Numerical Analysis

10th ed

R L Burden, J D Faires, and A M Burden

Beamer Presentation Slides
Prepared by
Dr. Annette M. Burden
Youngstown State University

September 6, 2015

Operations

$$
\begin{aligned}
E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1,\\
E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2,\\
&\qquad\qquad\qquad\vdots\\
E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n.
\end{aligned}
$$

is a linear system with given constants a_{ij}, for each $i, j = 1, 2, \ldots, n$, and b_i, for each $i = 1, 2, \ldots, n$, and we need to determine the unknowns $x_1, \ldots, x_n$.

1. Equation E_i can be multiplied by any nonzero constant λ with the resulting equation used in place of E_i. This operation is denoted $(\lambda E_i) \rightarrow (E_i)$.
2. Equation E_j can be multiplied by any constant λ and added to equation E_i with the resulting equation used in place of E_i. This operation is denoted $(E_i + \lambda E_j) \rightarrow (E_i)$.
3. Equations E_i and E_j can be transposed in order. This operation is denoted $(E_i) \leftrightarrow (E_j)$.

Definition (6.1)

An $\boldsymbol{n} \times \boldsymbol{m}$ ($\boldsymbol{n}$ by $\boldsymbol{m}$) **matrix** is a rectangular array of elements with n rows and m columns in which not only is the value of an element important, but also its position in the array.

The notation for an $n \times m$ matrix will be a capital letter such as A for the matrix and lowercase letters with double subscripts, such as a_{ij}, to refer to the entry at the intersection of the ith row and jth column; that is,

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}.$$

An $n \times (n+1)$ matrix can be used to represent the linear system

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

by constructing the **augmented matrix**

$$[A, \mathbf{b}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix}.$$

.

Gaussian elimination with backward substitution

Through a sequential procedure for $i = 2, 3, \ldots, n-1$ we perform the operation

$$(E_j - (a_{ji}/a_{ii})E_i) \to (E_j) \quad \text{for each } j = i+1, i+2, \ldots, n,$$

provided $a_{ii} \neq 0$. This eliminates (changes the coefficient to zero) x_i in each row below the ith for all values of $i = 1, 2, \ldots, n-1$. The resulting matrix has the form:

$$\tilde{\tilde{A}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & & & \vdots & \vdots \\ 0 & \cdots & 0 & a_{nn} & a_{n,n+1} \end{bmatrix},$$

where, except in the first row, the values of a_{ij} are not expected to agree with those in the original matrix $\tilde{A} = [A, b]$. The matrix $\tilde{\tilde{A}}$ represents a linear system with the same solution set as the original system .

Algorithm 6.1: GAUSSIAN ELIMINATION WITH BACKSUB

To solve the $n \times n$ linear system

$$\begin{array}{lccccccccc} E_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & a_{1,n+1} \\ E_2: & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & a_{2,n+1} \\ \vdots & \vdots & & \vdots & & & & \vdots & & \vdots \\ E_n: & a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & a_{n,n+1} \end{array}$$

INPUT number of unknowns and equations n; augmented matrix $A = [a_{ij}]$, where $1 \leq i \leq n$ and $1 \leq j \leq n+1$.

OUTPUT solution $x_1, x_2, \ldots, x_n$ or message that the linear system has no unique solution.

Algorithm 6.1: GAUSSIAN ELIMINATION WITH BACKSUB

Step 1 For $i = 1, \ldots, n-1$ do Steps 2–4. (*Elimination process.*)

Step 2 Let p be the smallest integer with $i \le p \le n$ and $a_{pi} \neq 0$.
If no integer p can be found
then OUTPUT ('no unique solution exists'); STOP.

Step 3 If $p \neq i$ then perform $(E_p) \leftrightarrow (E_i)$.

Step 4 For $j = i+1, \ldots, n$ do Steps 5 and 6.

Step 5 Set $m_{ji} = a_{ji}/a_{ii}$.

Step 6 Perform $(E_j - m_{ji}E_i) \rightarrow (E_j)$;

Step 7 If $a_{nn} = 0$ then OUTPUT ('no unique solution exists'); STOP.

Step 8 Set $x_n = a_{n,n+1}/a_{nn}$. (*Start backward substitution.*)

Step 9 For $i = n-1, \ldots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] \Big/ a_{ii}$.

Step 10 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP.

Operation Counts

Both the amount of time required to complete calculations and the subsequent round-off error depend on the number of floating-point arithmetic operations needed to solve a routine problem.

Multiplications/divisions

The total number of multiplications and divisions in Algorithm 6.1

$$\frac{2n^3+3n^2-5n}{6}+\frac{n^2+n}{2}=\frac{n^3}{3}+n^2-\frac{n}{3}.$$

Additions/subtractions

The total number of additions and subtractions in Algorithm 6.1

$$\frac{n^3-n}{3}+\frac{n^2-n}{2}=\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}.$$

Partial Pivoting

The simplest strategy is to select an element in the same column that is below the diagonal and has the largest absolute value; specifically, we determine the smallest $p \geq k$ such that

$$|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

and perform $(E_k) \leftrightarrow (E_p)$. In this case no interchange of columns is used.

Algorithm 6.2: GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING

To solve the $n \times n$ linear system

$$\begin{array}{lccccccccc} E_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & a_{1,n+1} \\ E_2: & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & a_{2,n+1} \\ & & & \vdots & & & & & & \vdots \\ E_n: & a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & a_{n,n+1} \end{array}$$

INPUT number of unknowns and equations n; augmented matrix $A = [a_{ij}]$ where $1 \leq i \leq n$ and $1 \leq j \leq n+1$.

OUTPUT solution $x_1, \ldots, x_n$ or message that the linear system has no unique solution.

Step 1 For $i = 1, \ldots, n$ set $NROW(i) = i$. (*Initialize row pointer.*)

Algorithm 6.2: GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING

Step 2 For $i = 1, \ldots, n-1$ do Steps 3–6. (*Elimination process.*)

Step 3 Let p be the smallest integer with $i \leq p \leq n$ and

$|a(NROW(p), i)| = \max_{i \leq j \leq n} |a(NROW(j), i)|$.

(*Notation:* $a(NROW(i), j) \equiv a_{NROW_i, j}$.)

Step 4 If $a(NROW(p), i) = 0$ then OUTPUT ('no unique solution exists'); STOP.

Step 5 If $NROW(i) \neq NROW(p)$ then set $NCOPY = NROW(i)$;

$NROW(i) = NROW(p)$;

$NROW(p) = NCOPY$.

(*Simulated row interchange.*)

Step 6 For $j = i + 1, \ldots, n$ do Steps 7 and 8.

Step 7 Set $m(NROW(j), i) = a(NROW(j), i)/a(NROW(i), i)$.

Step 8 Perform $(E_{NROW(j)} - m(NROW(j), i) \cdot E_{NROW(i)}) \rightarrow (E_{NROW(j)})$.

Step 9 If $a(NROW(n), n) = 0$ then OUTPUT ('no unique solution exists'); STOP.

Algorithm 6.2: GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING

Step 10 Set $x_n = a(NROW(n), n+1)/a(NROW(n), n)$.
(*Start backward substitution.*)

Step 11 For $i = n-1, \ldots, 1$

$$\text{set } x_i = \frac{a(NROW(i), n+1) - \sum_{j=i+1}^{n} a(NROW(i), j) \cdot x_j}{a(NROW(i), i)}.$$

Step 12 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP.

Algorithm 6.3: GAUSSIAN ELIMINATION WITH SCALED PIVOTING

The only steps in this algorithm that differ from those of Algorithm 6.2 are:

Step 1 For $i = 1, \ldots, n$ set $s_i = \max_{1 \le j \le n} |a_{ij}|$;
if $s_i = 0$ then OUTPUT ('no unique solution exists');
STOP.
set $NROW(i) = i$.

Step 2 For $i = 1, \ldots, n-1$ do Steps 3–6. (*Elimination process.*)

Step 3 Let p be the smallest integer with $i \le p \le n$ and

$$\frac{|a(NROW(p), i)|}{s(NROW(p))} = \max_{i \le j \le n} \frac{|a(NROW(j), i)|}{s(NROW(j))}.$$

COMPLETE PIVOTING

Pivoting can incorporate interchange of both rows and columns. **Complete** (or *maximal*) **pivoting** at the kth step searches all the entries a_{ij}, for $i = k, k+1, \dots, n$ and $j = k, k+1, \dots, n$, to find the entry with the largest magnitude. Both row and column interchanges are performed to bring this entry to the pivot position. The total additional time required to incorporate complete pivoting into Gaussian elimination is

$$\sum_{k=2}^{n}(k^2 - 1) = \frac{n(n-1)(2n+5)}{6}$$

comparisons. Complete pivoting is the strategy recommended only for systems where accuracy is essential and the amount of execution time needed for this method can be justified.

Definition (6.2)

Two matrices A and B are **equal** if they have the same number of rows and columns, say $n \times m$, and if $a_{ij} = b_{ij}$, for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

Definition (6.3)

If A and B are both $n \times m$ matrices, then the **sum** of A and B, denoted $A + B$, is the $n \times m$ matrix whose entries are $a_{ij} + b_{ij}$, for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

Definition (6.4)

If A is an $n \times m$ matrix and λ is a real number, then the **scalar multiplication** of λ and A, denoted λA, is the $n \times m$ matrix whose entries are λa_{ij}, for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

We let O denote a matrix all of whose entries are 0.

Theorem (6.5)

Let A, B, and C be $n \times m$ matrices and λ and μ be real numbers. The following properties of addition and scalar multiplication hold:

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

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

(iii) $A + O = O + A = A$,

(iv) $A + (-A) = -A + A = 0$,

(v) $\lambda(A + B) = \lambda A + \lambda B$,

(vi) $(\lambda + \mu)A = \lambda A + \mu A$,

(vii) $\lambda(\mu A) = (\lambda\mu)A$,

(viii) $1A = A$.

All these properties follow from similar results concerning the real numbers.

Definition (6.6)

Let A be an $n \times m$ matrix and $\mathbf{b}$ an m-dimensional column vector. The **matrix-vector product** of A and $\mathbf{b}$, denoted $A\mathbf{b}$, is an n-dimensional column vector given by

$$A\mathbf{b} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} a_{1i} b_i \\ \sum_{i=1}^{m} a_{2i} b_i \\ \vdots \\ \sum_{i=1}^{m} a_{ni} b_i \end{bmatrix}.$$

NOTE: For this product to be defined the number of columns of the matrix A must match the number of rows of the vector $\mathbf{b}$, and the result is another column vector with the number of rows matching the number of rows in the matrix.

Definition (6.7)

Let A be an $n \times m$ matrix and B an $m \times p$ matrix. The **matrix product** of A and B, denoted AB, is an $n \times p$ matrix C whose entries c_{ij} are

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{im} b_{mj},$$

for each $i = 1, 2, \ldots n$, and $j = 1, 2, \ldots, p$.

Theorem (6.8)

Let A be an $n \times m$ matrix, B be an $m \times k$ matrix, C be a $k \times p$ matrix, D be an $m \times k$ matrix, and λ be a real number. The following properties hold:

(a) $A(BC) = (AB)C;$

(b) $A(B + D) = AB + AD;$

(c) $\lambda(AB) = (\lambda A)B = A(\lambda B).$

Definition (6.9)

(i) A **square** matrix has the same number of rows as columns.

(ii) A **diagonal** matrix $D = [d_{ij}]$ is a square matrix with $d_{ij} = 0$ whenever $i \neq j$.

(iii) The **identity matrix of order** n, $I_n = [\delta_{ij}]$, is a diagonal matrix whose diagonal entries are all 1s. When the size of I_n is clear, this matrix is generally written simply as I. □

Definition (6.10)

An **upper-triangular** $n \times n$ matrix $U = [u_{ij}]$ has, for each $j = 1, 2, \ldots, n$, the entries

$$u_{ij} = 0, \quad \text{for each } i = j+1, j+2, \ldots, n;$$

and a **lower-triangular** matrix $L = [l_{ij}]$ has, for each $j = 1, 2, \ldots, n$, the entries

$$l_{ij} = 0, \quad \text{for each } i = 1, 2, \ldots, j-1.$$

Definition (6.11)

An $n \times n$ matrix A is said to be **nonsingular** (or *invertible*) if an $n \times n$ matrix A^{-1} exists with $AA^{-1} = A^{-1}A = I$. The matrix A^{-1} is called the **inverse** of A. A matrix without an inverse is called **singular** (or *noninvertible*).

Theorem (6.12)

For any nonsingular $n \times n$ matrix A:

(i) A^{-1} *is unique.*

(ii) A^{-1} *is nonsingular and* $(A^{-1})^{-1} = A$.

(iii) *If B is also a nonsingular $n \times n$ matrix, then* $(AB)^{-1} = B^{-1}A^{-1}$. □

Definition (6.13)

The **transpose** of an $n \times m$ matrix $A = [a_{ij}]$ is the $m \times n$ matrix $A^t = [a_{ji}]$, where for each i, the ith column of A^t is the same as the ith row of A. A square matrix A is called **symmetric** if $A = A^t$.

Theorem (6.14)

The following operations involving the transpose of a matrix hold whenever the operation is possible:

(i) $(A^t)^t = A$,

(ii) $(A + B)^t = A^t + B^t$,

(iii) $(AB)^t = B^t A^t$,

(iv) *if* A^{-1} *exists, then* $(A^{-1})^t = (A^t)^{-1}$.

Definition (6.15)

Suppose that A is a square matrix.

(i) If $A = [a]$ is a 1×1 matrix, then $\det A = a$.

(ii) If A is an $n \times n$ matrix, with $n > 1$ the **minor** M_{ij} is the determinant of the $(n-1) \times (n-1)$ submatrix of A obtained by deleting the ith row and jth column of the matrix A.

(iii) The **cofactor** A_{ij} associated with M_{ij} is defined by $A_{ij} = (-1)^{i+j} M_{ij}$.

(iv) The **determinant** of the $n \times n$ matrix A, when $n > 1$, is given either by

$$\det A = \sum_{j=1}^{n} a_{ij} A_{ij} = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij}, \quad \text{for any } i = 1, 2, \ldots, n,$$

or by

$$\det A = \sum_{i=1}^{n} a_{ij} A_{ij} = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} M_{ij}, \quad \text{for any } j = 1, 2, \ldots, n.$$

Theorem (6.16)

Suppose A is an $n \times n$ matrix:

- **(i)** *If any row or column of A has only zero entries, then* $\det A = 0$.
- **(ii)** *If A has two rows or two columns the same, then* $\det A = 0$.
- **(iii)** *If $\tilde{A}$ is obtained from A by the operation $(E_i) \leftrightarrow (E_j)$, with $i \neq j$, then* $\det \tilde{A} = -\det A$.
- **(iv)** *If $\tilde{A}$ is obtained from A by the operation $(\lambda E_i) \to (E_i)$, then* $\det \tilde{A} = \lambda \det A$.
- **(v)** *If $\tilde{A}$ is obtained from A by the operation $(E_i + \lambda E_j) \to (E_i)$ with $i \neq j$, then* $\det \tilde{A} = \det A$.
- **(vi)** *If B is also an $n \times n$ matrix, then* $\det AB = \det A \det B$.
- **(vii)** $\det A^t = \det A$.
- **(viii)** *When A^{-1} exists,* $\det A^{-1} = (\det A)^{-1}$.
- **(ix)** *If A is an upper triangular, lower triangular, or diagonal matrix, then* $\det A = \prod_{i=1}^{n} a_{ii}$.

Theorem (6.17)

The following statements are equivalent for any $n \times n$ matrix A:

- **(i)** *The equation $A\mathbf{x} = \mathbf{0}$ has the unique solution $\mathbf{x} = \mathbf{0}$.*
- **(ii)** *The system $A\mathbf{x} = \mathbf{b}$ has a unique solution for any n-dimensional column vector $\mathbf{b}$.*
- **(iii)** *The matrix A is nonsingular; that is, A^{-1} exists.*
- **(iv)** *$\det A \neq 0$.*
- **(v)** *Gaussian elimination with row interchanges can be performed on the system $A\mathbf{x} = \mathbf{b}$ for any n-dimensional column vector $\mathbf{b}$.*

Corollary (6.18)

Suppose that A and B are both $n \times n$ matrices with either $AB = I$ or $BA = I$. Then $B = A^{-1}$ (and $A = B^{-1}$).

Theorem (6.19)

If Gaussian elimination can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ without row interchanges, then the matrix A can be factored into the product of a lower-triangular matrix L and an upper-triangular matrix U, that is, $A = LU$, where $m_{ji} = a_{ji}^{(i)}/a_{ii}^{(i)}$,

$$U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & & \\ & & & a_{n-1,n}^{(n-1)} \\ 0 & & 0 & a_{nn}^{(n)} \end{bmatrix}, \text{ and } L = \begin{bmatrix} 1 & 0 & & 0 \\ m_{21} & 1 & & \\ & & & 0 \\ m_{n1} & & m_{n,n-1} & 1 \end{bmatrix}.$$

Algorithm 6.4: LU FACTORIZATION

To factor the $n \times n$ matrix $A = [a_{ij}]$ into the product of the lower-triangular matrix $L = [l_{ij}]$ and the upper-triangular matrix $U = [u_{ij}]$; that is, $A = LU$, where the main diagonal of either L or U consists of all ones:

INPUT dimension n; the entries a_{ij}, $1 \le i, j \le n$ of A; the diagonal $l_{11} = \cdots = l_{nn} = 1$ of L or the diagonal $u_{11} = \cdots = u_{nn} = 1$ of U.

OUTPUT the entries l_{ij}, $1 \le j \le i$, $1 \le i \le n$ of L and the entries, u_{ij}, $i \le j \le n$, $1 \le i \le n$ of U.

Step 1 Select l_{11} and u_{11} satisfying $l_{11}u_{11} = a_{11}$.
If $l_{11}u_{11} = 0$ then OUTPUT ('Factorization impossible');
STOP.

Step 2 For $j = 2, \ldots, n$ set $u_{1j} = a_{1j}/l_{11}$; (*First row of U.*)
$l_{j1} = a_{j1}/u_{11}$. (*First column of L.*)

Algorithm 6.4: LU FACTORIZATION

Step 3 For $i = 2, \ldots, n-1$ do Steps 4 and 5.

Step 4 Select l_{ii} and u_{ii} satisfying $l_{ii}u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$.

If $l_{ii}u_{ii} = 0$ then OUTPUT ('Factorization impossible');

STOP.

Step 5 For $j = i+1, \ldots, n$

set $u_{ij} = \frac{1}{l_{ii}}\left[a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\right]$; (*ith row of U.*)

$l_{ji} = \frac{1}{u_{ii}}\left[a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki}\right]$. (*ith column of L.*)

Step 6 Select l_{nn} and u_{nn} satisfying $l_{nn}u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk}u_{kn}$.

(*Note: If $l_{nn}u_{nn} = 0$, then $A = LU$ but A is singular.*)

Step 7 OUTPUT (l_{ij} for $j = 1, \ldots, i$ and $i = 1, \ldots, n$);

OUTPUT (u_{ij} for $j = i, \ldots, n$ and $i = 1, \ldots, n$);

STOP.

Permutation matrix

An $n \times n$ **permutation matrix** $P = [p_{ij}]$ is a matrix obtained by rearranging the rows of I_n, the identity matrix. This gives a matrix with precisely one nonzero entry in each row and in each column, and each nonzero entry is a 1.

NOTE: Any nonsingular matrix A can be factored into $A = P^t LU$.

Definition (6.20)

The $n \times n$ matrix A is said to be **diagonally dominant** when

$$|a_{ii}| \geq \sum_{\substack{j=1, \\ j \neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \ldots, n.$$

A diagonally dominant matrix is said to be **strictly diagonally dominant** when the inequality in (6.10) is strict for each n, that is, when

$$|a_{ii}| > \sum_{\substack{j=1, \\ j \neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \ldots, n.$$

Theorem (6.21)

A strictly diagonally dominant matrix A is nonsingular. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form $A\mathbf{x} = \mathbf{b}$ to obtain its unique solution without row or column interchanges, and the computations will be stable with respect to the growth of round-off errors.

Definition (6.22)

A matrix A is **positive definite** if it is symmetric and if $\mathbf{x}^t A\mathbf{x} > 0$ for every n-dimensional vector $\mathbf{x} \neq \mathbf{0}$.

Theorem (6.23)

If A is an $n \times n$ positive definite matrix, then

- **(i)** *A has an inverse;*
- **(ii)** $a_{ii} > 0$, *for each* $i = 1, 2, \ldots, n$;
- **(iii)** $\max_{1 \le k, j \le n} |a_{kj}| \le \max_{1 \le i \le n} |a_{ii}|$;
- **(iv)** $(a_{ij})^2 < a_{ii} a_{jj}$, *for each* $i \ne j$.

Definition (6.24)

A **leading principal submatrix** of a matrix A is a matrix of the form

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix},$$

for some $1 \le k \le n$.

Theorem (6.25)

A symmetric matrix A is positive definite if and only if each of its leading principal submatrices has a positive determinant.

Theorem (6.26)

The symmetric matrix A is positive definite if and only if Gaussian elimination without row interchanges can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ with all pivot elements positive. Moreover, in this case, the computations are stable with respect to the growth of round-off errors.

Corollary (6.27)

The matrix A is positive definite if and only if A can be factored in the form LDL^t, where L is lower triangular with 1s on its diagonal and D is a diagonal matrix with positive diagonal entries.

Corollary (6.28)

The matrix A is positive definite if and only if A can be factored in the form LL^t, where L is lower triangular with nonzero diagonal entries.

Algorithm 6.5: LDL^t FACTORIZATION

To factor the positive definite $n \times n$ matrix A into the form LDL^t, where L is a lower triangular matrix with 1s along the diagonal and D is a diagonal matrix with positive entries on the diagonal:

INPUT the dimension n; entries a_{ij}, for $1 \leq i, j \leq n$ of A.

OUTPUT the entries l_{ij}, for $1 \leq j < i$ and $1 \leq i \leq n$ of L, and d_i, for $1 \leq i \leq n$ of D.

Step 1 For $i = 1, \ldots, n$ do Steps 2–4.
  Step 2 For $j = 1, \ldots, i - 1$, set $v_j = l_{ij} d_j$.
  Step 3 Set $d_i = a_{ii} - \sum_{j=1}^{i-1} l_{ij} v_j$.
  Step 4 For $j = i + 1, \ldots, n$ set $l_{ji} = (a_{ji} - \sum_{k=1}^{i-1} l_{jk} v_k)/d_i$.
Step 5 OUTPUT (l_{ij} for $j = 1, \ldots, i - 1$ and $i = 1, \ldots, n$);
  OUTPUT (d_i for $i = 1, \ldots, n$);
  STOP.

Corollary (6.29)

Let A be a symmetric $n \times n$ matrix for which Gaussian elimination can be applied without row interchanges. Then A can be factored into LDL^t, where L is lower triangular with 1s on its diagonal and D is the diagonal matrix with $a_{11}^{(1)}, \ldots, a_{nn}^{(n)}$ on its diagonal.

Algorithm 6.6: CHOLESKY FACTORIZATION

To factor the positive definite $n \times n$ matrix A into LL^t, where L is lower triangular:

INPUT the dimension n; entries a_{ij}, for $1 \le i, j \le n$ of A.

OUTPUT the entries l_{ij}, for $1 \le j \le i$ and $1 \le i \le n$ of L. (*The entries of* $U = L^t$ *are* $u_{ij} = l_{ji}$, *for* $i \le j \le n$ *and* $1 \le i \le n$.)

Step 1 Set $l_{11} = \sqrt{a_{11}}$.
Step 2 For $j = 2, \ldots, n$, set $l_{j1} = a_{j1}/l_{11}$.
Step 3 For $i = 2, \ldots, n-1$ do Steps 4 and 5.
  Step 4 Set $l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{1/2}$.
  Step 5 For $j = i+1, \ldots, n$ set $l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}\right) / l_{ii}$.
Step 6 Set $l_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2\right)^{1/2}$.
Step 7 OUTPUT (l_{ij} for $j = 1, \ldots, i$ and $i = 1, \ldots, n$);
  STOP.

Definition (6.30)

An $n \times n$ matrix is called a **band matrix** if integers p and q, with $1 < p$, $q < n$, exist with the property that $a_{ij} = 0$ whenever $p \leq j - i$ or $q \leq i - j$. The **band width** of a band matrix is defined as $w = p + q - 1$.

Matrices of bandwidth 3 occurring when $p = q = 2$ are called **tridiagonal** because they have the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & \cdots & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

Algorithm 6.7: CROUT FACTORIZATION TRI DIAG

To solve the $n \times n$ linear system

$$
\begin{array}{lccccccccccc}
E_1: & a_{11}x_1 & + & a_{12}x_2 & & & & & & & = & a_{1,n+1}, \\
E_2: & a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & & & & & = & a_{2,n+1}, \\
\vdots & & & & & \vdots & & & & & \vdots & \\
E_{n-1}: & & & & & a_{n-1,n-2}x_{n-2} & + & a_{n-1,n-1}x_{n-1} & + & a_{n-1,n}x_n & = & a_{n-1,n+1}, \\
E_n: & & & & & & & a_{n,n-1}x_{n-1} & + & a_{nn}x_n & = & a_{n,n+1},
\end{array}
$$

which is assumed to have a unique solution:

INPUT the dimension n; the entries of A.

OUTPUT the solution $x_1, \ldots, x_n$.

(*Steps 1–3 set up and solve* $L\mathbf{z} = \mathbf{b}$.)

Algorithm 6.7: CROUT FACTORIZATION TRI DIAG

Step 1 Set $l_{11} = a_{11}$;
$u_{12} = a_{12}/l_{11}$;
$z_1 = a_{1,n+1}/l_{11}$.

Step 2 For $i = 2, \ldots, n-1$ set $l_{i,i-1} = a_{i,i-1}$; (*ith row of L.*)
$l_{ii} = a_{ii} - l_{i,i-1}u_{i-1,i}$;
$u_{i,i+1} = a_{i,i+1}/l_{ii}$; ($(i+1)$*th column of U.*)
$z_i = (a_{i,n+1} - l_{i,i-1}z_{i-1})/l_{ii}$.

Step 3 Set $l_{n,n-1} = a_{n,n-1}$; (*nth row of L.*)
$l_{nn} = a_{nn} - l_{n,n-1}u_{n-1,n}$.
$z_n = (a_{n,n+1} - l_{n,n-1}z_{n-1})/l_{nn}$.

(*Steps 4 and 5 solve* $U\mathbf{x} = \mathbf{z}$.)

Step 4 Set $x_n = z_n$.

Step 5 For $i = n-1, \ldots, 1$ set $x_i = z_i - u_{i,i+1}x_{i+1}$.

Step 6 OUTPUT $(x_1, \ldots, x_n)$;
STOP.

Theorem (6.31)

Suppose that $A = [a_{ij}]$ is tridiagonal with $a_{i,i-1}a_{i,i+1} \neq 0$, for each $i = 2, 3, \ldots, n-1$. If $|a_{11}| > |a_{12}|$, $|a_{ii}| \geq |a_{i,i-1}| + |a_{i,i+1}|$, for each $i = 2, 3, \ldots, n-1$, and $|a_{nn}| > |a_{n,n-1}|$, then A is nonsingular and the values of l_{ii} described in the Crout Factorization Algorithm are nonzero for each $i = 1, 2, \ldots, n$.