works reasonably well.

Practical Scaling Strategy

1. Choose units that are natural to the problem and do not distort the relationships between the sizes of things. These natural units are usually self-evident, and further column scaling past this point is not ordinarily attempted.
2. Row scale the system $[\mathbf{A}|\mathbf{b}]$ so that the coefficient of maximum magnitude in each row of $\mathbf{A}$ is equal to 1. That is, divide each equation by the coefficient of maximum magnitude.

Partial pivoting together with the scaling strategy described above makes Gaussian elimination with back substitution an extremely effective tool. Over the course of time, this technique has proven to be reliable for solving a majority of linear systems encountered in practical work.

Although it is not extensively used, there is an extension of partial pivoting known as *complete pivoting* which, in some special cases, can be more effective than partial pivoting in helping to control the effects of roundoff error.

Complete Pivoting

If $[\mathbf{A}|\mathbf{b}]$ is the augmented matrix at the k^{th} step of Gaussian elimination, then search the pivotal position together with every position in $\mathbf{A}$ that is below or to the right of the pivotal position for the coefficient of maximum magnitude. If necessary, perform the appropriate row and column interchanges to bring the coefficient of maximum magnitude into the pivotal position. Shown below is the third step in a typical situation:

$$\left(\begin{array}{ccccc|c} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & \textcircled{S} & S & S & * \\ 0 & 0 & S & S & S & * \\ 0 & 0 & S & S & S & * \end{array}\right)$$

Search the positions marked "S" for the coefficient of maximal magnitude. If necessary, interchange rows and columns to bring this maximal coefficient into the circled pivotal position. Recall from Exercise 1.2.13 that the effect of a column interchange in $\mathbf{A}$ is equivalent to permuting (or renaming) the associated unknowns.

You should be able to see that complete pivoting should be at least as effective as partial pivoting. Moreover, it is possible to construct specialized examples where complete pivoting is superior to partial pivoting—a famous example is presented in Exercise 1.5.7. However, one rarely encounters systems of this nature in practice. A deeper comparison between no pivoting, partial pivoting, and complete pivoting is given on p. 348.

Example 1.5.3

Problem: Use 3-digit arithmetic together with complete pivoting to solve the following system:

$$\begin{aligned} x - \quad y &= -2, \\ -9x + 10y &= 12. \end{aligned}$$

Solution: Since 10 is the coefficient of maximal magnitude that lies in the search pattern, interchange the first and second rows and then interchange the first and second columns:

$$\left(\begin{array}{cc|c} 1 & -1 & -2 \\ -9 & 10 & 12 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} -9 & 10 & 12 \\ 1 & -1 & -2 \end{array}\right)$$
$$\longrightarrow \left(\begin{array}{cc|c} 10 & -9 & 12 \\ -1 & 1 & -2 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} 10 & -9 & 12 \\ 0 & .1 & -.8 \end{array}\right).$$

The effect of the column interchange is to rename the unknowns to $\hat{x}$ and $\hat{y}$, where $\hat{x} = y$ and $\hat{y} = x$. Back substitution yields $\hat{y} = -8$ and $\hat{x} = -6$ so that

$$x = \hat{y} = -8 \quad \text{and} \quad y = \hat{x} = -6.$$

In this case, the 3-digit solution and the exact solution agree. If only partial pivoting is used, the 3-digit solution will not be as accurate. However, if scaled partial pivoting is used, the result is the same as when complete pivoting is used.

If the cost of using complete pivoting was nearly the same as the cost of using partial pivoting, we would always use complete pivoting. However, it is not difficult to show that complete pivoting approximately doubles the cost over straight Gaussian elimination, whereas partial pivoting adds only a negligible amount. Couple this with the fact that it is extremely rare to encounter a practical system where scaled partial pivoting is not adequate while complete pivoting is, and it is easy to understand why complete pivoting is seldom used in practice. Gaussian elimination with scaled partial pivoting is the preferred method for dense systems (i.e., not a lot of zeros) of moderate size.

Exercises for section 1.5

1.5.1. Consider the following system:

$$\begin{aligned} 10^{-3}x - y &= 1, \\ x + y &= 0. \end{aligned}$$

(a) Use 3-digit arithmetic with no pivoting to solve this system.
(b) Find a system that is exactly satisfied by your solution from part (a), and note how close this system is to the original system.
(c) Now use partial pivoting and 3-digit arithmetic to solve the original system.
(d) Find a system that is exactly satisfied by your solution from part (c), and note how close this system is to the original system.
(e) Use exact arithmetic to obtain the solution to the original system, and compare the exact solution with the results of parts (a) and (c).
(f) Round the exact solution to three significant digits, and compare the result with those of parts (a) and (c).

1.5.2. Consider the following system:

$$\begin{aligned} x + \quad y &= 3, \\ -10x + 10^5 y &= 10^5. \end{aligned}$$

(a) Use 4-digit arithmetic with partial pivoting and no scaling to compute a solution.
(b) Use 4-digit arithmetic with complete pivoting and no scaling to compute a solution of the original system.
(c) This time, row scale the original system first, and then apply partial pivoting with 4-digit arithmetic to compute a solution.
(d) Now determine the exact solution, and compare it with the results of parts (a), (b), and (c).

1.5.3. With no scaling, compute the 3-digit solution of

$$\begin{aligned} -3x + \ y &= -2, \\ 10x - 3y &= 7, \end{aligned}$$

without partial pivoting and with partial pivoting. Compare your results with the exact solution.

1.5.4. Consider the following system in which the coefficient matrix is the Hilbert matrix:

$$\begin{aligned} x + \frac{1}{2}y + \frac{1}{3}z &= \frac{1}{3}, \\ \frac{1}{2}x + \frac{1}{3}y + \frac{1}{4}z &= \frac{1}{3}, \\ \frac{1}{3}x + \frac{1}{4}y + \frac{1}{5}z &= \frac{1}{5}. \end{aligned}$$

(a) First convert the coefficients to 3-digit floating-point numbers, and then use 3-digit arithmetic with partial pivoting but with no scaling to compute the solution.

(b) Again use 3-digit arithmetic, but row scale the coefficients (after converting them to floating-point numbers), and then use partial pivoting to compute the solution.

(c) Proceed as in part (b), but this time row scale the coefficients *before each elimination step.*

(d) Now use exact arithmetic on the original system to determine the exact solution, and compare the result with those of parts (a), (b), and (c).

1.5.5. To see that changing units can affect a floating-point solution, consider a mining operation that extracts silica, iron, and gold from the earth. Capital (measured in dollars), operating time (in hours), and labor (in man-hours) are needed to operate the mine. To extract a pound of silica requires \$.0055, .0011 hours of operating time, and .0093 man-hours of labor. For each pound of iron extracted, \$.095, .01 operating hours, and .025 man-hours are required. For each pound of gold extracted, \$960, 112 operating hours, and 560 man-hours are required.

(a) Suppose that during 600 hours of operation, exactly \$5000 and 3000 man-hours are used. Let x, y, and z denote the number of pounds of silica, iron, and gold, respectively, that are recovered during this period. Set up the linear system whose solution will yield the values for x, y, and z.

(b) With no scaling, use 3-digit arithmetic and partial pivoting to compute a solution $(\tilde{x}, \tilde{y}, \tilde{z})$ of the system of part (a). Then approximate the exact solution (x, y, z) by using your machine's (or calculator's) full precision with partial pivoting to solve the system in part (a), and compare this with your 3-digit solution by computing the relative error defined by

$$e_r = \frac{\sqrt{(x-\tilde{x})^2 + (y-\tilde{y})^2 + (z-\tilde{z})^2}}{\sqrt{x^2 + y^2 + z^2}}.$$

(c) Using 3-digit arithmetic, column scale the coefficients by changing units: convert pounds of silica to tons of silica, pounds of iron to half-tons of iron, and pounds of gold to troy ounces of gold (1 lb. = 12 troy oz.).

(d) Use 3-digit arithmetic with partial pivoting to solve the column scaled system of part (c). Then approximate the exact solution by using your machine's (or calculator's) full precision with partial pivoting to solve the system in part (c), and compare this with your 3-digit solution by computing the relative error e_r as defined in part (b).

1.5.6. Consider the system given in Example 1.5.3.

(a) Use 3-digit arithmetic with partial pivoting but with no scaling to solve the system.

(b) Now use partial pivoting with scaling. Does complete pivoting provide an advantage over scaled partial pivoting in this case?

1.5.7. Consider the following well-scaled matrix:

$$\mathbf{W}_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 1 \\ -1 & 1 & 0 & \cdots & 0 & 0 & 1 \\ -1 & -1 & 1 & \ddots & 0 & 0 & 1 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ -1 & -1 & -1 & \ddots & 1 & 0 & 1 \\ -1 & -1 & -1 & \cdots & -1 & 1 & 1 \\ -1 & -1 & -1 & \cdots & -1 & -1 & 1 \end{pmatrix}.$$

(a) Reduce $\mathbf{W}_n$ to an upper-triangular form using Gaussian elimination with partial pivoting, and determine the element of maximal magnitude that emerges during the elimination procedure.

(b) Now use complete pivoting and repeat part (a).

(c) Formulate a statement comparing the results of partial pivoting with those of complete pivoting for $\mathbf{W_n}$, and describe the effect this would have in determining the t-digit solution for a system whose augmented matrix is $[\mathbf{W_n} \mid \mathbf{b}]$.

1.5.8. Suppose that $\mathbf{A}$ is an $n \times n$ matrix of real numbers that has been scaled so that each entry satisfies $|a_{ij}| \leq 1$, and consider reducing $\mathbf{A}$ to triangular form using Gaussian elimination with partial pivoting. Demonstrate that after k steps of the process, no entry can have a magnitude that exceeds 2^k. **Note:** The previous exercise shows that there are cases where it is possible for some elements to actually attain the maximum magnitude of 2^k after k steps.

1.6 ILL-CONDITIONED SYSTEMS

Gaussian elimination with partial pivoting on a properly scaled system is perhaps the most fundamental algorithm in the practical use of linear algebra. However, it is not a universal algorithm nor can it be used blindly. The purpose of this section is to make the point that when solving a linear system some discretion must always be exercised because there are some systems that are so inordinately sensitive to small perturbations that *no* numerical technique can be used with confidence.

Example 1.6.1

Consider the system

$$.835x + .667y = .168,$$
$$.333x + .266y = .067,$$

for which the exact solution is

$$x = 1 \quad \text{and} \quad y = -1.$$

If $b_2 = .067$ is only slightly perturbed to become $\hat{b}_2 = .066$, then the exact solution changes dramatically to become

$$\hat{x} = -666 \quad \text{and} \quad \hat{y} = 834.$$

This is an example of a system whose solution is extremely sensitive to a small perturbation. This sensitivity is intrinsic to the system itself and is not a result of any numerical procedure. Therefore, you cannot expect some "numerical trick" to remove the sensitivity. If the exact solution is sensitive to small perturbations, then any computed solution cannot be less so, regardless of the algorithm used.

Ill-Conditioned Linear Systems

A system of linear equations is said to be ***ill-conditioned*** when some small perturbation in the system can produce relatively large changes in the exact solution. Otherwise, the system is said to be ***well-conditioned.***

It is easy to visualize what causes a 2×2 system to be ill-conditioned. Geometrically, two equations in two unknowns represent two straight lines, and the point of intersection is the solution for the system. An ill-conditioned system represents two straight lines that are almost parallel.

If two straight lines are almost parallel and if one of the lines is tilted only slightly, then the point of intersection (i.e., the solution of the associated 2×2 linear system) is drastically altered.

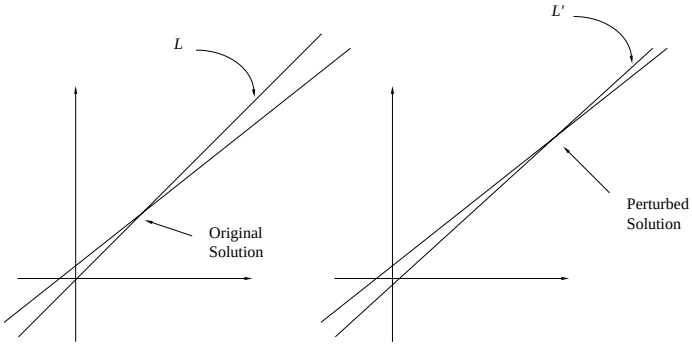


FIGURE 1.6.1

This is illustrated in Figure 1.6.1 in which line L is slightly perturbed to become line L'. Notice how this small perturbation results in a large change in the point of intersection. This was exactly the situation for the system given in Example 1.6.1. In general, ill-conditioned systems are those that represent almost parallel lines, almost parallel planes, and generalizations of these notions.

Because roundoff errors can be viewed as perturbations to the original coefficients of the system, employing even a generally good numerical technique—short of exact arithmetic—on an ill-conditioned system carries the risk of producing nonsensical results.

In dealing with an ill-conditioned system, the engineer or scientist is often confronted with a much more basic (and sometimes more disturbing) problem than that of simply trying to solve the system. Even if a minor miracle could be performed so that the exact solution could be extracted, the scientist or engineer might still have a nonsensical solution that could lead to totally incorrect conclusions. The problem stems from the fact that the coefficients are often empirically obtained and are therefore known only within certain tolerances. For an ill-conditioned system, a small uncertainty in any of the coefficients can mean an extremely large uncertainty may exist in the solution. This large uncertainty can render even the exact solution totally useless.

Example 1.6.2

Suppose that for the system

$$\begin{aligned} .835x + .667y &= b_1 \\ .333x + .266y &= b_2 \end{aligned}$$

the numbers b_1 and b_2 are the results of an experiment and must be read from the dial of a test instrument. Suppose that the dial can be read to within a

tolerance of $\pm.001$, and assume that values for b_1 and b_2 are read as $.168$ and $.067$, respectively. This produces the ill-conditioned system of Example 1.6.1, and it was seen in that example that the exact solution of the system is

$$(x, y) = (1, -1). \tag{1.6.1}$$

However, due to the small uncertainty in reading the dial, we have that

$$.167 \leq b_1 \leq .169 \quad \text{and} \quad .066 \leq b_2 \leq .068. \tag{1.6.2}$$

For example, this means that the solution associated with the reading $(b_1, b_2) = (.168, .067)$ is just as valid as the solution associated with the reading $(b_1, b_2) = (.167, .068)$, or the reading $(b_1, b_2) = (.169, .066)$, or any other reading falling in the range (1.6.2). For the reading $(b_1, b_2) = (.167, .068)$, the exact solution is

$$(x, y) = (934, -1169), \tag{1.6.3}$$

while for the other reading $(b_1, b_2) = (.169, .066)$, the exact solution is

$$(x, y) = (-932, 1167). \tag{1.6.4}$$

Would you be willing to be the first to fly in the plane or drive across the bridge whose design incorporated a solution to this problem? I wouldn't! There is just too much uncertainty. Since no one of the solutions (1.6.1), (1.6.3), or (1.6.4) can be preferred over any of the others, it is conceivable that totally different designs might be implemented depending on how the technician reads the last significant digit on the dial. Due to the ill-conditioned nature of an associated linear system, the successful design of the plane or bridge may depend on blind luck rather than on scientific principles.

Rather than trying to extract accurate solutions from ill-conditioned systems, engineers and scientists are usually better off investing their time and resources in trying to redesign the associated experiments or their data collection methods so as to avoid producing ill-conditioned systems.

There is one other discomforting aspect of ill-conditioned systems. It concerns what students refer to as "checking the answer" by substituting a computed solution back into the left-hand side of the original system of equations to see how close it comes to satisfying the system—that is, producing the right-hand side. More formally, if

$$x_c = (\, \xi_1 \quad \xi_2 \quad \cdots \quad \xi_n \,)$$

is a computed solution for a system

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 &+ a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ &\vdots \\ a_{n1}x_1 &+ a_{n2}x_2 + \cdots + a_{nn}x_n = b_n, \end{aligned}$$

then the numbers

$$r_i = a_{i1}\xi_1 + a_{i2}\xi_2 + \cdots + a_{in}\xi_n - b_i \quad \text{for } i = 1, 2, \ldots, n$$

are called the ***residuals.*** Suppose that you compute a solution x_c and substitute it back to find that all the residuals are relatively small. Does this guarantee that x_c is close to the exact solution? Surprisingly, the answer is a resounding "no!" whenever the system is ill-conditioned.

Example 1.6.3

For the ill-conditioned system given in Example 1.6.1, suppose that somehow you compute a solution to be

$$\xi_1 = -666 \quad \text{and} \quad \xi_2 = 834.$$

If you attempt to "check the error" in this computed solution by substituting it back into the original system, then you find—using exact arithmetic—that the residuals are

$$r_1 = .835\xi_1 + .667\xi_2 - .168 = 0,$$
$$r_2 = .333\xi_1 + .266\xi_2 - .067 = -.001.$$

That is, the computed solution $(-666, 834)$ *exactly* satisfies the first equation and comes *very close* to satisfying the second. On the surface, this might seem to suggest that the computed solution should be very close to the exact solution. In fact a naive person could probably be seduced into believing that the computed solution is within $\pm.001$ of the exact solution. Obviously, this is nowhere close to being true since the exact solution is

$$x = 1 \quad \text{and} \quad y = -1.$$

This is always a shock to a student seeing this illustrated for the first time because it is counter to a novice's intuition. Unfortunately, many students leave school believing that they can always "check" the accuracy of their computations by simply substituting them back into the original equations—it is good to know that you're not among them.

This raises the question, *"How can I check a computed solution for accuracy?"* Fortunately, if the system is well-conditioned, then the residuals do indeed provide a more effective measure of accuracy (a rigorous proof along with more insight appears in Example 5.12.2 on p. 416). But this means that you must be able to answer some additional questions. For example, how can one tell beforehand if a given system is ill-conditioned? How can one measure the extent of ill-conditioning in a linear system?

One technique to determine the extent of ill-conditioning might be to experiment by slightly perturbing selected coefficients and observing how the solution

changes. If a radical change in the solution is observed for a small perturbation to some set of coefficients, then you have uncovered an ill-conditioned situation. If a given perturbation does not produce a large change in the solution, then nothing can be concluded—perhaps you perturbed the wrong set of coefficients.

By performing several such experiments using different sets of coefficients, a feel (but not a guarantee) for the extent of ill-conditioning can be obtained. This is expensive and not very satisfying. But before more can be said, more sophisticated tools need to be developed—the topics of sensitivity and conditioning are revisited on p. 127 and in Example 5.12.1 on p. 414.

Exercises for section 1.6

1.6.1. Consider the ill-conditioned system of Example 1.6.1:

$$.835x + .667y = .168,$$
$$.333x + .266y = .067.$$

(a) Describe the outcome when you attempt to solve the system using 5-digit arithmetic with no scaling.
(b) Again using 5-digit arithmetic, first row scale the system before attempting to solve it. Describe to what extent this helps.
(c) Now use 6-digit arithmetic with no scaling. Compare the results with the exact solution.
(d) Using 6-digit arithmetic, compute the residuals for your solution of part (c), and interpret the results.
(e) For the same solution obtained in part (c), again compute the residuals, but use 7-digit arithmetic this time, and interpret the results.
(f) Formulate a concluding statement that summarizes the points made in parts (a)–(e).

1.6.2. Perturb the ill-conditioned system given in Exercise 1.6.1 above so as to form the following system:

$$.835x + .667y = .1669995,$$
$$.333x + .266y = .066601.$$

(a) Determine the exact solution, and compare it with the exact solution of the system in Exercise 1.6.1.
(b) On the basis of the results of part (a), formulate a statement concerning the necessity for the solution of an ill-conditioned system to undergo a radical change for every perturbation of the original system.

1.6.3. Consider the two straight lines determined by the graphs of the following two equations:

$$.835x + .667y = .168,$$
$$.333x + .266y = .067.$$

(a) Use 5-digit arithmetic to compute the slopes of each of the lines, and then use 6-digit arithmetic to do the same. In each case, sketch the graphs on a coordinate system.
(b) Show by diagram why a small perturbation in either of these lines can result in a large change in the solution.
(c) Describe in geometrical terms the situation that must exist in order for a system to be optimally well-conditioned.

1.6.4. Using geometric considerations, rank the following three systems according to their condition.

(a) $\begin{aligned} 1.001x - y &= .235, \\ x + .0001y &= .765. \end{aligned}$ (b) $\begin{aligned} 1.001x - y &= .235, \\ x + .9999y &= .765. \end{aligned}$

(c) $\begin{aligned} 1.001x + y &= .235, \\ x + .9999y &= .765. \end{aligned}$

1.6.5. Determine the exact solution of the following system:

$$\begin{aligned} 8x + 5y + 2z &= 15, \\ 21x + 19y + 16z &= 56, \\ 39x + 48y + 53z &= 140. \end{aligned}$$

Now change 15 to 14 in the first equation and again solve the system with exact arithmetic. Is the system ill-conditioned?

1.6.6. Show that the system

$$\begin{aligned} v - w - x - y - z &= 0, \\ w - x - y - z &= 0, \\ x - y - z &= 0, \\ y - z &= 0, \\ z &= 1, \end{aligned}$$

is ill-conditioned by considering the following perturbed system:

$$\begin{aligned} v - w - x - y - z &= 0, \\ -\frac{1}{15}v + w - x - y - z &= 0, \\ -\frac{1}{15}v + x - y - z &= 0, \\ -\frac{1}{15}v + y - z &= 0, \\ -\frac{1}{15}v + z &= 1. \end{aligned}$$

1.6.7. Let $f(x) = \sin \pi x$ on $[0, 1]$. The object of this problem is to determine the coefficients α_i of the cubic polynomial

$$p(x) = \sum_{i=0}^{3} \alpha_i x^i$$

that is as close to $f(x)$ as possible in the sense that

$$\begin{aligned} r &= \int_0^1 [f(x) - p(x)]^2 dx \\ &= \int_0^1 [f(x)]^2 dx - 2\sum_{i=0}^{3} \alpha_i \int_0^1 x^i f(x) dx + \int_0^1 \left(\sum_{i=0}^{3} \alpha_i x^i \right)^2 dx \end{aligned}$$

is as small as possible.

(a) In order to minimize r, impose the condition that $\partial r / \partial \alpha_i = 0$ for each $i = 0, 1, 2, 3$, and show this results in a system of linear equations whose augmented matrix is $[\mathbf{H_4} \mid \mathbf{b}]$, where $\mathbf{H_4}$ and $\mathbf{b}$ are given by

$$\mathbf{H_4} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} \frac{2}{\pi} \\ \frac{1}{\pi} \\ \frac{1}{\pi} - \frac{4}{\pi^3} \\ \frac{1}{\pi} - \frac{6}{\pi^3} \end{pmatrix}.$$

Any matrix $\mathbf{H_n}$ that has the same form as $\mathbf{H_4}$ is called a ***Hilbert matrix*** of order n.

(b) Systems involving Hilbert matrices are badly ill-conditioned, and the ill-conditioning becomes worse as the size increases. Use exact arithmetic with Gaussian elimination to reduce $\mathbf{H_4}$ to triangular form. Assuming that the case in which $n = 4$ is typical, explain why a general system $[\mathbf{H_n} \mid \mathbf{b}]$ will be ill-conditioned. Notice that even complete pivoting is of no help.

To isolate mathematics from the practical demands of the sciences is to invite the sterility of a cow shut away from the bulls.
— Pafnuty Lvovich Chebyshev (1821–1894)