Chapter 1

Triangular Factorization

This chapter deals with the factorization of arbitrary matrices into products of triangular matrices. Since the solution of a linear $n \times n$ system can be easily obtained once the matrix is factored into the product of triangular matrices, we will concentrate on the factorization of square matrices. Specifically, we will show that an arbitrary $n \times n$ matrix A has the factorization $PA = LU$ where P is an $n \times n$ permutation matrix, L is an $n \times n$ unit lower triangular matrix, and U is an $n \times n$ upper triangular matrix. In connection with this factorization we will discuss pivoting, *i.e.*, row interchange, strategies. We will also explore circumstances for which A may be factored in the forms $A = LU$ or $A = LL^T$. Our results for a square system will be given for a matrix with real elements but can easily be generalized for complex matrices.

The corresponding results for a general $m \times n$ matrix will be accumulated in Section 1.4. In the general case an arbitrary $m \times n$ matrix A has the factorization $PA = LU$ where P is an $m \times m$ permutation matrix, L is an $m \times m$ unit lower triangular matrix, and U is an $m \times n$ matrix having row echelon structure.

1.1 Permutation matrices and Gauss transformations

We begin by defining permutation matrices and examining the effect of premultiplying or postmultiplying a given matrix by such matrices. We then define Gauss transformations and show how they can be used to introduce zeros into a vector.

Definition 1.1 An $m \times m$ *permutation matrix* is a matrix whose columns consist of a rearrangement of the m unit vectors $\mathbf{e}^{(j)}$, $j = 1, \ldots, m$, in $\mathbf{R}^m$, *i.e.*, a rearrangement of the columns (or rows) of the $m \times m$ identity matrix.

The rows of an $n \times n$ permutation matrix consist of a rearrangement of the transposes $(\mathbf{e}^{(j)})^T$, $j = 1, \ldots, n$, of the n unit vectors in $\mathbf{R}^n$. The effect of premultiplying (postmultiplying) a matrix A by a permutation matrix P is to rearrange the rows (columns) of A into the same order as the rows (columns) of the identity matrix are ordered in P. For integers j and k such that $1 \leq j \leq k \leq m$, we denote by $P_{(j,k)}$ the special class of permutation matrices that result from the interchange

of the j-th and k-th columns of the identity matrix, *i.e.*,

$$P_{(j,k)} = \left(\mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(j-1)}, \mathbf{e}^{(k)}, \mathbf{e}^{(j+1)}, \cdots, \mathbf{e}^{(k-1)}, \mathbf{e}^{(j)}, \mathbf{e}^{(k+1)}, \cdots, \mathbf{e}^{(m)}\right) . \tag{1.1}$$

These permutation matrices, which are often referred to as *elementary permutation matrices*, have many useful properties such as

$$P_{(j,k)} = P_{(j,k)}^T = P_{(j,k)}^{-1} . \tag{1.2}$$

Of course, $P_{(j,j)} = I$.

We now define a special class of matrices that are rank one perturbations of the identity matrix and that can be used to introduce zeros into a vector.

Definition 1.2 For an integer j such that $1 \leq j < m$, let $\boldsymbol{\mu} \in \mathbf{R}^m$ such that $\mu_1 = \mu_2 = \cdots = \mu_j = 0$. The $m \times m$ matrix

$$M^{(j)} = I - \boldsymbol{\mu}(\mathbf{e}^{(j)})^T \tag{1.3}$$

is known as a *Gauss transformation*. Here I denotes the $m \times m$ identity matrix.

Clearly $M^{(j)}$ is a unit lower triangular matrix, *i.e.*, a lower triangular matrix with 1's on the diagonal. Also, off the main diagonal, $M^{(j)}$ has zero entries everywhere except in the j-th column. The following proposition shows that Gauss transformations can be used to introduce zeros into a vector. Specifically, for a given nonzero $\mathbf{x} \in \mathbf{R}^m$, it shows how to choose an integer p and a $\boldsymbol{\mu} \in \mathbf{R}^m$ such that $M^{(1)}P_{(1,p)}\mathbf{x}$ has a nonzero entry only in its first component.

Proposition 1.1 *Given* $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T \in \mathbf{R}^m$ *such that* $\mathbf{x} \neq \mathbf{0}$. *Choose any integer* p *such that* $1 \leq p \leq m$ *and* $x_p \neq 0$. *Define* $\boldsymbol{\mu} \in \mathbf{R}^m$ *by*

$$\mu_j = \begin{cases} 0 & \text{if } j = 1 \\ \dfrac{x_1}{x_p} & \text{if } j = p \text{ and } p \neq 1 \\ \dfrac{x_j}{x_p} & \text{if } j = 2, \ldots, m,\ j \neq p\,. \end{cases}$$

Then

$$M^{(1)}P_{(1,p)}\mathbf{x} = x_p\mathbf{e}^{(1)} .$$

Proof. The result follows from

$$M^{(1)}P_{(1,p)}\mathbf{x} = P_{(1,p)}\mathbf{x} - \boldsymbol{\mu}(\mathbf{e}^{(1)})^T P_{(1,p)}\mathbf{x} = P_{(1,p)}\mathbf{x} - \begin{pmatrix} 0 \\ \mu_2 x_p \\ \mu_3 x_p \\ \vdots \\ \mu_m x_p \end{pmatrix} = \begin{pmatrix} x_p \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

□

Note that if $x_1 \neq 0$, then one may choose $p = 1$. Also, note that p, $1 \leq p \leq m$, can be any index such that $x_p \neq 0$ so that there is not a unique Gauss transformation which transforms $\mathbf{x}$ into a constant times $\mathbf{e}^{(1)}$; this is illustrated by the following example.

Example 1.1 Let $\mathbf{x}= (0, -3, 4)^T$. If $p = 2$ and $\boldsymbol{\mu} = (0, 0, -4/3)^T$, then

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4/3 & 0 & 1 \end{pmatrix}$$

and $M^{(1)}P_{(1,2)}\mathbf{x} = (-3, 0, 0)^T$. On the other hand, if $p = 3$ and $\boldsymbol{\mu} = (0, -3/4, 0)^T$, then

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 3/4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and $M^{(1)}P_{(1,3)}\mathbf{x} = (4, 0, 0)^T$.

We can also use Gauss transformations to introduce zeros in any contiguous block of a vector. For example, given $\mathbf{x} \in \mathbf{R}^m$ and integers k and q, $1 \leq k < q \leq m$, such that x_j, $j = k, \ldots, q$, are not all zero, suppose we want to choose a Gauss transformation that zeros out the $(k + 1)$-st through q-th components of $P_{(k,p)}\mathbf{x}$. Again, an integer p, $k \leq p \leq q$, is chosen so that $x_p \neq 0$. We set

$$\mu_j = \begin{cases} 0 & \text{if } j = 1, \ldots, k \text{ or } j = q + 1, \ldots, m \\ \dfrac{x_k}{x_p} & \text{if } j = p \text{ and } p \neq k \\ \dfrac{x_j}{x_p} & \text{if } j = k + 1, \ldots, q, \ \ j \neq p \end{cases} \tag{1.4}$$

so that, from (1.3),

$$M^{(k)} = \begin{pmatrix} I_{k-1} & & \\ & 1 & \\ & -\mu_{k+1} & \\ & \vdots & \\ & -\mu_q & I_{m-k} \\ & 0 & \\ & \vdots & \\ & 0 & \end{pmatrix} \tag{1.5}$$

and

$$M^{(k)}P_{(k,p)}\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_{k-1} \\ x_p \\ 0 \\ \vdots \\ 0 \\ x_{q+1} \\ \vdots \\ x_m \end{pmatrix} .$$

We end this section by showing how an elementary permutation matrix which premultiplies a Gauss transformation "passes through" the Gauss transformation to produce another Gauss transformation postmultiplied by the same permutation matrix.

Proposition 1.2 *Let $M^{(k)}$ be a Gauss transformation and let j and p be indices such that $k < j < p$. Then there exists a Gauss transformation $\hat{M}^{(k)}$ such that*

$$P_{(j,p)}M^{(k)} = \hat{M}^{(k)}P_{(j,p)} \, . \tag{1.6}$$

Proof. The proof is by direct multiplication; $\hat{M}^{(k)}$ is the matrix obtained by interchanging the (j,k) and (p,k) entries of $M^{(k)}$. □

1.2 Triangular factorizations of an $n \times n$ matrix

The goal of this section is to prove that any $n \times n$ matrix A has the factorization $PA = LU$ where P is an $n \times n$ permutation matrix, L is an $n \times n$ unit lower triangular matrix, and U is an $n \times n$ upper triangular matrix. We will first show how, through the use of elementary permutation matrices and Gauss transformations, a given $n \times n$ can be reduced to an upper triangular matrix U. We remark that the construction given in the proof of the following proposition is exactly the classical Gaussian elimination algorithm expressed in matrix notation; we will examine this connection further in the next section.

Proposition 1.3 *Let A be a given $n \times n$ matrix. Then there exist an integer ℓ, $0 \le \ell \le n-1$, Gauss transformation matrices $M^{(k)}$, $k = 1, \ldots, \ell$, and elementary permutation matrices $P_{(k,p_k)}$, $k = 1, \ldots, \ell$, $k \le p_k \le n$, such that*

$$U = A^{(\ell+1)} = M^{(\ell)}P_{(\ell,p_\ell)} \cdots M^{(2)}P_{(2,p_2)}M^{(1)}P_{(1,p_1)}A \tag{1.7}$$

is an $n \times n$ upper triangular matrix.

Proof. Starting with $A^{(1)} = A$ and the integer counter $\sigma_1 = 1$, we assume that at the start of the k-th stage, $k \geq 1$, of the procedure we have a matrix of the form

$$A^{(k)} = \begin{pmatrix} U^{(k)} & \mathbf{a}^{(k)} & B^{(k)} \\ 0 & \mathbf{c}^{(k)} & D^{(k)} \end{pmatrix},$$

where $U^{(k)}$ is an $(k-1) \times (\sigma_k - 1)$ upper triangular matrix,, $\mathbf{a}^{(k)} \in \mathbf{R}^{k-1}$, $\mathbf{c}^{(k)} \in \mathbf{R}^{n-k+1}$, $B^{(k)}$ is $(k-1) \times (n - \sigma_k)$ and $D^{(k)}$ is $(n-k+1) \times (n-\sigma_k)$. If $\mathbf{c}^{(k)} = \mathbf{0}$, we increment σ_k by one and move on to the next column, continuing to so increment σ_k until either $\mathbf{c}^{(k)} \neq \mathbf{0}$ or $\sigma_k > n$. In the latter case we are finished since then $A^{(k)}$ is upper triangular. In the former case we use (1.4) with

$$\mathbf{x} = \begin{pmatrix} \mathbf{a}^{(k)} \\ \mathbf{c}^{(k)} \end{pmatrix}$$

and $q = m$ to choose an integer $p_k \geq k$ and construct a Gauss transformation $M^{(k)}$ such that the components of $M^{(k)} P_{(k,p_k)} \mathbf{x}$ with indices $j = k+1, \ldots, m$ vanish. If we write $M^{(k)}$ in the block form

$$\begin{pmatrix} I_{k-1} & 0 \\ 0 & \tilde{M}^{(k)} \end{pmatrix},$$

then $\tilde{M}^{(k)}$ is an $(m-k+1) \times (m-k+1)$ Gauss transformation formed by setting $\mathbf{x} = \mathbf{c}^{(k)}$ in Proposition 1.1. We have that

$$A^{(k+1)} = M^{(k)} P_{(k,p_k)} A^{(k)} = \begin{pmatrix} U^{(k)} & \mathbf{a}^{(k)} & B^{(k)} \\ 0 & c & \\ 0 & 0 & \\ \vdots & \vdots & \tilde{M}^{(k)} \hat{D}^{(k)} \\ 0 & 0 & \end{pmatrix},$$

where c denotes the $(p_k - k + 1)$-st component of $\mathbf{c}^{(k)}$ and $\hat{D}^{(k)}$ denotes the matrix $D^{(k)}$ after rows k and p_k have been interchanged. Then

$$A^{(k+1)} = \begin{pmatrix} U^{(k+1)} & \mathbf{a}^{(k+1)} & B^{(k+1)} \\ 0 & \mathbf{c}^{(k+1)} & D^{(k+1)} \end{pmatrix},$$

where $\mathbf{a}^{(k+1)}$ and $\mathbf{c}^{(k+1)}$ are the first columns of $B^{(k)}$ and $\tilde{M}^{(k)} \hat{D}^{(k)}$, respectively, $U^{(k+1)}$ is the $k \times (\sigma_{k+1} - 1)$ matrix given by

$$U^{(k+1)} = \begin{pmatrix} U^{(k)} & \mathbf{a}^{(k)} \\ 0 & c \end{pmatrix},$$

and $\sigma_{k+1} = \sigma_k + 1$. Clearly $U^{(k+1)}$ has upper triangular structure and $A^{(k+1)}$ has the same structure as $A^{(k)}$ with the index k augmented by one so that the inductive

step is complete. The total number of stages ℓ cannot exceed $(m-1)$ or n and may be less than both if $\sigma_k > k$ for some k. □

The process of row interchanges is often referred to as *row pivoting*. Since, for the most part, we will not consider column interchanges, we will henceforth refer to row pivoting as simply *pivoting*. The (k, σ_k) entry in the matrix $A^{(k)}$ defined in the above proof is referred to as the *pivot element* and (k, σ_k) itself is referred to as the *pivot position*. The pivot element is the denominator appearing in the vector $\boldsymbol{\mu}$ that determines the Gauss transformation $M^{(k)}$ used at the k-th stage. Thus, an interchange of rows is, in theory, only necessary whenever a pivot element vanishes. The row interchange process, *i.e.*, premultiplication by the elementary permutation matrix $P_{(k,p_k)}$, is invoked so that a nonzero entry is brought into the pivot position.

Proposition 1.3, coupled with Proposition 1.2, allows us to prove the major result of this section.

Theorem 1.4 *Given any $n \times n$ matrix A there exists an $n \times n$ permutation matrix P, an $n \times n$ unit lower triangular matrix L, and an $n \times n$ upper triangular matrix U such that*

$$PA = LU\,. \tag{1.8}$$

Furthermore, $\operatorname{rank}(U) = \operatorname{rank}(A)$.

Proof. By Proposition 1.3 we have that

$$A^{(\ell+1)} = M^{(\ell)} P_{(\ell,p_\ell)} M^{(\ell-1)} P_{(\ell-1,p_{\ell-1})} \cdots M^{(2)} P_{(2,p_2)} M^{(1)} P_{(1,p_1)} A$$

is upper triangular. The inverse of a square unit lower triangular matrix is also unit lower triangular so that the proof will be complete if we can show that $M^{(\ell)} P_{(\ell,p_\ell)} \cdots M^{(1)} P_{(1,p_1)} = MP$ for some unit lower triangular matrix M and permutation matrix P. Since $p_k \geq k$, from Proposition 1.2 we note that for $k = 2, \ldots, \ell$

$$P_{(k,p_k)} M^{(k-1)} = M_1^{(k-1)} P_{(k,p_k)}\,,$$

where $M_1^{(k-1)}$ is a unit lower triangular matrix. Of course, if no row interchanges are required, then $P_{(k,p_k)} = I$ and $M_1^{(k-1)} = M^{(k-1)}$. Thus we have

$$A^{(\ell+1)} = M^{(\ell)} M_1^{(\ell-1)} P_{(\ell,p_\ell)} M_1^{(\ell-2)} P_{(\ell-1,p_{\ell-1})} \cdots M_1^{(1)} P_{(2,p_2)} P_{(1,p_1)} A\,.$$

Again we use Proposition 1.2 to show that for $k = 3, \ldots, \ell$

$$P_{(k,p_k)} M_1^{(k-2)} = M_2^{(k-2)} P_{(k,p_k)}\,,$$

where $M_2^{(k-2)}$ is a unit lower triangular matrix. Continuing in this manner, we have that

$$U = A^{(\ell+1)} = MPA\,,$$

where

$$M = M^{(\ell)} M_1^{(\ell-1)} M_2^{(\ell-2)} \cdots M_{\ell-2}^{(2)} M_{\ell-1}^{(1)}$$

and

$$P = P_{(\ell,p_\ell)} \cdots P_{(2,p_2)} P_{(1,p_1)} \,.$$

Here M is the product of unit lower triangular matrices and thus M itself is unit lower triangular. Furthermore, P is the product of permutation matrices and thus P is a permutation matrix as well. Also, since M and P are invertible, clearly $\operatorname{rank}(U) = \operatorname{rank}(A)$. If A is real, then the above process involves only real arithmetic so that the end products L and U are real. □

Let $r = \operatorname{rank}(U) = \operatorname{rank}(A)$ denote the number of nonzero rows of U. It is, of course, possible for $r < n$, *e.g.*, if the rows of A are linearly dependent. If $r < n$, the $PA = LU$ factorization of the $n \times n$ matrix A may be partitioned in the form

$$PA = \begin{pmatrix} L_1 & L_2 \end{pmatrix} \begin{pmatrix} U_1 \\ 0 \end{pmatrix} = L_1 U_1 \,, \tag{1.9}$$

where L_1 is an $n \times r$ unit lower trapezoidal matrix, L_2 is $n \times (n-r)$, and U_1 is an $r \times n$ full rank upper triangular matrix. Thus (1.34) shows that an arbitrary $n \times n$ matrix with $n > r$ can be factored into the product of an $n \times r$ unit lower trapezoidal matrix and an $r \times n$ upper triangular matrix. Note that L_2 plays no essential role in the $PA = LU$ factorization of A. Also, if $r = n$, *i.e.*, A has full column rank, then U_1 is an $n \times n$ square, nonsingular, upper triangular matrix.

Example 1.2 Let

$$A = \begin{pmatrix} 0 & 0 & 4 \\ 2 & 1 & -1 \\ 6 & 3 & 1 \end{pmatrix} .$$

To form the factorization $PA = LU$ we could have the following steps:

$$\begin{aligned} M^{(2)} M^{(1)} P_{(1,2)} A &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 4 \\ 2 & 1 & -1 \\ 6 & 3 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix} = U \,. \end{aligned}$$

This gives that

$$PA = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 4 \\ 2 & 1 & -1 \\ 6 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix} = LU \,.$$

Note that if we partition L and U as in (1.34) then

$$L_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 3 & 1 \end{pmatrix} , \qquad L_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} , \qquad \text{and} \qquad U_1 = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \end{pmatrix} .$$

Note that, in general, the $PA = LU$ factorization of a matrix A is not unique. In the first place, if $r < n$ so that the partitioning of (1.34) holds, then obviously L_2 may be chosen to be any $n \times (n-r)$ matrix such that $L = (L_1\ L_2)$ is a unit lower triangular matrix. Moreover, at any stage of the process described in the proof of Proposition 1.3, one may need to interchange rows, i.e., premultiply $A^{(k)}$ by $P_{(k,p_k)}$. The index p_k, $k \le p_k \le n$, is not uniquely determined; one may in fact choose any such index such that the $(p_k - k + 1)$-st component of $\mathbf{c}^{(k)}$ is nonzero. In view of these observations we have the following uniqueness result.

Proposition 1.5 *Given an $n \times n$ matrix A. Partition its $PA = LU$ factorization as in* (1.34) *where U_1 has full row rank and L_1 is unit lower trapezoidal. Then, once the row interchange strategy is fixed, i.e., the permutation matrix P is fixed, the matrices L_1 and U_1 appearing in the factorization* (1.34) *are uniquely determined. If $r = \operatorname{rank}(A)$, the number of rows in U_1, then L_2 may be chosen to be any $n \times (n-r)$ matrix such that $L = (L_1\ L_2)$ is a unit lower triangular matrix.*

Proof. Let $PA = LU = \tilde{L}\tilde{U}$ be two factorizations of A using the same row interchange strategy. Then, $U = L^{-1}\tilde{L}\tilde{U}$. Further partition L into the form

$$L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}, \tag{1.10}$$

where L_{11} is an $r \times r$ unit lower triangular matrix, L_{22} is an $(n-r) \times (n-r)$ unit lower triangular matrix, and L_{21} is $(n-r) \times r$. Note that

$$L_1 = \begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix} \quad \text{and} \quad L_2 = \begin{pmatrix} 0 \\ L_{22} \end{pmatrix}. \tag{1.11}$$

We will use the analogous partitioning for $\tilde{L}$. Then

$$L^{-1}\tilde{L} = \begin{pmatrix} L_{11}^{-1}\tilde{L}_{11} & 0 \\ -L_{22}^{-1}L_{21}L_{11}^{-1}\tilde{L}_{11} + L_{22}^{-1}\tilde{L}_{21} & L_{22}^{-1}\tilde{L}_{22} \end{pmatrix}. \tag{1.12}$$

Then, since $U = L^{-1}\tilde{L}\tilde{U}$, we have that

$$U_1 = L_{11}^{-1}\tilde{L}_{11}\tilde{U}_1, \tag{1.13}$$

where we have partitioned U and $\tilde{U}$ as in (1.34). In (1.13), the matrix on the left is upper triangular, while the matrix on the right is the product of the unit lower triangular matrix $L_{11}^{-1}\tilde{L}_{11}$ and the upper triangular matrix $\tilde{U}_1$. By equating the elements to the left of the first nonzero entry in the rows on both sides of (1.13), we conclude that the lower triangular matrix $L_{11}^{-1}\tilde{L}_{11}$ is a diagonal matrix; since it is also a unit lower triangular matrix, we conclude that $L_{11}^{-1}\tilde{L}_{11} = I$, or $L_{11} = \tilde{L}_{11}$. Then, (1.13) also yields that $U_1 = \tilde{U}_1$.

The relation $U = L^{-1}\tilde{L}\tilde{U}$, (1.12), and the partitioning of U and $\tilde{U}$ of (1.34) also yield that

$$L_{22}^{-1} L_{21} L_{11}^{-1} \tilde{L}_{11} = L_{22}^{-1} \tilde{L}_{21} .$$

Then, since $L_{11} = \tilde{L}_{11}$, we have that $L_{21} = \tilde{L}_{21}$, and, from (1.11), $L_1 = \tilde{L}_1$. □

If A has full row rank, *i.e.*, if $r = n$, then $U = U_1$ and $L = L_1$ so that in this case we have that, once the row interchange strategy is fixed, the factorization of PA into the product of a unit lower triangular matrix and an upper triangular matrix is unique. In particular, this is the case when A is square and invertible.

There are numerous variants to the factorization $PA = LU$; the most important is considered in Section 1.2.1. Another variant is found by applying Theorem 1.4 to A^* which leads to the following result. Given any $n \times n$ matrix A there exists an $n \times n$ permutation matrix $\tilde{P}$, an $n \times n$ unit upper triangular matrix $\tilde{U}$, and an $n \times n$ lower triangular matrix $\tilde{L}$ such that $A\tilde{P} = \tilde{L}\tilde{U}$. In fact, $\tilde{L}$ is such that $\tilde{L}^*$ has row echelon structure.

Another variant of the basic factorization (1.33) is given by

$$PA = LD\hat{U} , \tag{1.14}$$

where P is an $n \times n$ permutation matrix, L is an $n \times n$ unit lower triangular matrix, $\hat{U}$ is an $n \times n$ unit upper triangular matrix, and D is an $n \times n$ diagonal matrix. If A has full row rank, *i.e.*, rank $(A) = n$, then, for $j = 1, \ldots, n$, $d_{j,j}$ is equal to the first nonzero entry in the j-th row of U and $\hat{U} = D^{-1}U$, where U denotes the upper trapezoidal matrix of (1.33); D^{-1} exists by virtue of A being full rank. If rank $(A) = r < n$, then, for $j = 1, \ldots, r$, $d_{j,j}$ is equal to the first nonzero entry in the j-th row of U and for $j > r$, $d_{j,j}$ may be chosen arbitrarily. Of course, the nontrivial rows of $\hat{U}$ may be determined from those of U by dividing the rows of the latter by the corresponding first nonzero entry. The $PA = LD\hat{U}$ factorization in the case of $r < n$ is illustrated by the following example.

Example 1.3 Let

$$A = \begin{pmatrix} 0 & 0 & 4 \\ 2 & 1 & -1 \\ 6 & 3 & 1 \end{pmatrix} .$$

Then, from Example 1.2, $P_{(1,2)}A$ has the LU factorization

$$P_{(1,2)}A = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \\ 6 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix} .$$

Even though A is singular, $P_{(1,2)}A$ also has the factorization $LD\hat{U}$ given by

$$P_{(1,2)}A = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 4 \\ 6 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & 1/2 & -1/2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} .$$

Note that $\hat{U}$ is in row echelon form and that the $(3,3)$ entry of D may be set arbitrarily since the third row of U contains only zero entries.

1.2.1 Triangular factorizations without row interchanges

The appearance of the permutation matrix P in (1.33) results from the need to pivot, *i.e.*, to perform row interchanges, whenever a zero pivot element is encountered. If row interchanges are not required, we may set $P = I$ and then $A = LU$. The following example gives factorizations for two nonsingular matrices. The first has an LU factorization, *i.e.*, no pivoting is necessary. The second matrix fails to have an LU factorization unless pivoting is performed.

Example 1.4 Let

$$A = \begin{pmatrix} 2 & -1 & 0 \\ 4 & -5 & 3 \\ 6 & -6 & -2 \end{pmatrix} .$$

Then A has the LU factorization

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & 1 \end{pmatrix} .$$

Let

$$B = \begin{pmatrix} 0 & 4 & 1 \\ 2 & -1 & 3 \\ -4 & 14 & -2 \end{pmatrix} .$$

Then, since the first pivot element, *i.e.*, $b_{1,1}$, vanishes, B fails to have an LU factorization, *i.e.*, we can't write $B = LU$. However, if we interchange the first and second rows we see that $P_{(1,2)}B$ has the LU factorization

$$P_{(1,2)}B = LU = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 3 \\ 0 & 4 & 1 \\ 0 & 0 & 1 \end{pmatrix} .$$

A characterization of matrices for which zero pivot elements are not encountered, and thus of matrices A that possess a factorization of the form $A = LU$, is given in the following result. Here, given a matrix A with elements $a_{i,j}$, $i = 1, \ldots, n$ and $j = 1, \ldots, n$, the k-th *leading principal submatrix* $\mathcal{A}_k$, $k \leq n$, is the $k \times k$ matrix with elements $a_{i,j}$, $i = 1, \ldots, k$ and $j = 1, \ldots, k$.

Proposition 1.6 *Given an $n \times n$ matrix A, denote its leading principal $k \times k$ submatrices by $\mathcal{A}_k$ for $k = 1, \ldots, n$. If $\mathcal{A}_k$ is nonsingular for $k = 1, \ldots, \ell = n - 1$, then there exists an $n \times n$ unit lower triangular matrix L and an $n \times n$ upper triangular matrix U such that*

$$A = LU , \tag{1.15}$$

where $U = A^{(\ell+1)}$ and L is given explicitly by

$$L = (L_1\ L_2)\,, \quad \textit{where} \quad L_1 = \begin{pmatrix} 1 & & & \\ \mu_2^{(1)} & 1 & & \\ \mu_3^{(1)} & \mu_3^{(2)} & & \\ & & \ddots & \\ & & & 1 \\ \vdots & \vdots & & \mu_{\ell+1}^{(\ell)} \\ & & & \vdots \\ \mu_m^{(1)} & \mu_m^{(2)} & \cdots & \mu_m^{(\ell)} \end{pmatrix} \tag{1.16}$$

and where L_2 is any $n \times (n-\ell)$ matrix such that $L = (L_1\ L_2)$ is an unit upper triangular matrix. Here $A^{(k)}$, $k = 1, \ldots, (\ell+1)$, is defined in the proof of Proposition (1.3) *and*

$$\mu_j^{(k)} = \frac{a_{j,k}^{(k)}}{a_{k,k}^{(k)}}\,. \tag{1.17}$$

Moreover, $u_{i,i} \neq 0$ for $i = 1, \ldots, n-1$. If A is real then U and L may be chosen to be real as well.

Proof. The proof is merely a specialization of the proofs of Proposition 1.3 and Theorem 1.4. Starting with $A^{(1)} = A$, suppose we have completed the $(k-1)$-st stage of the reduction procedure without having encountered any vanishing pivots, *i.e.*, in the proof of Proposition 1.3 we have that $P_{(j,p_j)} = I$ for $j = 1, \ldots, k-1$. Thus at the start of the k-th stage we assume that we have the partially reduced matrix

$$A^{(k)} = M^{(k-1)} \cdots M^{(1)} A\,,$$

where the unit lower triangular matrices $M^{(j)}$, $j = 1, \ldots, k-1$, are defined by (1.5) with $q = m$ and where the matrix $A^{(k)}$ has its first $(k-1)$ columns in upper triangular form and its first $(k-1)$ diagonal entries do not vanish. We then have that

$$A = L^{(k)} A^{(k)}\,,$$

where

$$L^{(k)} = \left(M^{(k-1)} \cdots M^{(1)} \right)^{-1}$$

is an $n \times n$ unit lower triangular matrix. By using the special structure of $L^{(k)}$ and $A^{(k)}$, we may partition $A = L^{(k)} A^{(k)}$ into blocks to obtain

$$A = \begin{pmatrix} \mathcal{A}_k & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_{m-k} \end{pmatrix} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}\,,$$

where $\mathcal{A}_k$, $L_{11}^{(k)}$, and $A_{11}^{(k)}$ are $k \times k$, $L_{11}^{(k)}$ is a unit lower triangular matrix, and $A_{11}^{(k)}$ is an upper triangular matrix. We equate the upper left-hand blocks to obtain $\mathcal{A}_k = L_{11}^{(k)} A_{11}^{(k)}$ and therefore $\det \mathcal{A}_k = \det(L_{11}^{(k)} A_{11}^{(k)}) = \det L_{11}^{(k)} \det A_{11}^{(k)} = \det A_{11}^{(k)}$. Since $A_{11}^{(k)}$ is upper triangular, we have that

$$\det \mathcal{A}_k = a_{1,1}^{(1)} a_{2,2}^{(2)} \cdots a_{k,k}^{(k)}, \tag{1.18}$$

i.e., $\det \mathcal{A}_k$ is the product of the first k pivot elements. Now, by hypothesis, $\det \mathcal{A}_k \neq 0$. Also, by virtue of the assumption that $A^{(k)}$ is deduced from A without encountering any vanishing pivot elements, $a_{j,j}^{(j)} \neq 0$ for $j = 1, \ldots, k-1$. Then, from (1.18), $a_{k,k}^{(k)} \neq 0$ and the inductive step is complete so that we have

$$A^{(\ell+1)} = M^{(\ell)} M^{(\ell-1)} \cdots M^{(1)} A,$$

where $A^{(\ell+1)} = U$ is upper triangular. The derivation of the explicit representation of (1.36) for

$$L = \left(M^{(\ell)} \cdots M^{(1)} \right)^{-1}$$

is left as an exercise. □

Some important classes of matrices satisfy the hypotheses of this proposition; these include positive definite and diagonally dominant matrices. We will discuss the former in the following section.

The uniqueness of the $A = LU$ factorization is explored in the exercises.

It is important to note that Proposition 1.12 gives *sufficient* conditions for a matrix to have an LU factorization without pivoting. The following example gives a matrix which does possess an LU factorization without pivoting but fails to satisfy the hypotheses of Proposition 1.12.

Example 1.5 Let

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

Then A has the LU factorization

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

Note that A does not satisfy the hypotheses of Proposition 1.12.

1.2.2 Symmetric positive definite matrices

We now consider an important class of square matrices, namely *symmetric positive definite* matrices. Recall that a square matrix is symmetric if $A^T = A$ and is positive definite if $\mathbf{x}^T A\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{x} \neq \mathbf{0}$.

When A is positive definite, we can easily show that A can be factored in the form $A = LU$ by demonstrating that A satisfies the hypotheses of Proposition 1.12. (See the first part of the proof of Proposition 1.7 below.) If A is also symmetric then it turns out there is only one matrix involved in the factorization, *i.e.*, we can write $A = LL^T$, where now L is an invertible lower triangular matrix that in general does not have unit diagonal entries. This is the classic result known as the *Cholesky factorization* of a symmetric positive definite matrix. In addition, we can show that the converse is true, *i.e.*, if A is symmetric and there exists an invertible lower triangular matrix L such that $A = LL^T$, then A must be positive definite. The following result formalizes these observations.

Proposition 1.7 *Let A be an $n \times n$ symmetric matrix. Then A is positive definite if and only if there exists an invertible lower triangular matrix L such that*

$$A = LL^T . \tag{1.19}$$

Furthermore, one can choose the diagonal elements $l_{i,i}$, $i = 1, \ldots, n$, of L to be real positive numbers. In this case the factorization (1.19) *is unique. If A is real then L is real as well and we have that $A = LL^T$.*

Proof. If A is positive definite, $\mathbf{x}^T A\mathbf{x} > 0$ for $\mathbf{x} \in \mathbf{R}^n$ such that $\mathbf{x} \neq \mathbf{0}$. Choose $\mathbf{x}^T = (\mathbf{y}^T, \mathbf{0})$ for an arbitrary nonzero vector $\mathbf{y} \in \mathbf{R}^k$. We have that

$$0 < \begin{pmatrix} \mathbf{y}^T & 0 \end{pmatrix} A \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{y}^T & 0 \end{pmatrix} \begin{pmatrix} \mathcal{A}_k & \times \\ \times & \times \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} = \mathbf{y}^T \mathcal{A}_k \mathbf{y}$$

so that $\mathcal{A}_k$, the k-th leading principal submatrix of A, is positive definite. Furthermore, since $\mathcal{A}_k$ is positive definite it is nonsingular. Thus from Proposition 1.12 we have that

$$A = LU = \begin{pmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \vdots & & \ddots & \\ l_{n,1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ & u_{2,2} & \cdots & u_{2,n} \\ & & \ddots & \vdots \\ & & & u_{n,n} \end{pmatrix}$$

and, since $\det A = \prod_{i=1}^{n} u_{i,i}$ and $\det A \neq 0$, we have $u_{i,i} \neq 0$ for $i = 1, \ldots, n$. Let D be the diagonal matrix $\operatorname{diag}(u_{1,1}, u_{2,2}, \ldots, u_{n,n})$ so that D^{-1} is well defined. Then we can write $A = LD\hat{U}$ where $\hat{U} = D^{-1}U$ is a unit upper triangular matrix. Now A is also symmetric so that $LD\hat{U} = \hat{U}^T D^T L^T$, or

$$D\hat{U}L^{-T} = L^{-1}\hat{U}^T D^T . \tag{1.20}$$

Now the product of the matrices on the left-hand side of (1.20) is an upper triangular matrix with diagonal entries given by those of D. The product of the matrices on the right-hand side is a lower triangular matrix with corresponding diagonal

entries given by those of D^T. Thus both are diagonal matrices and we have that $D = D\hat{U}L^{-T} = L^{-1}\hat{U}^T D^T = D^T$. Hence we have $D = D^T$ and $\hat{U} = L^T$, *i.e.*, $A = LDL^T$ with L a unit lower triangular matrix and D a diagonal matrix with real entries. Since A is positive definite, we have that $\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T LDL^T\mathbf{x} = \mathbf{y}^T D\mathbf{y} > 0$, where $\mathbf{y} = L^T\mathbf{x}$. Thus we conclude that the diagonal entries of D are positive. Let $D^{1/2} = \text{diag}\,(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$ and let $\hat{L} = LD^{1/2}$. Then $A = LDL^T = LD^{1/2}D^{1/2}L^T = \hat{L}\hat{L}^T$ where the diagonal elements of $\hat{L}$ are real positive numbers. If A is real only real arithmetic is used to arrive at (1.19) so that $\hat{L}$ is real.

To show that the factorization is unique when the diagonal elements of L are real positive numbers, we let $A = L_1 L_1^T = L_2 {L_2}^T$. Then

$$L_1{}^{-1}L_2 = L_1^T L_2{}^{-T}\,. \tag{1.21}$$

The product of the matrices on the left-hand side of (1.21) is lower triangular while that on the right-hand side is upper triangular; hence $L_1{}^{-1}L_2 = L_1^T L_2{}^{-T} = G$ for some diagonal matrix G. Now $G = L_1{}^{-1}L_2$ implies that $g_{i,i} = l_{2i,i}/l_{1i,i}$ and $G = L_1^T L_2{}^{-T}$ implies that $g_{i,i} = \bar{l}_{1i,i}/\bar{l}_{2i,i}$ so that $|l_{1i,i}| = |l_{2i,i}|$. Since both L_1 and L_2 have real positive diagonal entries this implies that $l_{1i,i} = l_{2i,i}$, or that $g_{i,i} = 1$. Hence $G = I$ and $L_1 = L_2$.

Now assume that A is symmetric and that (1.19) holds. Then $\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T LL^T\mathbf{x} = \mathbf{y}^T\mathbf{y}$ where $\mathbf{y} = L^T\mathbf{x}$. Now $\mathbf{y}^T\mathbf{y} > 0$ except for $\mathbf{y} = \mathbf{0}$ or equivalently when $\mathbf{x} = \mathbf{0}$ since L is invertible. Thus $\mathbf{x}^T A\mathbf{x} > 0$ unless $\mathbf{x} = \mathbf{0}$ which is, for symmetric matrices, the definition of A being positive definite. □

Example 1.6 Let A be given by

$$\begin{pmatrix} 1 & 2 & -1 \\ 2 & 13 & 13 \\ -1 & 13 & 42 \end{pmatrix}.$$

Then A has the Cholesky factorization

$$\begin{pmatrix} 1 & 2 & -1 \\ 2 & 13 & 13 \\ -1 & 13 & 42 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ -1 & 5 & 4 \end{pmatrix}\begin{pmatrix} 1 & 2 & -1 \\ 0 & 3 & 5 \\ 0 & 0 & 4 \end{pmatrix}.$$

1.3 Systems of algebraic equations

In this section we compare the use of triangular factorizations of a matrix with Gaussian elimination for solving linear systems of algebraic equations. We begin with the Gaussian elimination algorithm which makes use of elementary row operations to reduce the given $n \times n$ coefficient matrix of the linear system to an upper triangular matrix; at the same time, the elementary row operations are applied to the right-hand side of the system. If this system is consistent, it can then

be solved by a generalized back substitution algorithm. The Gaussian elimination algorithm we study here uses partial pivoting and implicit row scaling strategies. Examples illustrating the need for pivoting and scaling are given. The triangular factorization algorithms studied in the previous section, coupled with forward and back substitution algorithms, are also used to solve linear systems.

A system of n linear algebraic equations in n unknowns is described by the relations

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i \quad \text{for } i = 1, \ldots, n\,,$$

where the $n \times n$ numbers $a_{i,j}$ and the n numbers b_i are given and the n numbers x_j are to be determined. The system of equations may be expressed in matrix form as

$$A\mathbf{x} = \mathbf{b}\,, \tag{1.22}$$

where A is an $n \times n$ matrix with elements $a_{i,j}$, $\imath = 1, \ldots, n$, $j = 1, \ldots, n$ and $\mathbf{x}$, $\mathbf{b}$ are n- and m-column vectors with elements x_j, $j = 1, \ldots, n$ and b_i, $\imath = 1, \ldots, n$, respectively. We will refer to the matrix A as the *coefficient matrix* and the vector $\mathbf{b}$ as the *right-hand side vector* of the linear system. If A is $m \times n$, and $m > n$ we refer to the system (1.22) as *overdetermined* whereas if $m < n$ we refer to it as *underdetermined*.

1.3.1 Pivoting and scaling

If one computes with infinite precision arithmetic, then row interchanges are necessary only when a pivot element is exactly zero. However, in computations using finite precision arithmetic, small pivot elements can have a disastrous effect on the accuracy of the computed results. The following well known example illustrates this point.

Example 1.7 Consider the system

$$\begin{pmatrix} .001 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

whose exact solution is $x_1 = 1000/999$ and $x_2 = 998/999$. If one reduces this system to triangular form without interchanging rows using base ten arithmetic rounded to two significant digits, then one obtains the triangular system

$$\begin{pmatrix} .001 & 1 \\ 0 & 1000 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1000 \end{pmatrix}.$$

A back substitution procedure gives the computed solution $x_1 = 0$ and $x_2 = 1$. On the other hand, if we write the system by reordering the equations, *i.e.*, interchanging rows, then we have

$$\begin{pmatrix} 1 & 1 \\ .001 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Using two-digit arithmetic one finds the computed solution to be $x_1 = x_2 = 1$, which is also the exact solution rounded to two significant digits.

Now consider the problem

$$\begin{pmatrix} 10 & 10,000 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10,000 \\ 2 \end{pmatrix}$$

which is obtained from the original example by multiplying the first equation by $10,000$. Invoking a partial pivoting strategy would not effect an interchange of rows. The solution to this system using two-digit arithmetic is $x_1 = 0$ and $x_2 = 1$, which is exactly the answer we obtained for the original system without pivoting. It is thought that the trouble with the last example is that the matrix is not properly scaled, *i.e.*, the matrix elements vary wildly.

In determining the factorization $PA = LU$ we pivoted, *i.e.*, performed row interchanges, only when a zero pivot element $a^{(k)}_{k,\sigma_k}$ was encountered. However, as the previous example demonstrates, pivoting may be necessary even when a pivot element is nonzero in order to avoid roundoff error problems. Thus some criteria must be chosen to determine the row interchange strategy. The algorithm we present in Section 1.3.3 for $PA = LU$ employs *implicit row scaling* and *partial pivoting* strategies. Here we define these strategies. We define the *scale factors* s_i for each row $i = 1, \ldots, m$ by

$$s_i = \sum_{j=1}^{n} |a_{i,j}| \,. \tag{1.23}$$

At the k-th stage of the algorithm we have a matrix $A^{(k)}$ whose first $(k-1)$ rows and $(\sigma_k - 1)$ columns, for $\sigma_k \geq k$, have upper triangular structure. We search the $(n-k+1)$ entries $a^{(k)}_{i,\sigma_k}$, $i = k, \ldots, n$, of $A^{(k)}$ to find the first index p_k such that

$$\frac{|a^{(k)}_{p_k,\sigma_k}|}{s_{p_k}} = \max_{\substack{i=k,\ldots,n \\ s_i \neq 0}} \frac{|a^{(k)}_{i,\sigma_k}|}{s_i} \,. \tag{1.24}$$

This procedure of finding the maximal element in the column is called *partial pivoting* and the choice of scaling each element during the pivot search by the factors s_i given in (1.23) is called *implicit row scaling*. (If the search (1.24) indicates that $a^{(k)}_{i,\sigma_k} = 0$ for $i = k, \ldots, n$, we increment σ_k by one, *i.e.*, move to the next column and repeat the search.)

Ideally, *i.e.*, in infinite precision arithmetic, the search (1.24) is needed only if $a^{(k)}_{k,\sigma_k} = 0$. In practice one uses finite precision arithmetic so that the search for p_k and the maximal pivot element $a^{(k)}_{p_k,\sigma_k}$, along with the subsequent interchange of rows k and p_k, help to avoid problems due to roundoff errors. This issue was illustrated in the example.

We note that the scale factors s_i, $\imath = 1, \ldots, n$, in (1.23) are determined only once from the elements of the given matrix A and are only used in (1.24) for the

determination of the pivoting strategy; the elements of the matrix A are never explicitly scaled. Hence the terminology *implicit* row scaling.

In the search for the pivot element defined by (1.24) we only considered elements in column σ_k on or below the k-th row. A more complicated process selects the pivot element for the matrix $A^{(k)}$ encountered at the beginning of the k-th stage by a search of the type

$$|a_{p,q}^{(k)}| = \max_{\substack{i=k,\ldots,n \\ j=k,\ldots,n}} |a_{i,j}^{(k)}|\,,$$

where for simplicity we have omitted scalings. Thus we now choose the pivot element to be an element of maximum modulus among all elements of the $(n-k+1)\times(n-k+1)$ submatrix with entries $a_{i,j}^{(k)}$, $i=k,\ldots,n$, $j=k,\ldots,n$. This strategy is known as *full* or *complete pivoting*. In the presence of roundoff errors the full pivoting strategy is theoretically more stable than is the partial pivoting strategy of (1.24); however, in the great majority of practical cases the latter appears to be adequate with regards to the avoidance of ill effects resulting from roundoff errors.

Returning to our example, we see that if we use the implicit row scaling described here to determine the pivoting strategy, then the row scaling would force the interchange of the rows of the last system, *i.e.*, one solves the system

$$\begin{pmatrix} 1 & 1 \\ 10 & 10,000 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 10,000 \end{pmatrix}$$

which, in two-digit arithmetic, yields the same good answer $x_1 = x_2 = 1$.

To prevent difficulties of the type that occur in Example 1.3.1, we incorporate a row scaling into our algorithms. There are other types of scalings which would have a similar effect as the implicit row scaling used here.

1.3.2 Solving linear systems using Gaussian elimination

The goal of the Gaussian elimination method is to transform the $n\times n$ coefficient matrix A into an $n\times n$ matrix U which is upper triangular; this reduction is accomplished using *elementary row operations*, *i.e.*, the interchange of two rows, and the replacement of a row by the sum of that row and a scalar multiple of another row. At the same time, the elementary row operations are applied to the right-hand side vector $\mathbf{b}$. The algorithm given here is essentially the method described in Proposition 1.3 where we transformed an $n\times n$ matrix A into an upper triangular matrix U by premultiplication by permutation matrices and Gauss transformation matrices. In fact, premultiplication by these matrices effect the elementary row operations of the Gaussian elimination method, *i.e.*, the permutation matrices effect the row interchange operations and the Gauss transformations effect the row replacement operations. In Gaussian elimination the elementary row operations applied to the coefficient matrix A and right-hand side vector $\mathbf{b}$ result in a linear system of the form

$$U\mathbf{x} = \mathbf{c}\,, \tag{1.25}$$

where $\mathbf{x}$ is the solution of (1.22), U is exactly the same matrix as that obtained in the $PA = LU$ factorization of A, and $\mathbf{c}$ is given by

$$\mathbf{c} = M^{(\ell)} P_{(\ell,p_\ell)} \cdots M^{(2)} P_{(2,p_2)} M^{(1)} P_{(1,p_1)} \mathbf{b}\,. \tag{1.26}$$

Here $M^{(k)}$ and $P_{(k,p_k)}$, $k = 1, \ldots, \ell$, are the Gauss transformations and elementary permutation matrices, respectively, that are used to find the $PA = LU$ factorization of A. Thus, Gaussian elimination and triangular factorization are equivalent; in fact, it is easy to show that $\mathbf{c} = L^{-1}P\mathbf{b}$ so that (1.25) is equivalent to $PA\mathbf{x} = LU\mathbf{x} = P\mathbf{b}$.

Recall that in the algorithm for the triangular factorization of A we did not generate L and U as in Proposition 1.2 but rather we obtained the equations for the components of the matrices by equating entries on each side of equations such as $PA = LU$. In the Gaussian elimination method we actually carry out the steps found in the proof of Proposition 1.2 without explicitly forming the Gauss transformations $M^{(k)}$ or the permutation matrices $P_{(k,p_k)}$

In the following algorithm we use the implicit row scaling and partial pivoting strategies that were described above. As was the case in Algorithm 1.5, we do not physically interchange the rows, but rather keep track of the interchanges in the vector $\boldsymbol{\gamma}$. Also, the vector $\boldsymbol{\sigma}$ stores the column index of the first nonzero entry in each row and the scalar r keeps track of the number of nonzero pivot elements encountered.

Algorithm 1.1 Gaussian elimination with implicit row scaling and partial pivoting. *Given an $n \times n$ matrix A and an n-vector $\mathbf{b}$, this algorithm transforms the system $A\mathbf{x} = \mathbf{b}$ into a system of the form* (1.25)*, where U is overwritten onto A and $\mathbf{c}$ is overwritten onto $\mathbf{b}$. The algorithm uses the partial pivoting and implicit row scaling strategies discussed in Section* **??** *to choose the pivot element; in particular, the pivot search is determined by* (1.24)*. On output, $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}$ provide the same information as in Algorithm 1.5. In particular, if $\sigma_k > n$, then row k contains only zero elements.*

Set $k = 1$ and $\sigma_1 = 1$.

For $i = 1, \ldots, n$, set $\gamma_i = i$ and $s_i = \displaystyle\sum_{j=1}^{n} |a_{i,j}|\,.$

Do while $k \le n$, $\sigma_k \le n$, and $\displaystyle\sum_{i=k}^{m} s_{\gamma_i} \ne 0$:

do while $\displaystyle\max_{\substack{i=k,\ldots,n \\ s_{\gamma_i}\ne 0}} \frac{|a_{\gamma_i,\sigma_k}|}{s_{\gamma_i}} = 0$:

set $\sigma_k \leftarrow \sigma_k + 1$;

set p equal to the smallest integer such that

$$\frac{|a_{\gamma_p,\sigma_k}|}{s_{\gamma_p}} = \max_{\substack{i=k,\ldots,m \\ s_{\gamma_i}\neq 0}} \frac{|a_{\gamma_i,\sigma_k}|}{s_{\gamma_i}} ;$$

if $k < n$

if $k \neq p$, interchange the contents of γ_p and γ_k and

· for $i = k+1, \ldots, n$ set

$$\mu = -\frac{a_{\gamma_i,\sigma_k}}{a_{\gamma_k,\sigma_k}} ,$$

$$b_{\gamma_i} \leftarrow b_{\gamma_i} + \mu b_{\gamma_k} ; \quad \text{and}$$

· for $j = \sigma_k + 1, \ldots, n$, set

$$a_{\gamma_i,j} \leftarrow a_{\gamma_i,j} + \mu a_{\gamma_k,j} ;$$

set $\sigma_{k+1} = \sigma_k + 1$;

set $k \leftarrow k+1$.

For $i = k, \ldots, n$, set $\sigma_i = n+1$.

□

For some classes of matrices, *e.g.*, square positive definite matrices, it is known *a priori* that Gaussian elimination can proceed stably without the need for any row interchanges. In such cases it is wasteful to perform the pivot search and therefore that step is removed from the algorithm. Also, in this case there is no need to introduce the interchange array $\boldsymbol{\gamma}$, the pivot position array $\boldsymbol{\sigma}$, or the scale factors s_i.

A few comments should be made concerning the algorithm. First, the row replacement loop begins in colunbyn $(\sigma_k + 1)$ even though we are eliminating in column σ_k. This avoids the computation of a_{γ_i,σ_k} for $i = k+1, \ldots, n$ which are known to vanish. If the m-th stage is reached, then we are working with the last row and therefore we do not need to eliminate any elements. Thus if $k = n$ we exit after ascertaining whether or not the n-th row contains any nonzero elements and the pivot position for the n-th row. We also note that the scale factors s_i, $i = 1, \ldots, n$, are determined once and for all from the elements of the given matrix A and are only used in the determination of the interchange strategy; the elements of the matrix A are not explicitly scaled. Concerning the storage required by the algorithm, we see that through overwriting the elements of A we can implement the algorithm with roughly the same storage as that required to store A itself. Of course, if one overwrites then the matrix A will be destroyed during the computations. Once again we remark that other pivoting strategies could be incorporated such as a full pivoting strategy discussed in Section **??**.

A variant of the above algorithm, known as *Gauss-Jordan elimination*, explicitly scales rows so that the pivot element is set to unity and one eliminates above as well as below the pivot position. The outcome of Gauss-Jordan elimination is a matrix which is in *row reduced echelon form*; such a matrix is a row echelon matrix whose entries in a column containing a pivot element all vanish excepting, of course, for the pivot element itself. In particular, if the given matrix A is square and of full rank, *i.e.*, invertible, the result of Gauss-Jordan elimination is simply the identity matrix. Although Gauss-Jordan elimination is of substantial theoretical importance, it is not as efficient to use as the above version of Gaussian elimination for solving linear systems.

In the following examples the partial pivoting strategy of Algorithm 1.1 is used to reduce the given systems to the form (1.25).

Example 1.8 Let A be the nonsingular matrix

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 0 & -2 & 7 \\ 4 & 4 & 5 \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

The reduction of the system to the form (1.25) using the pivoting strategy of Algorithm 1.1 gives the following sequence of matrices

$$\left(\begin{array}{ccc|c} 2 & 1 & 3 & 1 \\ 0 & -2 & 7 & 2 \\ 4 & 4 & 5 & 4 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & 5 & 4 \\ 0 & -2 & 7 & 2 \\ 2 & 1 & 3 & 1 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & 5 & 4 \\ 0 & -2 & 7 & 2 \\ 0 & -1 & \frac{1}{2} & -1 \end{array}\right)$$
$$\rightarrow \left(\begin{array}{ccc|c} 4 & 4 & 5 & 4 \\ 0 & -2 & 7 & 2 \\ 0 & 0 & -3 & -2 \end{array}\right),$$

where we have augmented the coefficient matrix with the right-hand side vector. If the storage scheme of Algorithm 1.1 is used then the resulting upper triangular matrix is stored as

$$\begin{pmatrix} 0 & 0 & -3 \\ 0 & -2 & 7 \\ 4 & 4 & 5 \end{pmatrix}.$$

Example 1.9 Consider the system $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{pmatrix} 1 & -2 & 1 & -4 \\ 1 & 3 & 7 & 2 \\ 1 & -12 & -11 & -16 \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

The reduction of the system to the form (1.25) using the pivoting strategy of Algorithm 1.1 gives the following sequence of matrices

$$\left(\begin{array}{cccc|c} 1 & -2 & 1 & -4 & 1 \\ 1 & 3 & 7 & 2 & 1 \\ 1 & -12 & -11 & -16 & 1 \end{array}\right) \rightarrow \left(\begin{array}{cccc|c} 1 & -2 & 1 & -4 & 1 \\ 0 & 5 & 6 & 6 & 0 \\ 0 & -10 & -12 & -12 & 0 \end{array}\right)$$

$$\rightarrow \left(\begin{array}{rrrr|r} 1 & -2 & 1 & -4 & 1 \\ 0 & -10 & -12 & -12 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right),$$

where we have augmented the coefficient matrix with the right-hand side.

Using Algorithm 1.1 we can reduce any $n \times n$ linear system to one of the form (1.25). The case of most interest is when the square matrix A is invertible. In this case the unique solution of the upper triangular system (1.25) may be easily determined by a *back substitution* procedure. In this procedure we simply solve for the unknowns x_i, $i = 1, \ldots, n$, in reverse order. However, if A is not invertible (or if it is rectangular) we may still use a back substitution procedure to find a *particular solution* of (1.22) or (1.25). Of course, for a solution to exist these systems must be consistent, *i.e.*, the given right-hand side vector $\mathbf{b}$ must belong to the column space of A or equivalently, must satisfy $\mathbf{z}^T\mathbf{b} = 0$ for all n-vectors $\mathbf{z}$ such that $A^T\mathbf{z} = 0$. (See Appendix I.) The consistency of the system (1.25) is easily determined since the system $A\mathbf{x} = \mathbf{b}$ is consistent if and only if the sytem $U\mathbf{x} = \mathbf{c}$ is consistent. Indeed, one must only check if $c_k = 0$ for all values of k such that the k-th row of U contains only zero entries, *i.e.*, such that $\sigma_k > n$. Obviously, if $c_k \neq 0$ for such a row, the system is inconsistent.

Once the consistency of the system has been verified, one may proceed with the back substitution process. The components of $\mathbf{x}$ which correspond to columns of U which do not contain a pivot element may be set to any arbitrary value; we call these variables *free variables*. Starting with the last such row, the non-trivial rows of U are used to determine the value of the remaining components of $\mathbf{x}$ which we call *pivot variables*. Specifically, a non-trivial row, say row k, is used to determine the value of the component x_{σ_k}, where σ_k gives the pivot position of the k-th row. The other components x_i, $i > \sigma_k$, which can possibly appear in the equation corresponding to the k-th row of (1.25) are either free variables or have been previously determined from the equations corresponding to rows of (1.25) below the k-th row. Clearly, the number of pivot variables equals the rank of A and the number of free variables is $n - \text{rank}(A)$. In the algorithm which follows we set the free variables to unity; other choices for the values of the free variables are also easily implemented.

We give the *generalized back substitution algorithm* in the notation of the generalized Gaussian elimination algorithm 1.1. Here $a_{i,j}$, b_i, σ_k and γ_k refer to results of the Gaussian elimination process.

Algorithm 1.2 Generalized back substitution. *Given the linear system $A\mathbf{x} = \mathbf{b}$, where A is an $n \times n$ matrix and $\mathbf{b}$ is an $n-$vector. Assume that Algorithm 1.1 has been used to reduce the system to the form $U\mathbf{x} = \mathbf{c}$ where U and $\mathbf{c}$ are overwritten onto A and $\mathbf{b}$ respectively. Also let $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}$ denote the vectors generated in Algorithm 1.1. This algorithm finds a particular solution to the system $U\mathbf{x} = \mathbf{c}$ or equivalently, $A\mathbf{x} = \mathbf{b}$, if it is consistent; all free variables are set to unity.*

Set $k = n$ and set $x_j = 1$ for $j = 1, \ldots, n$.

Do while $k \geq 1$:

if $\sigma_k > n$ then:

if $b_{\gamma_k} \neq 0$, exit and indicate that the system is inconsistent;

else: set

$$x_{\sigma_k} = \frac{1}{a_{\gamma_k,\sigma_k}} \left(b_{\gamma_k} - \sum_{\sigma_k+1}^{n} a_{\gamma_k,j} x_j \right) ;$$

set $k \leftarrow k - 1$

□

Thus the combination of Algorithm 1.1 and the generalized back substitution given in Algorithm 1.2 enables one to find a particular solution of any consistent linear system of algebraic equations.

Example 1.10 Consider the system $A\mathbf{x} = \mathbf{b}$ of Example 1.3.2 which has been reduced to the upper triangular system

$$\begin{pmatrix} 1 & -2 & 1 & -4 \\ 0 & -10 & -12 & -12 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} .$$

This is a consistent system so that in the generalized back substitution algorithm we set $x_4 = x_3 = 1$ and solve for x_2 and then x_1. Thus a particular solution is $x_1 = -\frac{4}{5}$, $x_2 = -\frac{12}{5}$, $x_3 = 1$, and $x_4 = 1$.

1.3.3 Solving linear systems using triangular factorizations

In this section we first present algorithms for determining triangular factorizations of the types $PA = LU$, $A = LU$, and $A = LL^T$. Then we see how these algorithms can be employed to solve linear systems.

Triangular factorizations without pivoting

The elements of the matrices L and U appearing in (1.35) may be obtained as one proceeds through the reduction process of Proposition 1.12, *e.g.*, L is defined by (1.36) and $U = A^{(\ell+1)}$ is the end product of the process. However, the elements of L and U may also be obtained directly by equating, element by element, the left- and right-hand sides of the equation $A = LU$. We have that

$$a_{i,j} = \sum_{k=1}^{K} l_{i,k} u_{k,j} , \quad i = 1, \ldots, n \text{ and } j = 1, \ldots, n , \tag{1.27}$$

where $K = \min(i, j)$, $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Noting that $l_{i,i} = 1$ for $i = 1, \ldots, n$, we see that (1.27) represents n^2 equations for the n^2 nontrivial unknowns $l_{i,k}$ and $u_{k,j}$ where $k = 1, \ldots, n$, $i = k+1, \ldots, n$ and $j = k, \ldots, n$, *i.e.*, those which are not known *a priori* to be one or zero. From (1.27) explicit formulas for these unknowns are easily found and are given in the following algorithm for triangular factorization without pivoting. This algorithm is known as the *Doolittle reduction* of a matrix into triangular factors.

Algorithm 1.3 Triangular factorization without pivoting. *Let A be an $n \times n$ matrix which possesses an LU factorization $A = LU$ such that $u_{k,k} \neq 0$ for $k = 1, \ldots n-1$. This algorithm computes the factors L and U whenever no pivoting, i.e., no interchange of rows, is necessary. All elements of L and U that are not explicitly computed vanish. (The algorithm fails if $u_{k,k} = 0$ for some k such that $1 \leq k \leq n-1$; see Example 1.4.)*

For $k = 1, \ldots, n$, set $l_{k,k} = 1$.

For $j = 1, \ldots, n$, set $u_{1,j} = a_{1,j}$.

For $i = 2, \ldots, n$, set $l_{i,1} = a_{i,1}/u_{1,1}$.

For $k = 2, \ldots, n$, set

$$u_{k,j} = a_{k,j} - \sum_{\ell=1}^{k-1} l_{k,\ell} u_{\ell,j} \quad \text{for } j = k, \ldots, n$$

$$l_{i,k} = \frac{1}{u_{k,k}} \left(a_{i,k} - \sum_{\ell=1}^{k-1} l_{i,\ell} u_{\ell,k} \right) \quad \text{for } i = k+1, \ldots, n\,.$$

□

An examination of the algorithm reveals that one first solves the equations corresponding to the first row of A, then those corresponding to the first column, then the second row, then the second column, etc. Note that $u_{k,k} \neq 0$ for $k = 1, \ldots, \min(m-1, n)$ is a sufficient condition for the algorithm to proceed without row interchanges; these are exactly the denominators needed in the computations of the elements of L.

The operation count for Algorithm 1.3 for is approximately $n^3/2 - n^3/6 = n^3/3$ multiplications and a like number of additions.

As mentioned in Section 1.2, there are several variants to the factorization $A = LU$ given in Proposition 1.12 and algorithms analagous to the Doolittle reduction can be generated for these variants. For example, in the *Crout reduction* algorithm we choose U to be unit upper triangular and L to be lower triangular in the decomposition $A = LU$. To generate the equations for L and U in this case we solve for a column before solving for the corresponding row. Another variant is discussed in Exercise 3.15.

For the sake of clarity, in Algorithm 1.3 we have kept distinct the roles of the elements of A, L and U. If we do not wish to save the matrix A then the results of the decomposition of A into triangular factors may be written over the corresponding element of A. For example, we have, after the k-th stage, the following storage scheme:

$$\begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,k} & u_{1,k+1} & \cdots & u_{1,n} \\ l_{2,1} & u_{2,2} & \cdots & u_{2,k} & u_{2,k+1} & \cdots & u_{2,n} \\ l_{3,1} & l_{3,2} & \cdots & u_{3,k} & u_{3,k+1} & \cdots & u_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{k,1} & l_{k,2} & \cdots & u_{k,k} & u_{k,k+1} & \cdots & u_{k,n} \\ l_{k+1,1} & l_{k+1,2} & \cdots & l_{k+1,k} & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{m,1} & l_{m,2} & \cdots & l_{m,k} & a_{m,k+1} & \cdots & a_{m,n} \end{pmatrix}.$$

Of course, $l_{i,i} = 1$ need not be stored at all.

In the case where A is symmetric and positive definite, the elements of the Cholesky factor L may be computed by the following algorithm for the *Cholesky factorization of a positive definite symmetric matrix*.

Algorithm 1.4 Cholesky factorization of a positive definite symmetric matrix. *Let A be an $n \times n$ positive definite symmetric matrix. The following algorithm generates the factor L in the factorization $A = LL^T$.*

Set $l_{1,1} = \sqrt{a_{1,1}}$ and, for $i = 2, \ldots, n$, set $l_{i,1} = a_{i,1}/l_{1,1}$.

For $k = 2, \ldots, n$ set

$$l_{k,k} = \left(a_{k,k} - \sum_{\ell=1}^{k-1} |l_{k,\ell}|^2 \right)^{1/2} \quad \text{and}$$

for $i = k+1, \ldots, n$ set

$$l_{i,k} = \frac{1}{l_{k,k}} \left(a_{i,k} - \sum_{\ell=1}^{k-1} l_{i,\ell} \bar{l}_{k,\ell} \right).$$

□

Because of the square roots involved, it is sometimes preferable to use the LDL^T factorization for a symmetric positive definite matrix instead of the Cholesky factorization LL^T. (The former factorization may be effected without any square roots.)

If one attempts to apply the Cholesky Algorithm 1.4 to a symmetric matrix A which is not positive definite, then for some k the argument of the square root will be non-positive. Thus, either $l_{k,k} = 0$ or $l_{k,k}$ will be imaginary. This can serve as a test of whether or not a given symmetric matrix is positive definite.

We note that the operation count for the Cholesky factorization is approximately half of the count for the factorization $A = LU$, *i.e.*, for an $n \times n$ matrix the leading term is $n^3/6$ multiplications and a like number of additions.

From Algorithm 1.4 we see that for any $k = 1, \ldots, n$,

$$\sum_{\ell=1}^{k} |l_{k,\ell}|^2 = a_{k,k}$$

so that

$$|l_{k,i}|^2 \le a_{k,k} \quad \text{for } i = 1, \ldots, k\,.$$

Thus the elements of the Cholesky factor L are bounded in terms of the square roots of the diagonal elements of the given matrix A. This fact has important implications regarding the numerical stability of the Cholesky factorization (1.19).

Triangular factorizations with pivoting

We now consider the direct computation of the factorization $PA = LU$ given in Theorem 1.4. Since the pivoting strategy is not known at the start of the computation, *i.e.*, we do not know which pivot elements will vanish or be small, we cannot simply factor the matrix PA. Indeed, the pivoting strategy is determined as one proceeds through the reduction procedure. We obtain the following algorithm for the Doolittle reduction with partial pivoting and implicit row scaling.

Algorithm 1.5 Triangulation factorization with partial pivoting and implicit row scaling. *Let A be an $n \times n$ matrix. This algorithm computes the factors L and U and the permutation matrix P in the factorization $PA = LU$ and also the rank r of the matrix A. The algorithm uses partial pivoting and implicit row scaling as discussed in Section* **??**.

Set $k = 1$, $\sigma_1 = 1$, and $r = 0$.

For $i = 1, \ldots, n$, set $\gamma_i = i$ and $s_i = \sum_{j=1}^{n} |a_{i,j}|\,.$

Do while $k \le n$, $\sigma_k \le n$, and $\sum_{i=k}^{n} s_{\gamma_i} \ne 0$:

set $c = 0$;

do while $c = 0$:

for $i = k, \ldots, n$, set $d_{\gamma_i} = a_{\gamma_i,\sigma_k} - \sum_{t=1}^{k-1} l_{\gamma_i,t} u_{\gamma_t,\sigma_k}$;

set $c = \max_{\substack{i=k,\ldots,n \\ s_{\gamma_i} \ne 0}} \dfrac{|d_{\gamma_i}|}{s_{\gamma_i}}$;

set $\sigma_k \leftarrow \sigma_k + 1$;

set $p =$ smallest integer such that $\dfrac{|d_{\gamma_p}|}{s_{\gamma_p}} = c$;

set $r \leftarrow r + 1$;

if $k \neq p$, interchange the contents of γ_p and γ_k ;

set $u_{\gamma_k,\sigma_k} = d_{\gamma_k}$;

for $j = \sigma_k + 1, \ldots, n$, set

$$u_{\gamma_k,j} = a_{\gamma_k,j} - \sum_{t=1}^{k-1} l_{\gamma_k,t} u_{\gamma_t,j} ;$$

for $i = k + 1, \ldots, n$, set

$$l_{\gamma_i,k} = \frac{d_{\gamma_i}}{d_{\gamma_k}} ;$$

set $\sigma_{k+1} = \sigma_k + 1$;

set $k \leftarrow k + 1$.

For $i = k, \ldots, n$, set $\sigma_i = n + 1$.

□

Note that the algorithm does not involve the physical interchange of rows of the partially reduced matrices; the pivoting strategy is recorded in the integer γ_k. At any stage, $\gamma_i = j$ indicates that the i-th row of the partially determined matrices L and U are found in the j-th row of storage. The integers σ_k are used to keep track of the columns in which the pivot elements, *i.e.*, the leading nonzero entries, appear in any row. For the k-th row, the pivot element is in column σ_k. Again, in Algorithm 1.5, we have kept distinct the roles of the elements of A, L, and U. However, once again one may overwrite the elements of A with the nontrivial elements of L and U, *e.g.*, by using the overwriting scheme

$$\begin{aligned} d_{\gamma_i} &\rightarrow (\gamma_i, k + \sigma_k) \text{ position in storage}, \quad i = k, \ldots, n , \\ u_{\gamma_k,j} &\rightarrow (\gamma_k, j) \text{ position in storage}, \quad j = k + \sigma_k, \ldots, n , \\ l_{\gamma_i,k} &\rightarrow (\gamma_i, k + \sigma_k) \text{ position in storage}, \quad i = k + 1, \ldots, n . \end{aligned}$$

In particular, no extra storage is required for the d_{γ_i}'s and thus Algorithm 1.5 has storage requirements the same as that for the original matrix A, plus two integer arrays of length m for the γ_i's and the σ_i's, and, if scaling is used, another array of length m for the scale factors s_i.

These points are illustrated by the following example.

Example 1.11 Let A be an $n \times n$ matrix for which we are using Algorithm 1.5 to obtain its $PA = LU$ factorization. We illustrate the storage at the end of the third stage of reduction, *i.e.*, after the calculations with $k = 3$ are carried out. Suppose

the pivoting strategy of the three steps is found to be $p_1 = 3$, $p_2 = 5$ and $p_3 = 5$ so that $\gamma_1 = 3$, $\gamma_2 = 5$, $\gamma_3 = 2$, $\gamma_4 = 4$ $\gamma_5 = 1$, and $\gamma_i = i$ for $i > 5$. Also, assume that $\sigma_1 = 1$, $\sigma_2 = 3$, and $\sigma_3 = 4$ so that the columns containing the pivot elements in first three rows have indices 1, 3, and 4, respectively. We then have the following storage scheme at the beginning of the fourth step:

$$\begin{pmatrix} l_{5,1} & l_{5,2} & l_{5,3} & \oplus & a_{5,5} & \cdots & a_{5,n} \\ l_{3,1} & l_{3,2} & \times & u_{3,4} & u_{3,5} & \cdots & u_{3,n} \\ u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & \cdots & u_{1,n} \\ l_{4,1} & l_{4,2} & l_{4,3} & \times & a_{4,5} & \cdots & a_{4,n} \\ l_{2,1} & \times & u_{2,3} & u_{2,4} & u_{2,5} & \cdots & u_{2,n} \\ l_{6,1} & l_{6,2} & l_{6,3} & \oplus & a_{6,5} & \cdots & a_{6,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n,1} & l_{n,2} & l_{n,3} & & a_{n,5} & \cdots & a_{n,n} \end{pmatrix}.$$

The elements denoted by the symbol $\oplus$ will be filled in later with the elements of the fourth column of L. The elements denoted by the $\times$ symbol should be zero due to the fact that $\sigma_k > k$ for $k \geq 2$. In Algorithm 1.5 these zeros are not always actually computed (this would be wastefull), so that these locations in storage do not necessarily contain zeros. However, in any subsequent use of the factorization effected by Algorithm 1.5, these storage locations are never accessed.

We note that it is sometimes recommended that the d_{γ_i}'s be accumulated in double precision in order to minimize the effects of roundoff error.

After completing Algorithm 1.5, a pivoting strategy is known so that *a posterori* we may form the matrix PA. Interestingly, if we proceed to factor PA using Algorithm 1.3 (suitably amended as indicated in Exercise 1.17), it can be shown that the factorization so obtained, *i.e.*, the L and the U, is the same as that obtained from Algorithm 1.5. Thus if one knew a good pivoting strategy beforehand, and reordered the rows of the original matrix A following that strategy, then the reordered matrix can be factored without invoking row interchanges.

We note that the leading term in the operation count for the factorization $PA = LU$ is the same as that for $A = LU$. This is due to the fact that the extra work required for the pivot search is of the order $n^2 - n^2/2$ which is of lower order than the leading term in the operation count for the factorization. The process of row interchanges is often referred to as *row pivoting*. Since, for the most part, we will not consider column interchanges, we will henceforth refer to row pivoting as simply *pivoting*. The (k, σ_k) entry in the matrix $A^{(k)}$ defined in the above proof is referred to as the *pivot element* and (k, σ_k) itself is referred to as the *pivot position*. The pivot element is the denominator appearing in the vector $\boldsymbol{\mu}$ that determines the Gauss transformation $M^{(k)}$ used at the k-th stage. Thus, an interchange of rows is, in theory, only necessary whenever a pivot element vanishes. The row interchange process, *i.e.*, premultiplication by the elementary permutation matrix $P_{(k,p_k)}$, is invoked so that a nonzero entry is brought into the pivot position.

Using triangular factorizations to solve linear systems

The algorithms for triangular factorizations described above can also be used for solving linear systems. In fact, triangular factorization algorithms are the most efficient algorithms for solving a sequence of several systems of equations having the same coefficient matrix but having different right-hand side vectors. In this situation one computes the factorization of the coefficient matrix once and then performs two solves (a backward and forward) for each right-hand side. This is in contrast to Algorithm 1.1 in which the modifications to the right-hand side are performed at the same time the matrix is being reduced.

As an immediate consequence of Theorem 1.4, we have the following proposition.

Proposition 1.8 *Let A be a given $n \times n$ matrix A and $\mathbf{b}$ an n-vector. Then the linear system $A\mathbf{x} = \mathbf{b}$ is equivalent to the linear system*

$$LU\mathbf{x} = P\mathbf{b}\,, \tag{1.28}$$

where the matrices L, U, and P are defined by the factorization $PA = LU$ in Theorem 1.4.

Once P, L, and U are determined in the factorization $PA = LU$, we may solve the system $LU\mathbf{x} = P\mathbf{b}$, or equivalently $A\mathbf{x} = \mathbf{b}$, by applying a a *forward solve* to

$$L\mathbf{y} = P\mathbf{b} \tag{1.29}$$

to determine $\mathbf{y}$ and then determining $\mathbf{x}$ by applying a *back solve* to

$$U\mathbf{x} = \mathbf{y} \tag{1.30}$$

using Algorithm 1.2. The forward solve is given by the following algorithm where we assume that L has been generated by Algorithm 1.5.

Algorithm 1.6 Forward substitution procedure. *Let L be an $n \times n$ unit lower triangular matrix generated by Algorithm 1.5 and let $\mathbf{b} \in \mathbf{R}^n$. Then this algorithm solves the unit lower triangular system $L\mathbf{y} = P\mathbf{b}$ where the row interchange information is stored in the vector $\boldsymbol{\gamma}$, which is output from Algorithm* 1.5.

Set $\quad y_1 = b_1$.

For $i = 2, \ldots, n$, set $\quad y_i = \left(b_{\gamma_i} - \sum_{j=1}^{i-1} l_{\gamma_i, j} y_j \right)$.

□

We note that if no pivoting was performed to calculate the LU factorization then $P = I$ and this forward solve algorithm is still valid since in this case $\gamma_i = i$. However, this algorithm is not valid for the Cholesky factorization LL^T since in this

case L is not *unit* lower triangular. The modifications to Algorithm 1.6 for the case of L being an arbitrary lower triangular matrix is left as an exercise.

Example 1.12 Consider the system $A\mathbf{x}=\mathbf{b}$ where

$$A = \begin{pmatrix} 2 & -1 & 0 \\ 4 & -5 & 3 \\ 6 & -6 & -2 \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}.$$

From Example 1.2.1, A has the LU factorization

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -5 \end{pmatrix}.$$

We thus solve the system $L\mathbf{y} = \mathbf{b}$ to obtain $\mathbf{y}= (1, 0, -5)^T$ and the system $U\mathbf{x}=\mathbf{y}$ to obtain the solution $\mathbf{x}= (1, 1, 1)^T$.

Operation counts

We now turn to the operation counts for solving linear systems. We stated that the LU factorization required approximately $(n^3)/3$ multiplications and a like number of additions. The number of operations for the Gaussian elimination algorithm is the same as for the LU factorization except we must also include the work required to transform the right-hand side. However, this requires $(n^2 + n)/2$ multiplications and a like number of additions so that it does not affect the leading term. The back substitution requires another $(n^2 - n)/2$ multiplications and a like number of additions. Thus the total to solve an $n \times n$ system by Gaussian elimination is approximately $(n^3)/3$ multiplications and a like number of additions. To solve a linear system using the LU factorization we see that we need to do both a forward and back solve. However, the forward solve requires approximately $n^2/2$ multiplications and the back solve approximately $n^2/2$ so that the leading term in the operation count is the same, which, of course, is what we expect.

At the beginning of this section we mentioned that using triangular factorizations for solving linear systems is especially useful when one wants to solve a sequence of linear systems having the same coefficient matrix. Consider finding $\mathbf{x}^{(k)}$ such that $A\mathbf{x}^{(k)} = \mathbf{b}^{(k)}$ for $k = 1, \ldots, K$, where $\{\mathbf{b}^{(k)}\}$ denotes a sequence of right-hand side vectors that are not known *a priori*. For example, the right-hand side vector $\mathbf{b}^{(k)}$ for the k-th linear system could be a function of the solution vector $\mathbf{x}^{(k-1)}$ of the $(k - 1)$-st linear system. Then, for example, if A is a square matrix, re-eliminating for every k requires, for large n approximately $Kn^3/3$ multiplications and a like number of additions or subtractions in order to determine the sequence $\{\mathbf{x}^{(k)}\}$. However, if we compute the factors L and U and the pivoting strategy P and then solve for $\mathbf{x}^{(k)}$ by $L\mathbf{y}^{(k)} = P\mathbf{b}^{(k)}$ and $U\mathbf{x}^{(k)} = \mathbf{y}^{(k)}$, $k = 1, \ldots, K$, the multiplication or addition count reduces to approximately $n^3/3 + Kn^2$, the first term accounting for the determination of the unchanging factors L and U and the second term accounting for the K forward and backward solves.

1.3.4 Calculation of determinants

The determinant of a matrix may be computed as a by-product of triangular factorization or the Gaussian elimination algorithm. From the factorization $PA = LU$ where L is unit lower triangular, we see that $\det(PA) = \det U$. But $\det(PA)$ is either equal to $\det A$ or equal to $-\det A$ since interchanging two rows of a matrix changes only the sign of the determinant. Then it follows that

$$\det A = (-1)^{\xi} \det U = (-1)^{\xi} \prod_{i}^{n} u_{i,i} \,, \tag{1.31}$$

where ξ denotes the total number of row interchanges performed in transforming the matrix A into the matrix U. The value of ξ may be easily accumulated during the elimination procedure. Of course, if $A = LU$, *i.e.*, no row interchanges were performed, then

$$\det A = \prod_{i=1}^{n} u_{i,i} \,.$$

Also we note that if A is a Hermitian positive definite matrix then as a byproduct of the Cholesky factorization $A = LL^*$ we have

$$\det A = |l_{1,1}|^2 |l_{2,2}|^2 \cdots |l_{n,n}|^2 \,,$$

where $l_{i,j}$ are the entries of the matrix L.

Similarly, in the Gaussian elimination algorithm the $\det A$ is given by (1.31) since we have obtained U by applying elementary row operations to A. The only effect of a row interchange is to change the sign of the determinant while a row replacement operation, *i.e.*, the replacement of a row by the sum of that row and multiple another row, does not change the value of the determinant.

1.4 Triangular factorizations for $m \times n$ matrices

In this section we state the results for an $m \times n$ matrix where its entries can be complex, analogous to those proved in Section 1.2 for a square matrix. The proofs are left to the exercises.

We will refer to certain matrices as having the following special structure.

Definition 1.3 A matrix is said to have *row echelon structure* if it differs from a row echelon matrix only in that the first nonzero entry of any row need not be a 1.

A matrix having row echelon structure is clearly upper trapezoidal; the converse is not necessarily true.

The goal of this section is to give the result that any $m \times n$ matrix A has the factorization $PA = LU$ where P is an $m \times m$ permutation matrix, L is an $m \times m$ unit lower triangular matrix, and U is an $m \times n$ matrix having row echelon structure. The first results show show, through the use of elementary permutation matrices and Gauss transformations, a given $m \times n$ matrix A can be reduced to a matrix U having row echelon structure .

Proposition 1.9 *Let A be a given $m \times n$ matrix. Then there exist an integer ℓ, $0 \le \ell \le \min(m-1, n)$, Gauss transformation matrices $M^{(k)}$, $k = 1, \ldots, \ell$, and elementary permutation matrices $P_{(k,p_k)}$, $k = 1, \ldots, \ell$, $k \le p_k \le m$, such that*

$$U = A^{(\ell+1)} = M^{(\ell)} P_{(\ell,p_\ell)} \cdots M^{(2)} P_{(2,p_2)} M^{(1)} P_{(1,p_1)} A \tag{1.32}$$

is an $m \times n$ matrix having row echelon structure.

The following result provides the general factorization.

Theorem 1.10 *Given any $m \times n$ matrix A there exists an $m \times m$ permutation matrix P, an $m \times m$ unit lower triangular matrix L, and an $m \times n$ matrix U having row echelon structure such that*

$$PA = LU\,. \tag{1.33}$$

Furthermore, $\operatorname{rank}(U) = \operatorname{rank}(A)$ *and if A is real, then L and U may be chosen to be real as well.*

Let $r = \operatorname{rank}(U) = \operatorname{rank}(A)$ denote the number of nonzero rows of U. It is, of course, possible for $r < m$, *e.g.*, if $m > n$ or in other cases for which the rows of A are linearly dependent. If $r < m$, the $PA = LU$ factorization of the $m \times n$ matrix A may be partitioned in the form

$$PA = \begin{pmatrix} L_1 & L_2 \end{pmatrix} \begin{pmatrix} U_1 \\ 0 \end{pmatrix} = L_1 U_1\,, \tag{1.34}$$

where L_1 is an $m \times r$ unit lower trapezoidal matrix, L_2 is $m \times (m-r)$, and U_1 is an $r \times n$ full rank matrix having row echelon structure. Thus (1.34) shows that an arbitrary $m \times n$ matrix with $m > r$ can be factored into the product of an $m \times r$ unit lower trapezoidal matrix and an $r \times n$ matrix with row echelon structure. Note that L_2 plays no essential role in the $PA = LU$ factorization of A. Also, if $r = n$, *i.e.*, A has full column rank, then U_1 is an $n \times n$ square, nonsingular, upper triangular matrix; if $r = m$, *i.e.*, A has full row rank, then $L = L_1$ and $U = U_1$.

Proposition 1.11 *Given an $m \times n$ matrix A. Partition its $PA = LU$ factorization as in* (1.34) *where U_1 has full row rank and L_1 is unit lower trapezoidal. Then, once the row interchange strategy is fixed, i.e., the permutation matrix P is fixed, the matrices L_1 and U_1 appearing in the factorization* (1.34) *are uniquely determined. If $r = \operatorname{rank}(A)$, the number of rows in U_1, then L_2 may be chosen to be any $m \times (m-r)$ matrix such that $L = (L_1\ L_2)$ is a unit lower triangular matrix.*

The following results considers the case when we can perform the decomposition without row interchanges.

Proposition 1.12 *Given an $m \times n$ matrix A, denote its leading principal $k \times k$ submatrices by $\mathcal{A}_k$ for $k = 1, \ldots, \min(m, n)$. If $\mathcal{A}_k$ is nonsingular for $k = 1, \ldots, \ell =$*

$\min(m-1,n)$, *then there exists an* $m \times m$ *unit lower triangular matrix* L *and an* $m \times n$ *upper trapezoidal matrix* U *such that*

$$A = LU\,, \tag{1.35}$$

where $U = A^{(\ell+1)}$ *and* L *is given explicitly by*

$$L = (L_1\ L_2)\,, \quad \textit{where} \quad L_1 = \begin{pmatrix} 1 & & & \\ \mu_2^{(1)} & 1 & & \\ \mu_3^{(1)} & \mu_3^{(2)} & & \\ & & \ddots & \\ & & & 1 \\ \vdots & \vdots & & \mu_{\ell+1}^{(\ell)} \\ & & & \vdots \\ \mu_m^{(1)} & \mu_m^{(2)} & \cdots & \mu_m^{(\ell)} \end{pmatrix} \tag{1.36}$$

and where L_2 *is any* $m \times (m-\ell)$ *matrix such that* $L = (L_1\ L_2)$ *is an unit upper triangular matrix. Here* $A^{(k)}$, $k = 1, \ldots, (\ell+1)$, *is defined in the proof of Proposition* (1.3) *and*

$$\mu_j^{(k)} = \frac{a_{j,k}^{(k)}}{a_{k,k}^{(k)}}\,. \tag{1.37}$$

Moreover, $u_{i,i} \neq 0$ *for* $i = 1, \ldots, \min(m-1,n)$. *If* A *is real then* U *and* L *may be chosen to be real as well.*

Exercises

1.1 Let $P_{(k,\ell)}$ be defined by (1.1). Show that (1.2) holds.

1.2 It is not efficient to store an entire $m \times m$ permutation matrix P on a computer. Describe an efficient way to give sufficient information to describe a permutation matrix.

1.3 Show that if $M^{(j)}$ is given by (1.3), then

$$\left(M^{(j)}\right)^{-1} = I + \boldsymbol{\mu}(\mathbf{e}^{(j)})^T\,.$$

Given that $M^{(j)}$ and $M^{(\ell)}$ have the form (1.5) with possibly different values of q, determine $(M^{(j)})^{-1}(M^{(\ell)})^{-1}$ for $j < \ell$.

1.4 Show that a permutation matrix is unitary. Show that the inverse of a permutation matrix is a permutation matrix. Show that the product of two permutation matrices is a permutation matrix.

1.5 Give an example of a nontrivial matrix that fails to have a unique LU factorization even if the factorization can be obtained without any row interchanges. Prove or disprove the assertion that if a square matrix A has an $A = LU$ factorization, then A is invertible.

1.6 Suppose an $m \times n$ matrix A has an $A = LU$ factorization. State and prove the most general uniqueness result for the factors L and U.

1.7 Prove that if A has the factorization $A = LU$ and A is nonsingular then the factorization is unique.

1.8 Show that if an $m \times n$ matrix A has an $A = LU$ factorization, then L is given by (1.36).

1.9 Write down the equations for the Crout factorization $A = LU$ where U is unit triangular and L is lower trapezoidal.

1.10 Let A be a $n \times n$ Hermitian matrix all of whose leading principal matrices are nonsingular. Prove that there exists a unique unit lower triangular matrix L and a real diagonal matrix D such that $A = LDL^*$.

1.11 Let A be an $n \times n$ positive definite matrix. Prove that there exists a unit lower triangular matrix L, a unit upper triangular matrix U, and a diagonal matrix D such that $A = LDU$ where the diagonal entries of D have positive real parts and L, D, and U are uniquely determined.

1.12 Write down an algorithm for the LDL^* decomposition of a Hermitian matrix.

1.13 Write down an algorithm of the LDU decomposition of a positive definite matrix.

1.14 Derive a storage scheme for Algorithm 1.4 that takes advantage of the Hermitian structure of A and of its Cholesky factors.

1.15 Proposition 1.12 gives sufficient conditions for a matrix to have an LU decomposition without pivoting. Give necessary conditions on the matrix A so that is has a factorization $A = LU$. Modify Algorithm 1.3 to handle the general case.

1.16 Let A be an $m \times n$ matrix. Verify the operation count for the factorization $A = LU$ for the case $m > n$. Determine the operation count for the case $m < n$.