Chapter 2

LINEAR EQUATIONS and MATRIX INVERSION

2.1 Properties of Matrices

2.1.1 Definitions and Notation

Definition 2.1.1 *An $m \times n$ matrix A is a rectangular array of numbers denoted by $A = [a_{i,j}], 1 \leq i \leq m$, $1 \leq j \leq n$, or less compactly by*

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}.$$

The element $a_{i,j}$ appearing in the i-th row and j-th column of A is called the (i,j)th element of A.

Definition 2.1.2 *If $m = n$ then A is said to be a square matrix of order n.*

Definition 2.1.3 *If $m = 1$ then A is a row vector of order n while if $n = 1$ then A is a column vector of order m.*

When we wish to emphasize that a matrix is a vector we use a lower case letter. Upper case letters denote a matrix that is either not a vector or not thought of as a vector. For square matrices we adhere to the custom of saying "matrix of order n" instead of "square matrix of order n". Further, if we wish to stress n we write A_n in place of A. With respect to vectors, the letters $a, b, c, \ldots, q$ denote column vectors whose elements are part of given data while the letters x, y, z stand for column vectors determined by solving equations of the form $Ax = f$.

Definition 2.1.4 *A^T denotes the transpose of a matrix A and is defined by $A^T = [a_{i,j}]^T = [a_{j,i}]$. If A is $m \times n$ then A^T is $n \times m$. If a is a column vector then a^T is a row vector.*

Definition 2.1.5 *The $m \times n$ matrices $A = [a_{i,j}]$ and $B = [b_{i,j}]$ are equal if and only if $a_{i,j} = b_{i,j}$ for all i, j.*

Definition 2.1.6 *The square matrix A is symmetric if $A^T = A$.*

Definition 2.1.7 *The system of equations*

$$Ax = f \tag{2.1.1}$$

is said to be consistent for the given A and f if there exists a vector x such that (2.1.1) is satisfied. Otherwise (2.1.1) is said to be inconsistent.

Please note that if m, the row dimension of A, is greater than n, the column dimension of A, then one should proceed under the assumption that the system (2.1.1) is inconsistent. Despite lacking a solution inconsistent systems of the form (2.1.1) often arise in practice. One could simply ignore enough equations to generate a solution but a better alternative is to choose the x that minimizes the average error of the m equations. This puts us into the realm of least squares analysis, a subject we wish to defer for a later chapter. Therefore, for purposes of this chapter we address solve (2.1.1) assuming that the row dimension of A is less than its column dimension

When A is a dense matrix, that is most of its elements are nonzero, then factoring A into the product of upper and lower triangular matrices is usually the most economical means of solving $Ax = f$.

Definition 2.1.8 *$L = [l_{i,j}], 1 \leq i, j \leq n$, is a lower triangular matrix if $l_{i,j} = 0$ when $i < j$. Similarly, $U = [u_{i,j}], 1 \leq i, j \leq n$, is an upper triangular matrix if $u_{i,j} = 0$ when $i > j$.*

Definition 2.1.9 *The square matrix A is diagonal if it is both lower and upper triangular.*

Definition 2.1.10 *The unit or identity matrix of order n is denoted by I where unambiguous and I_n otherwise. I is defined by $I = [\delta_{i,j}]$ where*

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2.1.2}$$

Definition 2.1.11 *The determinant $d(A)$ of a square matrix A is defined by*

$$d(A) = \sum_{\sigma} \mu(\sigma) a_{1,\sigma(1)} a_{2,\sigma(2)} \cdots a_{n,\sigma(n)} \tag{2.1.3}$$

where σ runs over all $n!$ permutations of n integers, and $\mu(\sigma)$ is -1 if σ is an odd permuation, $+1$ if σ is an even permutation.

Since this definition may be unfamiliar to the student we offer the alternative definition given in terms of minors.

Definition 2.1.12 *The minor $M_{i,j}$ of the matrix A is formed by throwing away row i and column j of A.*

Definition 2.1.13 *The cofactor $A_{i,j}$ of A is*

$$A_{i,j} = (-1)^{i+j} d(M_{i,j})$$

Definition 2.1.14 *The determinant of the $n \times n$ matrix A is a linear combination of any row i with the cofactors of row i; that is,*

$$d(A) = a_{i,1} A_{i,1} + a_{i,2} A_{i,2} + \cdots + a_{i,n} A_{i,n} \ . \tag{2.1.4}$$

A familiar means of evaluating the determinant by hand uses (2.1.4) repeatedly along with

$$d\left(\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \right) = a_{1,1} a_{2,2} - a_{1,2} a_{2,1}$$

The identity matrix and the notion of determinants are necessary for defining eigenvalues.

Definition 2.1.15 *If for a nonzero vector x there exists a scalar λ such that*

$$Ax = \lambda I x = \lambda x$$

then λ, x is called an eigenvalue, eigenvector pair of A.

Note that for the eigenvector x, λI acting on x is equivalent to A acting on x.

Theorem 2.1.16 *The number λ is an eigenvalue of A if and only if $d(A - \lambda I) = 0$.*

Definition 2.1.17 *For the $n \times n$ matrix A the n-th degree polynomial*

$$p(\lambda) = d(A - \lambda I)$$

is called the characteristic equation of A.

Definition 2.1.18 *The set of all eigenvalues of the matrix A is called the spectrum of A.*

Definition 2.1.19 *The square matrix A is said to be invertible if there exists a matrix B such that $AB = BA = I$. B is traditionally denoted A^{-1}.*

We mention that it is possible to calculate A^{-1} in terms of cofactors and $d(A)$, however, numerically it is difficult to find a worse choice of algorithms for the task.

2.1.2 Matrix Arithmetic

Throughout the list of definitions it is clear that some notion of matrix arithmetic is necessary. The easiest operation is scalar multiplication where if λ is a scalar and $A = [a_{i,j}]$ then the elements of λA are $\lambda a_{i,j}$. The addition of matrices $A = [a_{i,j}]$ and $B = [b_{i,j}]$ requires that their dimensions be identical; if so $C = A + B, C = [c_{i,j}]$, is given elementwise by

$$c_{i,j} = a_{i,j} + b_{i,j} \ .$$

Matrix multiplication is considerably less intuitive. If the number of columns of $A = [a_{i,j}]$, $1 \le i \le m$, $1 \le j \le n$, equals the number of rows of $B = [b_{j,k}]$, $1 \le j \le n$, $1 \le k \le p$, then $C = AB, C = [c_{i,k}]$ is defined with

$$c_{i,k} = \sum_{j=1}^{n} a_{i,j} b_{j,k}, \ \ 1 \le i \le m, 1 \le k \le p.$$

Thus the row dimension of C is inherited from A and the column dimension from B.

Theorem 2.1.20 *Let α and β be scalars, then assuming compatible dimensions the following are easy to prove using properties of real numbers.*

(i) $A + B = B + A$

(ii) $(A + B) + C = A + (B + C)$

(iii) $\alpha(A + B) = \alpha A + \alpha B$

(iv) $(\alpha + \beta)A = \alpha A + \beta A$

(v) $A = IA$

(vi) $B = BI$

(vii) $A(B + C) = AB + AC$

(viii) $(A + B)C = AC + BC$

(xi) $(\alpha A)B = A(\alpha B)$

(x) $(AB)C = A(BC)$

(xi) $(A + B)^T = A^T + B^T$

(xii) $(A^T)^T = A$

(xiii) $(AB)^T = B^T A^T$.

Theorem 2.1.21 *Assuming the matrices A and B are both of order n, it can be shown that*

(xiv) $(A^{-1})^{-1} = A$, *if A is invertible*

(xv) $(AB)^{-1} = B^{-1}A^{-1}$, *if A and B are invertible*

(xvi) $d(AB) = d(A)d(B)$.

Triangular matrices lead to computational savings when performing matrix arithmetic. For instance, the determinant of a triangular matrix is the product of its diagonal entries (see Problem 2.1.9). We emphasize at this point that formula (2.1.3) is not a practical means of evaluating $d(A_n)$ for $n > 4$, rather it is usually more economical to first factor A_n into LU, then use $d(A_n) = d(L)d(U)$.

2.1.3 Vector Spaces and Linear Independence

In this subsection we address the existence and uniqueness of solutions to $Ax = f$.

Definition 2.1.22 *A set S is a vector space if for all $\alpha, \beta \in \mathbb{R}, v_1, v_2 \in S$, $\alpha v_1 + \beta v_2 \in S$.*

Example 2.1.23 The set of column vectors with n components is referred to as the vector space $\mathbb{R}^n$.

Example 2.1.24 The set of vectors $[x, y, 0]^T$ is a vector space since

$$\alpha \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha x_1 + \beta x_2 \\ \alpha y_1 + \beta y_2 \\ 0 \end{bmatrix}$$

maintains the 0 in the third position.

Example 2.1.25 The set of vectors $[x, y, 1]^T$ is not a vector space since for $\alpha = \beta = 1$

$$\alpha \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \\ 2 \end{bmatrix}.$$

Definition 2.1.26 *A subset S_1 of a vector space S is called a subspace of S if S_1 is a vector space in its own right.*

Example 2.1.24 implies the x-y plane is a two-dimensional subspace of $\mathbb{R}^3$, that is, any linear combination of vectors in the x-y plane remains in the x-y plane. More generally, $\mathbb{R}^m$ is a subspace of $\mathbb{R}^n$ if $m \leq n$.

It is well-known that any vector $[x, y, z]^T$ in $\mathbb{R}^3$ may be written in terms of $\hat{\imath} = [1, 0, 0]^T$, $\hat{\jmath} = [0, 1, 0]^T$, and $\hat{k} = [0, 0, 1]^T$ as $[x, y, z]^T = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$. Alternatively the vector may also be written

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{3} z \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} + (\frac{1}{2} y - \frac{1}{6} z) \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} + (x - \frac{1}{2} y + \frac{1}{6} z) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

However, using the vectors $[1, 2, 3]^T, [4, 5, 6]^T$, and $[7, 8, 9]^T$ it is not possible to represent $[x, y, z]^T = [1, 1, 0]^T$ while if

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -\frac{1}{3} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \frac{1}{3} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = -\frac{1}{3} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} + \frac{1}{3} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

then its representation is not unique. The criterion governing the situation is the linear independence of vectors.

Definition 2.1.27 *The set of vectors $v_1, v_2, \ldots, v_n$ is linearly independent if and only if*

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = 0 \Rightarrow \alpha_1 = \alpha_2 = \cdots = \alpha_n = 0.$$

Otherwise $v_1, v_2, \ldots, v_n$ are linearly dependent.

Definition 2.1.28 *The span of $v_1, v_2, \ldots, v_n$ is the set of all linear combinations of the vectors.*

Definition 2.1.29 *If $v_1, v_2, \ldots, v_n$ are linearly independent then they are said to form a basis for their span.*

An immediate consequence is that if $\{v_1, v_2, \ldots, v_n\}$ are linearly independent and $f \in \text{span}\{v_1, v_2, \ldots, v_n\}$ then f admits a unique representation in terms of the basis.

Turning to $A_n x = f$, we first write the system as

$$\begin{bmatrix} & \vdots & & \vdots & & \vdots & \\ a_1 & \vdots & a_2 & \vdots & \cdots & \vdots & a_n \\ & \vdots & & \vdots & & \vdots & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = f \tag{2.1.5}$$

where a_j is the j-th column of A treated as a vector. Expanding (2.1.5) yields

$$x_1 a_1 + x_2 a_2 + \cdots + x_n a_n = f$$

or, in other words, solving the system is equivalent to finding the correct linear combination of $a_1, a_2, \ldots, a_n$ that produces f. If no such linear combination exists then $\{a_1, a_2, \ldots, a_n\}$ is linearly dependent and f does not lie in $\text{span}\{a_1, a_2, \ldots, a_n\}$. The system is called inconsistent. If more than one solution x exists then again $\{a_1, a_2, \ldots, a_n\}$ is linearly dependent but now $f \in \text{span}\{a_1, a_2, \ldots, a_n\}$. Finally, a unique solution implies $\{a_1, a_2, \ldots, a_n\}$ is a basis for $\mathbb{R}^n$. Theorem 2.1.30 is stated without proof (see ?).

Theorem 2.1.30 *Given A the following are equivalent*

(i) $Ax = f$ *has a unique solution for all* f.

(ii) $Ax = 0$ *has only the trivial solution* $x = 0$.

(iii) A^{-1} *exists.*

(iv) $det(A) \neq 0$.

(v) The eigenvalues of A *are nonzero.*

Problems

2.1.1 Find AB and BA if

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ -1 & -2 & -3 & -4 \\ 5 & -6 & -7 & -8 \end{bmatrix}.$$

Conclude that in general $AB \neq BA$.

2.1.2 Show that the inverse of

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{bmatrix} \text{ is } A^{-1} = \begin{bmatrix} -2 & 0 & 1 \\ 0 & 3 & -2 \\ 1 & -2 & 1 \end{bmatrix}.$$

2.1.3 Find the eigenvalues and eigenvectors of the matrix A given in Problem 2.1.2

2.1.4 Show that the matrix B in Definition 2.1.19 is unique. Hint: assume that B is a right inverse, i.e. $AB = I$, and C is a left inverse, i.e. $CA = I$. Show that $B = C$.

2.1.5 Show that $d(A)$ is the product of the eigenvalues of A. Hint: evaluate $p(0)$ where $p(\lambda) = d(A - \lambda I)$.

2.1.6 If λ is an eigenvalue of A with eigenvector x show that λ^k is an eigenvalue of A^k with eigenvector x.

2.1.7 If λ is an eigenvalue of the nonsingular matrix A then $1/\lambda$ is an eigenvalue of A^{-1}. Prove.

2.1.8 Show that the spectrums of A^T and A are identical.

2.1.9 Prove that the eigenvalues of a triangular matrix A_n are its diagonal entries, hence,

$$d(A_n) = \prod_{i=1}^{n} a_{i,i} \ .$$

2.1.10 Prove that if any diagonal element of

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} & a_{2,3} \\ 0 & 0 & a_{3,3} \end{bmatrix}$$

is zero, then the columns of A are linearly dependent.

2.1.11 Let $D(\alpha; i)$ denote the matrix obtained from I by replacing the 1 in the (i,i) position of I by α. Show that the operation $D(\alpha; i)A$ replaces row i of A by α row i while the operation $AD(\alpha; i)$ replaces column i of A by α column i.

2.1.12 Let $E(\alpha; i, j)$ denote the matrix obtained from I by replacing the zero in the (i, j)-th position of I by α. Show that the operation $E(\alpha; i, j)A$ replaces row i of A by row i + α row j; the operation $AE(\alpha; i, j)$ replaces column j of A by column j + α column i.

2.2 Forward and Backward Substitution

The solution of $Ax = f$ is greatly simplified when A is either upper or lower triangular. Let us illustrate with an example,

$$\begin{bmatrix} l_{1,1} & 0 & 0 \\ l_{2,1} & l_{2,2} & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, l_{i,i} \neq 0 \ i = 1, 2, 3. \tag{2.2.1}$$

The unkown x_1 is determined from the first equation

$$l_{1,1}x_1 = a_1, \text{ or } x_1 = a_1/l_{1,1}.$$

The second equation is

$$l_{2,1}x_1 + l_{2,2}x_2 = a_2, \text{ or } x_2 = (a_2 - l_{2,1}x_1)/l_{2,2}$$

which is quickly evaluated since x_1 is already known. Similarly

$$l_{3,1}x_1 + l_{3,2}x_2 + l_{3,3}x_3 = a_3, \text{ or } x_3 = (a_3 - l_{3,1}x_1 - l_{3,2}x_2)/l_{3,3}.$$

In general, if $L = [l_{i,j}]$ is an $n \times n$ lower triangular matrix the system $Lx = a$ is solved via forward substitution.

Algorithm 2.2.1 Forward Substitution

$$\begin{aligned}
&\text{if } l_{1,1} = 0 \rightarrow \text{exit} \\
&x_1 = a_1/l_{1,1} \\
&i = 2, \ldots, n; 1 \\
&\quad \text{if } l_{i,i} = 0 \rightarrow \text{exit} \\
&\quad x_i = (a_i - \textstyle\sum_{j=1}^{i-1} l_{i,j}x_j)/l_{i,i}
\end{aligned}$$

Forward substitution is straightforward and results in a unique solution as long as no $l_{i,i}$ vanishes. If at least one of the $l_{i,i}$ is zero the method fails.

We next consider solving $Uy = b$, $U = [u_{i,j}]$, $u_{i,i} \neq 0$, $i = 1, 2, \ldots, n$. Thus we again examine a simple system of order 3.

$$\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

We first solve for y_3, then for y_2, and finally for y_1:

$$\begin{aligned}
y_3 &= b_3/u_{3,3} \\
y_2 &= (b_2 - u_{2,3}y_3)/u_{2,2} \\
y_1 &= (b_1 - u_{1,2}y_2 - u_{1,3}y_3)/u_{1,1}
\end{aligned}$$

Again problems arise if for some i, $u_{i,i} = 0$. The procedure shown here is a specific example of backward substitution. Problem 2.2.1 asks the student to sketch out the algorithm for an $n \times n$ upper triangular matrix.

Problems

2.2.1 Produce an algorithm for backward substitution.

2.2.2 Consider $f(x) = \int x^n e^{ax}\,dx$. Repeated integrations by parts establish that $f(x) = p(x)e^{ax}$ where $p(x) = c_0 + c_1 x + \cdots + c_n x^n$. Since we know that

$$e^{-ax} f'(x) = x^n$$

it is possible to determine $c_0, c_1, \ldots, c_n$ by equating powers of x. This leads to a system in the unknowns $c_0, c_1, \ldots, c_n$. Develop its general form. For $a = 2, n = 4$ solve the particular linear system.

2.3 Gaussian Elimination and the *LU* Method

An elementary course in linear algebra usually offers Gaussian elimination and Cramer's rule as the two options for solving $Ax = f$. Numerically Cramer's rule is so costly that its use is restricted to a theoretical tool. Versions of Gausssian elimination, on the other hand, remain as applicable in today's computer environment as in the time of Gauss. The object of elimination algorithms is to reduce the coefficient matrix to triangular form. Minor extensions of the algorithms factor A, or a simply related matrix, into the product of lower and upper triangular matrices.

2.3.1 Gaussian Elimination

As the next few pages illustrate, the elimination procedure answers theoretical consistency and uniqueness questions assuming calculations are performed in exact arithmetic. Computationally this is insufficient; however, before addressing the added problems that occur due to machine implementation those underlying the algorithm take precedence.

When solving $Ax = f$ with Gaussian elimination f is augmented to A creating an $n \times (n+1)$ matrix $[A \vdots f]$. Suppose, for example, that the system is

$$\begin{array}{rcrcrcr} 2x & & & - & 3z & = & 3 \\ 4x & + & y & - & 5z & = & 7 \\ 2x & - & 3y & - & z & = & 5 \end{array}$$

then

$$[A \vdots f] = \begin{bmatrix} 2 & 0 & -3 & \vdots & 3 \\ 4 & 1 & -5 & \vdots & 7 \\ 2 & -3 & -1 & \vdots & 5 \end{bmatrix}.$$

Subtracting appropriate multiples of row one from rows two and three reduces $[A \vdots f]$ to

$$\begin{bmatrix} 2 & 0 & -3 & \vdots & 3 \\ 0 & 1 & 1 & \vdots & 1 \\ 0 & -3 & 2 & \vdots & 2 \end{bmatrix}.$$

The number in the $(1,1)$ position is called the first pivot. Performing one sweep of Gaussian elimination refers to the process of eliminating the elements below the pivot by adding multiples of the i-th row to rows $i+1, \ldots, n$. With respect to the above example, the sweep around the $(2,2)$ position is accomplished by adding 3 times row 2 to row 3. This yields the configuration

$$\begin{bmatrix} 2 & 0 & -3 & \vdots & 3 \\ 0 & 1 & 1 & \vdots & 1 \\ 0 & 0 & 5 & \vdots & 5 \end{bmatrix}.$$

Backward substitution gives $z = 1, y = 0$, and $x = 3$.

The method above may fail but in such a manner that only a minor revision is needed. As an example, the augmented matrix corresponding to

$$\begin{array}{rcrcrcr} x & + & y & + & z & = & 2 \\ 3x & + & 3y & + & z & = & 0 \\ 5x & + & 3y & + & z & = & 2 \end{array}$$

reduces to

$$\begin{bmatrix} 1 & 1 & 1 & \vdots & 2 \\ 3 & 3 & 1 & \vdots & 0 \\ 5 & 3 & 1 & \vdots & 2 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & \vdots & 2 \\ 0 & 0 & -2 & \vdots & -6 \\ 0 & -2 & -4 & \vdots & -8 \end{bmatrix}$$

after one pivot. The zero in the $(2,2)$ position raises a momentary worry but fortunately the order of the equations is theoretically irrelevant so that interchanging the second and third equations furnishes a quick fix. That is,

$$\begin{bmatrix} 1 & 1 & 1 & \vdots & 2 \\ 0 & 0 & -2 & \vdots & -6 \\ 0 & -2 & -4 & \vdots & -8 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & \vdots & 2 \\ 0 & -2 & -4 & \vdots & -8 \\ 0 & 0 & -2 & \vdots & -6 \end{bmatrix},$$

which yields $z = 3$, $y = -2$, and $x = 1$.

While the last difficulty admits a simple bookkeeping cure, other problems are without remedy under Gaussian elimination. Consider the system

$$\begin{array}{rcrcrcr} x & + & y & + & z & = & 6 \\ 2x & + & 2y & + & z & = & 11 \\ 4x & + & 4y & + & 3z & = & * \end{array}$$

where for a moment * is an unspecified constant. Proceeding with the elimination routine,

$$\left[\begin{array}{ccccc} 1 & 1 & 1 & \vdots & 6 \\ 2 & 2 & 1 & \vdots & 11 \\ 4 & 4 & 3 & \vdots & * \end{array}\right] \to \left[\begin{array}{ccccc} 1 & 1 & 1 & \vdots & 6 \\ 0 & 0 & -1 & \vdots & -1 \\ 0 & 0 & -1 & \vdots & ** \end{array}\right].$$

Notice that unlike the previous example, exchanging rows two and three does not introduce a nonzero pivot in the $(2, 2)$ position. At this stage most programmed algorithms flag an error that A is singular and the routine has failed. By hand it is possible to go the additional step

$$\left[\begin{array}{ccccc} 1 & 1 & 1 & \vdots & 6 \\ 0 & 0 & -1 & \vdots & -1 \\ 0 & 0 & -1 & \vdots & ** \end{array}\right] \left[\begin{array}{ccccc} 1 & 1 & 1 & \vdots & 6 \\ 0 & 0 & -1 & \vdots & -1 \\ 0 & 0 & 0 & \vdots & *** \end{array}\right].$$

Two cases occur. In the first, $* = 23$, $*** = 0$, and the system is consistent with solutions $z = 1$, $x + y = 5$. Setting $y = t$ the solutions expressed parametrically are

$$x = 5 - t, y = t, z = 1$$

which is the equation of a line in $\mathbb{R}^3$. If $***$ is nonzero then the system is inconsistent. We may interpret this in two ways: either $[6, 11, *]^T \notin \text{span}\{[1, 2, 4]^T, [1, 1, 3]^T\}$ or the three planes represented by the equations do not intersect in a common point. Unfortunately, a programmed elimination routine cannot perform this analysis and hand computation for large systems is not feasible. Singular value decomposition is an option discussed in Section 2.10.

2.3.2 The LU Method

For the remainder of this section A is assumed nonsingular. The LU method may be described via matrices as follows. In order to solve

$$Ax = f \tag{2.3.1}$$

we first set

$$A = LU \tag{2.3.2}$$

where L and U are lower and upper triangular matrices, respectively. Substituting (2.3.2) into (2.3.1) gives

$$LUx = f. \tag{2.3.3}$$

Now setting $y = Ux$ we solve the lower triangular system

$$Ly = f \tag{2.3.4}$$

with forward substitution then follow up by solving

$$Ux = y \tag{2.3.5}$$

with backward substitution. If neither $d(L)$ nor $d(U)$ vanishes then (2.3.4) and (2.3.5) have unique solutions.

The LU method's chief asset is readily seen when solutions for several right hand sides are sought. Given that the LU decomposition is known, finding x for each f uses $O(n^2)$ operations (see Problem 2.3.4) versus the $O(n^3)$ operations associated with Gaussian elimination. Of course, the start up cost of factoring A must be considered. Further, as yet an unanswered concern is whether or not A admits a factorization.

The answer to this last question is elegantly simple when A reduces to upper triangular form without row interchanges. In this case we may explain elimination via a series of elementary matrix operations and in so doing immediately derive the LU decomposition. The form of those operations is given by the following.

Definition 2.3.1 *The operation $E(\alpha; k, l)$ replaces row l of the $n \times n$ matrix A by row $k + \alpha$ row l. As an $n \times n$ matrix $E(\alpha; k, l)$ is given by*

$$E(\alpha; k, l) = \begin{cases} 1 & i = j \\ \alpha & i = k, j = l \\ 0 & \text{otherwise} \end{cases}.$$

Let us suppose that A has undergone $p-1$ steps of Gaussian elimination so that its configuration is

$$A_{p-1} = \begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots & \cdots & \cdots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \cdots & \cdots & \cdots & a_{2,n}^{(2)} \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & a_{p,p}^{(p)} & \cdots & a_{p,n}^{(p)} \\ \vdots & & \vdots & a_{p+1,p}^{(p)} & \cdots & a_{p+1,n}^{(p)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{n,p}^{(p)} & \cdots & a_{n,n}^{(p)} \end{bmatrix}$$

where the notation $a_{i,j}^{(k)}$ stands for the element replacing $a_{i,j}$ after $k-1$ elimination steps. Since we are assuming $a_{p,p}^{(p)} \neq 0$ we can define the multipliers

$$m_{i,p} = \frac{a_{i,p}^{(p)}}{a_{p,p}^{(p)}}\ , i = p+1, \ldots, n.$$

Notice that operating on A_{p-1} with $E(-m_{p+1,p}; p+1, p)$ reduces the element in the $(p+1, p)$ position to zero; that is,

$$E(-m_{p+1,p}; p+1, p)A_{p-1} = \begin{bmatrix} a_{1,1}^{(1)} & \cdots & \cdots & \cdots & \cdots & a_{1,n}^{(1)} \\ & \ddots & & & & \vdots \\ & & a_{p,p}^{(p)} & a_{p,p+1}^{(p)} & \cdots & a_{p,n}^{(p)} \\ & & 0 & a_{p+1,p+1}^{(p+1)} & \cdots & a_{p+1,n}^{(p+1)} \\ & & a_{p+2,p}^{(p)} & a_{p+2,p+1}^{(p)} & \cdots & a_{p+2,n}^{(p)} \\ & & \vdots & & & \vdots \\ & & a_{n,p}^{(p)} & \cdots & \cdots & a_{n,n}^{(p)} \end{bmatrix}.$$

Applying this concept to the remaining rows $p+2, \ldots, n$, the p-th elimination step is defined by

$$A_p = E_p A_{p-1}$$

where

$$E_p = E(-m_{n,p}; n, p) \cdots E(-m_{p+1,p}; p+1, p) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & -m_{p+1,p} & 1 & & & \\ & & -m_{p+2,p} & & \ddots & & \\ & & \vdots & & & \ddots & \\ & & -m_{n,p} & & & & 1 \end{bmatrix}. \tag{2.3.6}$$

Example 2.3.2 Suppose that

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ -3 & -17 & 1 & 2 \\ 4 & -17 & 8 & -5 \\ 0 & -5 & -2 & 1 \end{bmatrix}$$

then

$$\begin{aligned} A_3 &= E_3 E_2 E_1 A \\ &= E_3 E_2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ -3 & -17 & 1 & 2 \\ 4 & -17 & 8 & -5 \\ 0 & -5 & 2 & 1 \end{bmatrix} \\ &= E_3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3/2 & 1 & 0 \\ 0 & -5/14 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & -14 & 4 & 2 \\ 0 & -21 & 4 & -5 \\ 0 & -5 & -2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -12/7 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & -14 & 4 & 2 \\ 0 & 0 & -2 & -8 \\ 0 & 0 & -24/7 & 2/7 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & -14 & 4 & 2 \\ 0 & 0 & -2 & -8 \\ 0 & 0 & 0 & 14 \end{bmatrix}. \end{aligned}$$

Clearly the matrices E_p, $p = 1, 2, \ldots, n-1$, retain pivoting information.

Theorem 2.3.3 *If Gaussian elimination requires no row exchanges then the nonsingular matrix A reduces to upper triangular form via $E_{n-1} \cdots E_2 E_1 A = U$ where the diagonal elements of U are precisely the pivots.*

We next turn our attention to constructing a lower triangular matrix L such that $A = LU$. A direct consequence of Theorem 2.3.3 is

$$\begin{aligned} A &= (E_{n-1}E_{n-2}\cdots E_1)^{-1}U \\ &= E_1^{-1}E_2^{-1}\cdots E_{n-1}^{-1}U \end{aligned}$$

Theorem 2.3.4 *If E_p is as in (2.3.6) then*

$$E_p^{-1} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{p+1,p} & 1 & & \\ & & m_{p+2,p} & & \ddots & \\ & & \vdots & & & \ddots \\ & & m_{n,p} & & & & 1 \end{bmatrix}.$$

Theorem 2.3.5 *The product $E_1^{-1}E_2^{-1}\cdots E_{n-1}^{-1}$ is the lower triangular matrix L where*

$$L = \begin{bmatrix} 1 & & & & & \\ m_{2,1} & 1 & & & & \\ m_{3,1} & m_{3,2} & 1 & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ \vdots & \vdots & & \ddots & \ddots & \\ m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,n-1} & 1 \end{bmatrix}.$$

Notice that (2.3.2) may be regarded as a system of n^2 equations with n^2+n unknowns; however, if L must satisfy $l_{i,i} = 1, i = 1, 2, \ldots, n$, then the number of unkowns is reduced to n^2. Hence $A = (E_{n-1}\cdots E_2E_1)^{-1}U = LU$ is a unique factorization under the conditions stated throughout.

As a practical matter we may store both L and U by overwriting A. Thus when no row interchanges are necessary the final array stored in A is

$$A = \begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots & \cdots & a_{1,n}^{(1)} \\ m_{2,1} & a_{2,2}^{(2)} & \cdots & \cdots & a_{2,n}^{(2)} \\ m_{3,1} & m_{3,2} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ m_{n,1} & \cdots & \cdots & m_{n,n-1} & a_{n,n}^{(n-1)} \end{bmatrix}.$$

The upper triangular portion of A is U while L corresponds to the lower portion of A with 1's substituted on the main diagonal.

Example 2.3.6 For A as in Example 2.3.1??,

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 4 & 3/2 & 1 & 0 \\ 0 & 5/14 & 12/7 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & -14 & 4 & 2 \\ 0 & 0 & -2 & -8 \\ 0 & 0 & 0 & 14 \end{bmatrix}$$

which would be stored as

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ -3 & -14 & 4 & 2 \\ 4 & 3/2 & -2 & -8 \\ 0 & 5/14 & 12/7 & 14 \end{bmatrix}.$$

Problems

2.3.1 If A is $n \times n$ write out an algorithm to solve $Ax = f$ using Gaussian elimination. Assume that row interchanges may be necessary.

2.3.2 Determine the number of multiplications/divisions and additions/subtractions needed to solve $Ax = f$ using Gaussian elimination.

2.3.3 An $n \times n$ matrix A is called tridiagonal if $a_{i,j} = 0$ whenever $|i-j| > 1$. The form of A is

$$\begin{bmatrix} a_1 & b_1 & & & & \\ c_2 & a_2 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{n-1} & a_{n-1} & b_{n-1} \\ & & & & c_n & a_n \end{bmatrix}.$$

Modify your Gaussian elimination algorithm of Problem 2.3.1 to solve $Ax = f$. Your algorithm should take advantage of storage savings. What is the algorithms operation count?

2.3.4 Prove Theorem 2.3.4 .

2.3.5 Prove Theorem 2.3.5 .

2.3.6 Given that A is of order n and that $A = LU$ is known, calculate the number of multiplications/divisions and additions/subtractions needed to solve $Ax = f$.

2.3.7 Find the value of * for which

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & -1 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ * \end{bmatrix}$$

has a solution. Illustrate the solution geometrically.

2.3.8 For

$$A = \begin{bmatrix} \omega & 3 & 1 \\ 2 & -1 & 2\omega \\ 1 & 4 & \omega \end{bmatrix}, \ f = \begin{bmatrix} 5 \\ 3 \\ 6 \end{bmatrix}$$

the system $Ax = f$ has a unique solution when $\omega \neq \pm 1$, no solution when $\omega = -1$, and infinitely many solutions when $\omega = 1$. Show this.

2.3.9 Factor A into LU and solve the system

$$Ax = \begin{bmatrix} 2 & 3 & 3 \\ 0 & 5 & 7 \\ 6 & 9 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}.$$

2.4 The LU Method with Row Interchanges

In the last section we saw that when no row exhanges are needed solving $Ax = f$ with the LU method is equivalent to Gaussian elimination. However, if A is nonsingular but a zero pivot occurs then at first glance factorization appears to fail and we must resort to elimination. In this section we adapt the LU method to permit row interchanges, not only for the purpose of circumventing zero pivots but also, as experience has shown, as a means of combatting round off error. A major difference between the theoretical versus numerical solution of a linear system is that the ordering of the equations and, indeed, even the ordering of the unknowns is significant.

Example 2.4.1 If

$$A = \begin{bmatrix} 1.10 & 0.50 & 1.00 \\ 2.40 & 1.10 & 3.00 \\ -1.00 & 0.00 & 1.00 \end{bmatrix}$$

then without row interchanges

$$L = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 2.18 & 1.00 & 0.00 \\ -0.91 & 50.00 & 1.00 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 1.10 & 0.50 & 1.00 \\ 0.00 & 0.01 & 0.82 \\ 0.00 & 0.00 & -39.00 \end{bmatrix}.$$

Solving $Ax = b$ where $b = [-.65, -3.12, -2.60]^T$ yields $\hat{x} = [.41, 2.00, -2.10]^T$ while the true solution is $x = [0.50, 1.80, -2.10]^T$. Exchanging equations two and three alters the coefficient matrix to

$$\begin{aligned} B &= \begin{bmatrix} 1.10 & 0.50 & 1.00 \\ -1.00 & 0.00 & 1.00 \\ 2.40 & 1.10 & 3.00 \end{bmatrix} \\ &= \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ -0.91 & 1.00 & 0.00 \\ 2.18 & 0.02 & 1.00 \end{bmatrix} \begin{bmatrix} 1.10 & 0.50 & 1.00 \\ 0.00 & 0.45 & 1.91 \\ 0.00 & 0.00 & 0.78 \end{bmatrix}. \end{aligned}$$

Now solving $Bx = c$ where $c = [-.65, -2.60, -3.12]^T$, $\hat{x} = [.49, 1.82, -2.10]^T$.

The previous example illustrates that factoring A into LU and then solving $LUx = b$ is less accurate than starting with $Bx = c$. Notice that the nonzero off diagonal entries of U are of the same order of magnitude but the diagonal elements vary by four orders of magnitude. This discrepancy also shows up in the nonzero elements of L making the situation ripe for round off errors. On the other hand the LU decomposition of B indicates almost no variation in orders of magnitude. Pivoting strategies, that is the selective interchange of rows and columns, seek to minimize magnitude differences.

Glancing at the matrix A is Example 2.4.1 there is no particular reason to suspect problems beforehand. Only during elimination does the difficulty present itself. It is only practical to pivot when the need appears at each elimination step. Therefore we need a means to keep track of pivoting. For instance the matrix B in Example 2.4.1 is simply related to A by

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} A.$$

The matrix relating the two is a permutation matrix.

Definition 2.4.2 *A permutation matrix is a square matrix P with exactly one unit in each row and column, the remaining elements being zero. The* *notation*

$$P = \begin{bmatrix} 1 & 2 & 3 & \cdots & n \\ j_1 & j_2 & j_3 & \cdots & j_n \end{bmatrix} \tag{2.4.1}$$

is used where $(j_1, j_2, \ldots, j_n)$ *is a permutation of the integers* $(1, 2, \ldots, n)$, *and where each* (i, j_i) *element of* P *is* 1, *the remaining positions being zero.*

Example 2.4.3 The matrix

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

is a permutation matrix of order 4. In the notation of (2.4.1)

$$P = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{bmatrix}.$$

The operation PA of the permutation matrix P on a square matrix A interchanges rows of A while the operation AP interchanges columns of A. For example, if A is 4 x 4 and P is the matrix in Example 2.4.3 then PA simultaneously transforms

$$\begin{array}{lcl} \text{row } 1 & \longrightarrow & \text{row } 3 \\ \text{row } 2 & \longrightarrow & \text{row } 1 \\ \text{row } 3 & \longrightarrow & \text{row } 4 \\ \text{row } 4 & \longrightarrow & \text{row } 2. \end{array}$$

Similarly AP takes

$$\begin{array}{lcl} \text{column } 1 & \longrightarrow & \text{column } 3 \\ \text{column } 2 & \longrightarrow & \text{column } 1 \\ \text{column } 3 & \longrightarrow & \text{column } 4 \\ \text{column } 4 & \longrightarrow & \text{column } 2. \end{array}$$

Curiously, for the same P the operations $P^T A$ and AP^T, respectively, interchange

$$\begin{array}{lcl} \text{row } 1 & \longrightarrow & \text{row } 2 \\ \text{row } 2 & \longrightarrow & \text{row } 4 \\ \text{row } 3 & \longrightarrow & \text{row } 1 \\ \text{row } 4 & \longrightarrow & \text{row } 3 \end{array}$$

and

$$\begin{array}{lcl} \text{column } 1 & \longrightarrow & \text{column } 2 \\ \text{column } 2 & \longrightarrow & \text{column } 4 \\ \text{column } 3 & \longrightarrow & \text{column } 1 \\ \text{column } 4 & \longrightarrow & \text{column } 3 \end{array}$$

hence P^T reverses the operation P. Indeed, it is trivial to verify that for a square matrix A, $A = P^T PA = PP^T A = APP^T = AP^T P$.

Theorem 2.4.4 *Let P be a permutation matrix. Then*

$$P^T P = PP^T = I. \tag{2.4.2}$$

The proof is left as an exercise.

Decomposition methods that allow for row interchanges rest on the following.

Theorem 2.4.5 *If A is nonsingular then there exists a permutation matrix P such that*

$$PA = LU. \tag{2.4.3}$$

Ignoring for a moment the actual construction of P, L and U, let us rewrite $Ax = f$ as

$$PAx = LUx = Pf. \tag{2.4.4}$$

Once P is known (2.4.4) admits immediate solution via forward and backward substitution.

Our goal is to determine a P such that stable numerical calculations ensue from the LU decomposition of PA. As a side benefit zero pivots are managed naturally in the process. An easily implemented, well accepted pivoting strategy is scaled column pivoting.

To describe the algorithm we begin by assuming the $n \times n$ matrix A has undergone $k-1$ elimination steps and that the multipliers are stored in the portion of A that has been zeroed out. Therefore, the present configuration of A is

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & \cdots & a_{1,n} \\ m_{2,1} & a_{2,2} & & & & \vdots \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & m_{k,k-1} & a_{k,k} & \cdots & a_{k,n} \\ \vdots & & \vdots & a_{k+1,k} & \cdots & a_{k+1,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ m_{n,1} & \cdots & m_{n,k-1} & a_{n,k} & \cdots & a_{n,n} \end{bmatrix}$$

where it is understood that elements of A are overwritten as we proceed. For $i = k, k+1, \ldots, n$ let

$$d_i^{(k)} = \max_{k \leq j \leq n} |a_{i,j}|.$$

Scaled column pivoting selects the pivot of maximum magnitude with respect to the element of maximum absolute value in its row, in other words the first value of $i = k, k+1, \ldots, n$, for which

$$\frac{|a_{i,k}|}{d_i^{(k)}}$$

is largest. Denote this row by i_k. The next step of the algorithm exchanges rows k and i_k of A, including those columns containing the multipliers. Now continue as before.

The final configuration of A is the LU decomposition of a matrix simply related to A by a permutation matrix. As in the last section

$$U = \begin{cases} a_{i,j} & i \leq j \\ 0 & \text{otherwise} \end{cases} \tag{2.4.5}$$

and

$$L = \begin{cases} a_{i,j} & i > j \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} . \tag{2.4.6}$$

It remains to establish P.

Definition 2.4.6 *Let $P(i,j)$ denote the permutation matrix obtained from I by interchanging the i-th and j-th rows of I.*

Theorem 2.4.7 *The operation $P(i,j)A$ interchanges rows i and j of A. The operation $AP(i,j)$ interchanges columns i and j of A.*

As may be expected, a general permutation matrix P is related to the operations $P(i,j)$.

Theorem 2.4.8 *Every permutation matrix P can be expressed as a finite product of elementary operations $P(i,j)$.*

Returning to our algorithm, denote $P(k,i_k)$ as the permutation matrix that exchanges rows k and $i_k \geq k$ prior to carrying out the kth elimination step on A.

Theorem 2.4.9 *If $P = P(n-1,i_{n-1})P(n-2,i_{n-2})\cdots P(1,i_1)$ then $PA = LU$ where L and U are given by (2.4.6) and (2.4.5), respectively.*

Example 2.4.10 Suppose

$$A = \begin{bmatrix} 2 & -2 & 4 \\ 4 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Then under scaled column pivoting we exchange rows 1 and 2 and proceed with the first elimination step. At its completion the matrix stored in A is

$$\begin{bmatrix} 4 & 1 & 1 \\ 1/2 & -5/2 & 7/2 \\ 1/4 & -1/4 & -1/4 \end{bmatrix}.$$

Comparing 5/7 and 1/1, scaled column pivoting once again dictates row exchanges, this time rows 2 and 3. The final array is

$$\begin{bmatrix} 4 & 1 & 1 \\ 1/4 & -1/4 & -1/4 \\ 1/2 & 10 & 6 \end{bmatrix}.$$

Therefore

$$PA = LU$$

where

$$\begin{aligned} P &= P(2,3)P(1,2) \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \\ L &= \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/2 & 10 & 1 \end{bmatrix}, \end{aligned}$$

and

$$U = \begin{bmatrix} 4 & 1 & 1 \\ 0 & -1/4 & 1/4 \\ 0 & 0 & 6 \end{bmatrix}.$$

To solve

$$Ax = \begin{bmatrix} -2 \\ 6 \\ 1 \end{bmatrix},$$

let

$$g = P(2,3)P(1,2) \begin{bmatrix} -2 \\ 6 \\ 1 \end{bmatrix} = P(2,3) \begin{bmatrix} 6 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ -2 \end{bmatrix}$$

so that

$$LUx = g .$$

Now apply forward and backward substitution

A second, more costly, pivoting strategy is maximal pivoting. Before the k-th elimination step select an element

$$d^{(k)} = \max_{k \le i,j \le n} |a_{i,j}|$$

then using row and column interchanges move the element to the (k, k) position. Proceed as usual. This strategy should be reserved for systems the user suspects are particularly susceptible to round off error, such as those whose elements differ by several orders of magnitude.

For general linear systems it is virtually impossible to state a priori how much more accurate a numerical solution is when computed with a pivoting routine. There are, however, conditions under which it is known that pivoting does not significantly improve numerical approximations.

Definition 2.4.11 *The $n \times n$ matrix A is said to be strictly row diagonally dominant if for all $i = 1, 2, \ldots, n$*

$$|a_{i,i}| > \sum_{j=1, j \ne i}^{n} |a_{i,j}|.$$

Definition 2.4.12 *The $n \times n$ matrix A is strictly column diagonally dominant if for all $j = 1, 2, \ldots, n$*

$$|a_{j,j}| > \sum_{i=1, i \ne j}^{n} |a_{i,j}|.$$

Theorem 2.4.13 *If the $n \times n$ matrix A is either strictly row or column diagonally dominant then A is nonsingular. Moreover an LU decomposition of A based on Gaussian elimination can be carried out without row exchanges and is stable with respect to the growth of rounding errors.*

Definition 2.4.14 *The $n \times n$ symmetric matrix A is called positive definite if for all $x \in \mathbb{R}^n$ such that $x \neq 0$ $x^T Ax > 0$.*

Theorem 2.4.15 *If the $n \times n$ matrix A is positive definite then its eigenvalues are real and positive. Moreover A admits an LU decomposition via Gaussian elimination without row interchanges and this decomposition is stable with respect to rounding errors.*

Problems

2.4.1 Prove Theorem 2.4.4.

2.4.2 Prove Theorem 2.4.7.

2.4.3 Prove that the eigenvalues of a positive definite matrix are positive.

2.4.4 Find the LU decomposition of

$$A = \begin{bmatrix} 1 & 4 & 2 \\ -2 & -8 & 3 \\ 0 & 1 & 1 \end{bmatrix}$$

using scaled column pivoting. Solve the system $Ax = f$ where $f = [-2, 32, 1]^T$.

2.4.5 Show that the $n \times n$ positive definite matrix A satisfies the following:

(i) $a_{ii} > 0$ for $i = 1, 2, \ldots, n$

(ii) $\max_{1 \le i \le n} |a_{i,i}| \ge \max_{1 \le i,j \le n} |a_{i,j}|$

(iii) $(a_{i,j})^2 < a_{i,i} a_{j,j}$ for $i, j = 1, 2, \ldots, n$; $i \neq j$

2.4.6 Show that

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

is positive definite.

2.5 Vector and Matrix Norms

Solving $Ax = b$ in finite precision arithmetic yields a computed solution $\hat{x}$ that must be regarded as an approximate to the true solution x. We can further cloud the picture by noting that round off may not permit exact representation of A and b; hence, most often we are solving a perturbed version of the true system. Clearly we need tools to measure the error $e = x - \hat{x}$ before we can approach the problem of ensuring that the error is "small". In this section we introduce the concept of a norm in $\mathbb{R}^n$ as a measure of vector length.

Definition 2.5.1 *A norm in $\mathbb{R}^n$ is a function, $\|\cdot\|$, from $\mathbb{R}^n$ to the nonnegative real numbers such that for all x,$y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$*

(i) $\|x\| \geq 0$ *and* $\|x\| = 0 \Leftrightarrow x = 0$

(ii) $\|\alpha x\| = |\alpha|\|x\|$

(iii) $\|x + y\| \leq \|x\| + \|y\|$.

Criterion (i) implies that the length of a nonzero vector in $\mathbb{R}^n$ is always positive, (ii) guarantees that the vectors x and $-x$ have the same length, and (iii) is the familiar triangle inequality. It follows that a natural measure of the distance between two vectors x and y is $\|x - y\|$.

Although an infinity of norms exist we confine our discussion to the ℓ_1 (taxicab), ℓ_2 (Euclidean), and ℓ_∞ (max) norms.

Definition 2.5.2 *For $x = [x_1, x_2, \ldots, x_n]^T$ the ℓ_1, ℓ_2, and ℓ_∞ norms are denoted by $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ where*

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \ ,$$

$$\|x\|_2 = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2} ,$$

and

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \ .$$

The proofs that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ satisfy the definition of a norm is left as an exercise. Since the proof for $\|\cdot\|_2$ differs somewhat we include it here.

Theorem 2.5.3 *The function $\|\cdot\|_2$ is a vector norm in $\mathbb{R}^n$.*

Proof: Clearly (i) is satisfied while (ii) follows from

$$\|\alpha x\|_2 = \left\{\sum_{i=1}^{n}(\alpha x_i)^2\right\}^{1/2} = \left\{\alpha^2 \sum_{i=1}^{n} x_i^2\right\}^{1/2} = |\alpha|\left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} = |\alpha|\|x\|_2 \ .$$

To establish (iii) we rely on the Cauchy-Schwarz inequality

$$\sum_{i=1}^{n} |x_i y_i| \le \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \left\{\sum_{i=1}^{n} y_i^2\right\}^{1/2} \tag{2.5.1}$$

that the reader is asked to verify in Problem 2.5.3. Proceeding under the assumption that the Cauchy-Schwarz inequality holds we have

$$\begin{aligned} \|x+y\|_x^2 &= \sum_{i=1}^{n} x_i^2 + 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\ &\le \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2 \\ &\le (\|x\|_2 + \|y\|_2)^2 \ . \end{aligned}$$

Taking square roots concludes the proof.

Example 2.5.4 For $x = [2, -5, 3/2]^T$, $\|x\|_1 = 17/2$, $\|x\|_2 = 5\sqrt{5}/2$, and $\|x\|_\infty = 5$.

At this juncture we pause to offer geometric interpretations of the ℓ_1 and ℓ_2 unit speres in $\mathbb{R}^2$, that is the values $x \in \mathbb{R}^2$ such that $\|x\|_p = 1$ for $p = 1, 2$. If $x = [x, y]^T$ then the Euclidean unit sphere is $\{(x, y) : x^2 + y^2 = 1\}$, a circle of radius 1 centered at the origin (see Figure 2.5 below). The ℓ_1 sphere follows almost as rapidly in we note that

$$\begin{aligned} \{(x,y) : |x| + |y| = 1\} &= \{(x,y) : \text{ for } x \ge 0, y \ge 0, x + y = 1\} \\ &\quad \cup \{(x,y) : \text{ for } x > 0, y < 0, x - y = 1\} \\ &\quad \cup \{(x,y) : \text{ for } x < 0, y > 0, -x + y = 1\} \\ &\quad \cup \{(x,y) : \text{ for } x < 0, y < 0, x + y = -1\} \ . \end{aligned}$$

Figure 2.5 illustrates the diamond shaped ℓ_1 sphere.

Example 2.5.4 and Figure 2.5 clearly show that barring a few exceptional vectors the choice of norm leads to different values of $\|x\|$. If we are trying to control $\|e\|$ where e is some error this variation between norms could be disturbing however the succeeding definition and theorem imply that in essence we can pick a norm to suit our convenience.

Figure 2.1: The ℓ_2 and ℓ_1 unit spheres

Definition 2.5.5 *The norms $\|\cdot\|$ and $\|\cdot\|'$ are called equivalent if there exist constants α and β such that for all $x \in \mathbb{R}^n$*

$$\alpha\|x\| \leq \|x\|' \leq \beta\|x\| \ .$$

Equivalence of norms is reflexive in the sense that the statement above implies there exist constants γ and θ such that

$$\gamma\|x\|' \leq \|x\| \leq \theta\|x\|' \ .$$

Theorem 2.5.6 *The ℓ_1, ℓ_2, and ℓ_∞ norms are pairwise equivalent.*

Proof: See ?.

Having defined equivalent norms it is now convenient to mention convergent sequences, a central concept for the algorithms included in the next two sections.

Definition 2.5.7 *The sequence $\left\{x^{(k)}\right\}_{k=1}^{\infty}$ in $\mathbb{R}^n$ is said to converge to x in norm $\|\cdot\|$ if for all $\varepsilon > 0$ there exists $N(\varepsilon)$ such that*

$$\|x^{(k)} - x\| < \varepsilon \text{ for all } k > N(\varepsilon).$$

Theorem 2.5.8 *If the vector sequence $x^{(k)} \to x$ in any one of the norms ℓ_1, ℓ_2, or ℓ_∞ then it converges to the identical limit in the remaining norms.*

Proof: By equivalence of norms.

Example 2.5.9 The sequence $x^{(k)} = [e^{2-k}, -2/k^2, 1/k]^T$ converges to $x = [0,0,0]^T$.

Besides measuring the length of a vector in $\mathbb{R}^n$ we need some measure of size for an $n \times n$ matrix. At the outset a matrix norm apears very similar to a vector norm with one added constraint.

Definition 2.5.10 *Let A and B be real $n \times n$ matrices and $\alpha \in \mathbb{R}$. The real valued function $\|\cdot\|$ is said to be a matrix norm if $\|\cdot\|$ satisfies the following*

(i) $\|A\| \geq 0$ *and* $\|A\| = 0 \Leftrightarrow A = 0$

(ii) $\|\alpha A\| = |\alpha|\|A\|$

(iii) $\|A + B\| \leq \|A\| + \|B\|$

(iv) $\|AB\| \leq \|A\|\|B\|$.

One standard measuring $\|A\|$ is by comparison of the size of Ax with x:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} . \qquad (2.5.2)$$

This is called an induced or natural norm. An inequality immediately follows for induced norms

$$\|Ax\| \leq \|A\|\|x\| .$$

It can be shown that (2.5.2) is equivalent to

$$\|A\| = \max_{\|x\|=1} \|Ax\| . \qquad (2.5.3)$$

Fortunately, calculating $\|A\|_p$ is fairly straightforward when $p = 1, \infty$.

Theorem 2.5.11 $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}|$.

Proof: The proof follows in two parts. First, let $x \in \mathbb{R}^n$ be such that $\|x\|_1 = 1$. Then

$$\begin{aligned} \|Ax\|_1 &= \sum_{i=1}^{n} |(Ax)_i| \\ &= \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{i,j} x_j| \end{aligned}$$

$$\begin{aligned} &\leq \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{i,j}| \\ &\leq \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}| \end{aligned}$$

hence

$$\max_{\|x\|_1 = 1} \|Ax\|_1 \leq \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}| \; .$$

To establish the inequality in the other direction let m correspond to a column such that $\sum_{i=1}^{n} |a_{i,j}|$ is a maximum. Define the particular vector $y = [y_1, y_2, \ldots, y_n]^T$ by

$$y_j = \begin{cases} 0 & j \neq m \\ 1 & j = m \end{cases} \; .$$

Then

$$\begin{aligned} \max_{\|x\|_1 = 1} \|Ax\|_1 &\geq \|Ay\|_1 \\ &= \sum_{i=1}^{n} |a_{i,m}| \\ &= \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}| \end{aligned}$$

which completes the proof.

Theorem 2.5.12 $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{i=1}^{n} |a_{i,j}|$.

Proof: See Problem 2.5.6.

The ℓ_2 norm of a matrix A is seldom computed via (2.5.3) simply because it is too difficult. However, there is an intimate connection between the singular values of A and its ℓ_2 norm.

Definition 2.5.13 *The singular values of a real $n \times n$ matrix A are the eigenvalues of $A^T A$.*

Example 2.5.14 For

$$A = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 10 & -10 \\ -10 & 20 \end{bmatrix} .$$

Therefore, the singular values of A are the roots of $p(\lambda) = \lambda^2 - 30\lambda + 100$, or $\lambda = 15 \pm 5\sqrt{5}$.

Definition 2.5.15 *The spectral radius $\rho(A)$ of a matrix A is given by*

$$\rho(A) = \left\{ \max_{\lambda} |\lambda| \ : \ Ax = \lambda x, x \neq 0 \right\} .$$

Theorem 2.5.16 *If A is a real $n \times n$ matrix then*

(i) $\|A\|_2 = \sqrt{\rho(A^T A)}$.

(ii) $\rho(A) \leq \|A\|$ *for any induced norm* $\| \cdot \|$.

Proof: For a proof of (i) see ?.

(ii) Let λ be an eigenvalue of A with unit eigenvector x. Then

$$|\lambda| = \|\lambda x\| \leq \|A\|$$

hence

$$\rho(A) \leq \|A\| .$$

Problems

2.5.1 Show that $\| \cdot \|_1$ satisfies the definition of a norm.

2.5.2 Show that $\| \cdot \|_\infty$ satisfies the definition of a norm.

2.5.3 Prove the Cauchy-Schwarz inequality (see 2.5.1). Hint: evaluate $\|x - \lambda y\|^2$ and select a useful value of λ.

2.5.4 Find $\|x\|_1$, $\|x\|_2$, and $\|x\|_\infty$ for $x = [2\sin\theta, 2\cos\theta, 8]^T$.

2.5.5 Find $\lim_{k\to\infty} x^{(k)}$ where $x^{(k)} = [e^{1/k}, \sqrt{k^2 + k} - k, (\sin k)/k]^T$.

2.5.6 Prove Theorem 2.5.12 .

2.5.7 Calculate $\|A\|_1$, $\|A\|_2$, and $\|A\|_\infty$ for

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} .$$

2.5.8 If λ is an eigenvalue of the nonsingular matrix A show that for any natural norm $\|\cdot\|$, $(1/\|A^{-1}\|) \leq |\lambda|$.

2.5.9 Show that if A is symmetric then $\|A\|_2 = \max\{|\lambda|; Ax = \lambda x\}$.

2.6 Residual Refinement

A dilemma inherent in the numerical solution of linear systems is that the error $e = x - \hat{x}$ can only be determined if the true solution x is known. A sensible task is to estimate either $\|e\|$ or $\|e\|/\|x\|$. Most often the residual vector $r = b - A\hat{x}$ serves as a reliable indicator of e in the sense that $\|r\|/\|b\|$ and $\|e\|/\|x\|$ are within a few orders of magnitude of each other. Intuitively it seems quite reasonable that if the computed solution $\hat{x}$ closely satisfies the equation $Ax = b$ then $\hat{x}$ should closely approximate x. Unfortunately there are systems for which our intuition is at fault as the next example illustrates.

Example 2.6.1 Let

$$A = \begin{bmatrix} 1.001 & 1.999 \\ .999 & 2.001 \end{bmatrix}, \; b_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \; \text{and } b_2 = \begin{bmatrix} -.3 \\ .3 \end{bmatrix}.$$

First we consider the system $Ax = b_1$ and note that its true solution is $x = [1,1]^T$. If $\hat{x} = [1.001, .999]^T$ then $e = [.001, -.001]^T$ and $r = [-.001, -.001]^T$. Measuring with the ℓ_1 norm, $\|r\|/\|b\| = .0003$ and $\|e\|/\|x\| = .0005$. Now if for some bizarre reason $\hat{x} = [3,0]^T$ then $r = [.003, -.003]^T$, $e = [2, -1]^T$ and $\|r\|/\|b\| = .001$ while $\|e\|/\|x\| = .333$. Clearly a small relative residual does not imply a small relative error. Likewise, if we turn our attention to the system $Ax = b_2$, a large relative residual may occur with a small relative error. Here the true solution is $x = [-200, 100]^T$. If $\hat{x} = [-202, 99]^T$ then $\|r\|/\|b\| = 13.33$ while $\|e\|/\|x\| = .01$.

Depending on lady luck to produce better results than expected is unwise, hence, we need some means of deciding when it is safe to rely on the residual. For a system of order 2 it is possible to foresee problems graphically. The figure below shows two lines that are nearly parallel. For a goodly range of values (x_1, x_2) the lines remain close even though the point of intersection may be some distance away.

Figure 2.2: Nearly Parallel Lines

Potential difficulties with large systems are detected via norms. We begin with the fundamental relationship between residual and error

$$r = b - A\hat{x} = Ax - A\hat{x} = Ae \tag{2.6.1}$$

or equivalently

$$e = A^{-1}r \tag{2.6.2}$$

where we are assuming A^{-1} exists. Applying norm properties to (2.6.2) yields

$$\begin{aligned} \|e\| &\leq \|A^{-1}\|\|r\| \\ &= \|A^{-1}\|\|Ax\|\frac{\|r\|}{\|b\|} \\ &\leq \|A^{-1}\|\|A\|\|x\|\frac{\|r\|}{\|b\|} \end{aligned}$$

or,in terms of relative error,

$$\frac{\|e\|}{\|x\|} \leq \|A^{-1}\|\|A\|\frac{\|r\|}{\|b\|} \, .$$

Similarly we can bound the relative error from below as follows:

$$\|A\|\|e\| \geq \|Ae\| \geq \|r\|$$

which implies

$$\begin{aligned}\frac{\|e\|}{\|x\|} &\geq \frac{1}{\|x\|\|A\|}\|r\| \\ &= \frac{1}{\|A^{-1}b\|\|A\|}\|r\| \\ &\geq \frac{1}{\|A^{-1}\|\|A\|}\frac{\|r\|}{\|b\|} .\end{aligned}$$

Definition 2.6.2 *The condition number of a matrix A is given by*

$$cond(A) = \|A^{-1}\|\|A\| .$$

Combining the results above, we have

$$\frac{1}{\text{cond}(A)}\frac{\|r\|}{\|b\|} \leq \frac{\|e\|}{\|x\|} \leq \text{cond}(A)\frac{\|r\|}{\|b\|} . \tag{2.6.3}$$

By Problem 2.6.1, $\text{cond}(A) \geq 1$. When $\text{cond}(A)$ is near 1 the relative error is constrained to closely follow the relative residual while a large value of $\text{cond}(A)$, especially with respect to machine precision, allows for wide variation between relative error and residual.

Example 2.6.3 (Example 2.6.1 revisited) For

$$A = \begin{bmatrix} 1.001 & 1.999 \\ .999 & 2.001 \end{bmatrix}$$

the ℓ_1 norm, $\|A\|_1 = 4$, is small; however,

$$A^{-1} = \begin{bmatrix} 333.6 & -333.2 \\ -166.5 & 166.9 \end{bmatrix}$$

has a much larger ℓ_1 norm, $\|A^{-1}\|_1 = 500.1$, which combines to give $\text{cond}_1(A) = 2000$. For the right hand side b_1 and $\hat{x} = [1.001, .999]^T$ the relative error and residual are the same order of magnitude. When $\hat{x} = [3, 0]^T$ the relative error is roughly 300 times larger than the relative residual. Finally for b_2 and $\hat{x} = [-202, 99]^T$, $\|e\|/\|x\|$ is about 1/1000-th of $\|r\|/\|b\|$. The variation is clearly attributable to $\text{cond}_1(A)$.

Two items we wish to mention are first that the value of $\text{cond}(A)$ depends on the norm used and, second, that in practice $\text{cond}(A)$ is at best estimated since computing A^{-1} is costly. Even if we attempt to calculate A^{-1} it may be unreliable for precisely the same reason solutions of $Ax = b$ are. Estimates of cond(A) are given by **IMSL**, **MATLAB**, and **EISPACK** routines. We elaborate more shortly.

Iterative refinement is a technique applied to overcome numerical errors due to a poorly conditioned coefficient matrix. Consider the linear system

$$Ay = r \ . \tag{2.6.4}$$

If $r = 0$ then $y = 0$ and $\hat{x}$ is without error. If $r \neq 0$ then $y \neq 0$ and

$$y = A^{-1}r = x - \hat{x} \ .$$

Of course, we do not actually calculate y but rather $\hat{y}$. Nevertheless, $\hat{x} + \hat{y}$ should be an improved approximation to x.

Example 2.6.4 Consider solving the system

$$\begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -15 \\ 96 \\ -90 \end{bmatrix}$$

with two decimal rounding arithmetic. The LU decomposition of the coefficient matrix,

$$\begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix} = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ -4.00 & 1.00 & 0.00 \\ 3.33 & -1.25 & 1.00 \end{bmatrix} \begin{bmatrix} 9.00 & -36.00 & 30.00 \\ 0.00 & 48.00 & -60.00 \\ 0.00 & 0.00 & 5.10 \end{bmatrix}$$

leads to the approximate $\hat{x} = [2.94, 1.96, .97]^T$. The residual is $r = b - A\hat{x} = [0, .12, 0]^T$. Using the LU decomposition again, we solve $Ay = r$ and get $\hat{y} = [.06, .04, .03]^T$. Therefore, an updated approximate is $\hat{\hat{x}} = [3.00, 2.00, 1.00]^T$. We recognize this vector as the true solution.

When one step of iterative refinement is insufficient to produce a satisfactory result we merely define the iterative process

$$\hat{x}^{(k)} = \hat{x}^{(k-1)} + \hat{y}^{(k-1)}$$

where $\hat{y}^{(k-1)}$ is the numerical solution of

$$Ay = r^{(k-1)}$$

$$r^{(k-1)} = b - A\hat{x}^{(k-1)} \ .$$

In practice $\hat{y}^{(k)}$ is calculated using double precision arithmetic because its entries become successively smaller as the iteration continues. Some stopping criteria are necessary such as a limit on the maximum number of iterations or

$$\|\hat{x}^{(k)} - \hat{x}^{(k-1)}\| \leq \varepsilon$$

where ε is a user supplied tolerance.

Residual refinement tends to be costly and, as a result, is used only when the coefficient matrix is suspected of being ill-conditioned. The cost does not justify the expense for well-behaved matrices. This is true even if we perform an LU decomposition of A so that the solution of (2.6.4) is an $O(n^2)$ versus an $O(n^3)$ operational step.

We close with an estimate of cond(A). If t-digit arithmetic is used to perform a Gaussian type LU decomposition for A then ii can be shown [ref]

$$\mathrm{cond}(A) \approx \frac{\|\hat{y}\|}{\|\hat{x}\|} 10^t \ .$$

Problems

2.6.1 Show $\mathrm{cond}(A) \geq 1$ for any natural norm.

2.6.2 Show that if B is singular then

$$\frac{1}{\mathrm{cond}(A)} \leq \frac{\|A - B\|}{\|A\|}$$

2.6.3 The condition number also measures the sensitivity of A to round off error. Consider the system

$$\begin{bmatrix} 1 & 3 \\ 1.0001 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4.0001 \end{bmatrix}$$

whose solution is $x = [1, 1]^T$. If A is perturbed slightly so that we are solving

$$\begin{bmatrix} 1 & 3 \\ .9999 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} 4 \\ 4.0001 \end{bmatrix}$$

what is the solution now (use four digit rounding)? Comment on cond(A). Estimate cond(A) using the result of Problem 2.6.2.

2.6.4 Solve the system

$$\begin{array}{rcrcrcrcl} 6.99x_1 & + & 10.01x_2 & + & 6.99x_3 & + & 8.01x_4 & = & 32 \\ 5.01x_1 & + & 6.99x_2 & + & 5.01x_3 & + & 5.99x_4 & = & 23 \\ 5.99x_1 & + & 8.01x_2 & + & 8.99x_3 & + & 10.01x_4 & = & 33 \\ 5.01x_1 & + & 6.99x_2 & + & 10.01x_3 & + & 8.99x_4 & = & 31 \end{array}$$

with an LU decomposition truncating (not rounding) all calculations to two digits. Compare your results with the true solution $x_1 = x_2 = x_3 = x_4 = 1$. Perform one iteration of residual refinement. Estimate cond(A).

2.7 Iterative Methods

2.7.1 Motivation

It is not uncommon for a physical problem to give rise to extremely large sparse linear systems, say on an order of 10^3 to 10^6. For systems of this order, direct methods based on Gaussian elimination may require too much machine time and storage to be feasible. Further, a direct method may fill in many of the previously zero entries thereby creating more sources of round off error. Iterative methods are a viable alternative if the cost of each iteration is significantly less than elimination and if the sequence generated converges very rapidly.

Our first concern is motivating iterative methods and, hopefully, in the process indicating circumstances that guarantee the methods convergence.

Many linear systems arising in applications are naturally ordered so that the coefficient of x_k in the k-th equation is large in magnitude relative to the other coefficients in that equation. If this is true then rewriting

$$Ax = b$$

as

$$\begin{aligned}
x_1 &= \frac{1}{a_{1,1}}(b_1 - a_{1,2}x_2 - a_{1,3}x_3 - \cdots - a_{1,n}x_n) \\
x_2 &= \frac{1}{a_{2,2}}(b_2 - a_{2,1}x_1 - a_{2,3}x_3 - \cdots - a_{2,n}x_n) \\
&\vdots \\
x_n &= \frac{1}{a_{n,n}}(b_n - a_{n,1}x_1 - a_{n,2}x_2 - \cdots - a_{n,n-1}x_{n-1})
\end{aligned}$$

a good initial guess for x_i, $i = 1, 2, \ldots, n$, is

$$x_i^{(0)} = \frac{1}{a_{i,i}}b_i \ .$$

In vector form

$$x^{(0)} = D^{-1}b$$

where D is a diagonal matrix whose diagonal elements are those of A.

Next we must determine how to update an approximate $x^{(k-1)}$ to $x^{(k)}$. Residual refinement would dictate

$$x^{(k)} = x^{(k-1)} + y^{(k-1)}$$

where $Ay^{(k-1)} = r^{(k-1)}$. To avoid solving $Ay^{(k-1)} = r^{(k-1)}$ we approximate $y^{(k-1)}$ by $D^{-1}r^{(k-1)}$ which yields the iterative equation

$$x^{(k)} = x^{(k-1)} + D^{-1}r^{(k-1)}$$

or componentwise

$$x_i^{(k)} = x_i^{(k-1)} + \frac{1}{a_{i,i}} r_i^{(k-1)} . \tag{2.7.1}$$

There is some leeway in defining $r_i^{(k-1)}$. For instance, the most obvious choice is

$$r_i^{(k-1)} = b_i - \sum_{i=1}^{n} a_{i,j} x_j^{(k-1)}$$

which substituted into 2.7.1 gives

$$\begin{aligned} x_i^{(k)} &= x_i^{(k-1)} + \frac{1}{a_{i,i}} \left(b_i - \sum_{i=1}^{n} a_{i,j} x_j^{(k-1)} \right) && (2.7.2) \\ &= -\frac{1}{a_{i,i}} \sum_{j=1, j\neq i}^{n} a_{i,j} x_j^{(k-1)} + \frac{1}{a_{i,i}} b_i . && (2.7.3) \end{aligned}$$

If we let

$$L = \begin{bmatrix} 0 & & & & \\ -a_{2,1} & 0 & & & \\ -a_{3,1} & -a_{3,2} & 0 & & \\ \vdots & & \ddots & \ddots & \\ -a_{n,1} & -a_{n,2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 0 & -a_{1,2} & -a_{1,3} & \cdots & -a_{1,n} \\ & 0 & -a_{2,3} & \cdots & -a_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

then the matrix equation represented by 2.7.3 is

$$x^{(k)} = T_J x^{(k-1)} + c$$

where

$$T_J = D^{-1}(L + U)$$

and

$$c = D^{-1}b \ .$$

This is called the Jacobi iterative method.

A second choice for residual utilizes the known components of $x_i^{(k)}$ via

$$r_i^{(k-1)} = b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k)} - \sum_{j=i}^{n} a_{i,j}x_j^{(k-1)} \ . \tag{2.7.4}$$

Again we substitute into 2.7.1 and after simplifying find

$$x_i^{(k)} = -\frac{1}{a_{i,i}}\left(\sum_{j=1}^{i-1} a_{i,j}x_j^{(k)} + \sum_{j=i+1}^{n} a_{i,j}x_j^{(k-1)}\right) + \frac{1}{a_{i,i}}b_i$$

or

$$\sum_{j=1}^{i} a_{i,j}x_j^{(k)} = -\sum_{j=i+1}^{n} a_{i,j}x_j^{(k-1)} + b_i \ .$$

Letting $i = 1, 2, \ldots, n$ we have the matrix equation

$$(-L + D)x^{(k)} = Ux^{(k-1)} + b$$

or

$$x^{(k)} = T_G x^{(k-1)} + c$$

where

$$T_G = (D - L)^{-1}U$$

and

$$c = (D - L)^{-1}b \ .$$

The method outlined above is referred to as Gauss-Seidel iteration. While there is no general rule stating that Gauss-Seidel iteration converges faster than Jacobi iteration this is usually true.

A third class of iterative techniques are called relaxation methods. One such method uses the same form of the residual as Gauss-Seidel but the iterative equation,

$$x_i^{(k)} = x_i^{(k-1)} + \frac{\omega}{a_{i,i}} r_i^{(k-1)} \ , \tag{2.7.5}$$

now allows for a parameter ω whose purpose is to accelerate convergence. Rearranging terms in 2.7.5 gives

$$a_{i,i}x_i^{(k)} + \omega\sum_{j=1}^{i-1} a_{i,j}x_j^{(k)} = (1-\omega)a_{i,i}x_i^{(k-1)} - \omega\sum_{j=i+1}^{n} a_{i,j}x_i^{(k-1)} + \omega b_i$$

or, running through $i = 1, 2, \ldots, n$

$$x^{(k)} = T_\omega x^{(k-1)} + c$$

where

$$T_\omega = (D - \omega L)^{-1}((1-\omega)D + \omega U)$$

and

$$c = \omega(D - \omega L)^{-1} b \ .$$

If $\omega = 1$ the relaxation method is simply Gauss-Seidel. When $\omega < 1$ we are said to be under-relaxing, while $1 < \omega$ corresponds to over-relaxation. For $0 < \omega < 2$ we refer to the method as SOR for successive over-relaxation.

2.7.2 Convergence of Iterative Methods

While it may not be possible to establish a priori how fast an iterative method converges there are circumstances that at least guarantee convergence.

Definition 2.7.1 *An $n \times n$ matrix A is convergent if*

$$\lim_{k\to\infty} (A^k)_{i,j} = 0 \ \textit{for} \ 1 \le i, j \le n$$

where $(A^k)_{i,j}$ refers to the (i,j)-th element of A^k.

Theorem 2.7.2 *For an $n \times n$ matrix A the following are equivalent*

(i) A is convergent

(ii) $\lim_{n\to\infty} \|A^n\| = 0$ for some natural norm $\|\cdot\|$

(iii) $\rho(A) < 1$.

Lemma 2.7.3 *If for the $n \times n$ matrix A, $\rho(A) < 1$ then $(I - A)^{-1}$ exists and admits the convergent expansion*

$$(I - A)^{-1} = I + A + A^2 + \cdots$$

Theorem 2.7.4 *For any initial guess $x^{(0)}$ the sequence given by*

$$x^{(k)} = Tx^{(k-1)} + c \ , k = 1, 2, \ldots$$

where $c \neq 0$ converges to the unique solution of $x = Tx + c$ if and only if $\rho(T) < 1$.

Proof: Stepping back through the first $k-1$ iterates we have

$$x^{(k)} = T^k x^{(0)} + (T^{k-1} + \cdots + T + I)c \, . \tag{2.7.6}$$

If $\rho(T) < 1$ then taking the limit as $k \to \infty$ of 2.7.6 and applying the lemma gives

$$\begin{aligned}\lim_{k\to\infty} x^{(k)} &= \lim_{k\to\infty} T^k x^{(0)} + \lim_{k\to\infty} (T^{k-1} + \cdots + T + I)c \\ &= 0 + (I-T)^{-1}c \\ &= x \, .\end{aligned}$$

The proof of the converse is left as an exercise.

Not only does the spectral radius determine whether or not an iterative method converges it also provides a good estimate of how rapid convergence is.

Corollary 2.7.5 *Suppose that* $\|\cdot\|$ *is a natural norm. If* $\|T\| < 1$ *the sequence* $x^{(k)} = Tx^{(k-1)} + c$*,* $c \neq 0$*, converges to* x *for any initial guess* $x^{(0)}$*. Moreover,* $\|x - x^{(k)}\|$ *is bounded by*

$$\|x - x^{(k)}\| \leq \|T\|^k \|x - x^{(0)}\| \tag{2.7.7}$$

or

$$\|x - x^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|} \|x^{(1)} - x^{(0)}\| \, . \tag{2.7.8}$$

Proof: By definition

$$\begin{aligned}x - x^{(k)} &= Tx - Tx^{(k-1)} \\ &= T\left(x - x^{(k)}\right) + T\left(x^{(k)} - x^{(k-1)}\right)\end{aligned}$$

which implies

$$\begin{aligned}(I-T)(x - x^{(k)}) &= T\left(x^{(k)} - x^{(k-1)}\right) \\ &= T^k\left(x^{(1)} - x^{(0)}\right) \, .\end{aligned}$$

Taking norms

$$\begin{aligned}\|x - x^{(k)}\| &\leq \frac{\|T\|^k}{\|I - T\|} \|x^{(1)} - x^{(0)}\| \\ &\leq \frac{\|T\|^k}{1 - \|T\|} \|x^{(1)} - x^{(0)}\| \, .\end{aligned}$$

If we replace $\|T\|$ in Corollary 2.7.5 with $\rho(T)$ then

$$\|x - x^{(k)}\| \approx \rho(T)^k \|x - x^{(0)}\| \ .$$

Therefore, when $\rho(T)$ is close to 1 we expect slow convergence. In particular, we can estimate the number of iterations needed to obtain t digits of relative error by

$$\rho(T)^k \leq 10^{-t}$$

or, rather,

$$k \geq \frac{-t}{\log_{10} \rho(T)} \ .$$

The object of relaxation is to select the parameter ω so that the spectral radius of $T_\omega = (D - \omega L)^{-1}((1-\omega)D + \omega U)$ is a minimum. It is known [see?] that when $0 < \omega < 1$ it is possible to force convergence of systems for which Gauss-Seidel does not converge. When $1 < \omega < 2$ over-relaxations attempts to accelerate the convergence of Gauss-Seidel iteration.

Guaranteeing convergence of iterative methods based on properties of the iterative matrix T is all good and fine but sufficient conditions for A also exist. These tend to be easier to verify but are fairly restrictive.

Theorem 2.7.6 *If A is strictly diagonally dominant then for any initial guess $x^{(0)}$ the sequences defined by the Jacobi and Guass-Seidel iterative methods converge to the true solution of $Ax = b$.*

Before stating the analgous theorem for the SOR method the next theorem addresses the restriction $0 < \omega < 2$.

Theorem 2.7.7 *If the diagonal elements of A are nonzero then $\rho(T_\omega) \geq |\omega - 1|$;hence, a necessary condition of $\rho(T_\omega) < 1$ is $0 < \omega < 2$.*

Theorem 2.7.8 *If A is positive definite and $0 < \omega < 2$ then the SOR method converges to the true solution of $Ax = b$ for any initial guess $x^{(0)}$.*

Problems

2.7.1 Show that

$$A = \begin{bmatrix} 1 & 0 \\ \frac{1}{9} & \frac{1}{3} \end{bmatrix}$$

is not convergent but

$$B = \begin{bmatrix} \frac{1}{3} & 0 \\ 9 & \frac{1}{3} \end{bmatrix}$$

is convergent. In general, for a lower triangular matrix what is a necessary condition to guarantee convergence?

2.7.2 Show that if A is strictly diagonally dominant then $\|T_J\|_\infty < 1$.

2.7.3 Show that A is a convergent matrix if and only if $\lim_{k\to\infty} A^k x \to 0$ for every vector x.

2.7.4 Consider the system

$$\begin{bmatrix} 1.4 & 2.3 & 3.7 \\ 3.3 & 1.6 & 4.3 \\ 2.5 & 1.9 & 4.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6.5 \\ 10.3 \\ 8.8 \end{bmatrix} .$$

Comment on applying Gauss-Seidel iteration to the system. Experiment with SOR by varying the parameter ω.