Direct Methods for Solving Linear Systems

Simon Fraser University – Surrey Campus
MACM 316 – Spring 2005
Instructor: Ha Le

Overview

- General Linear Systems
  - Gaussian Elimination
  - Triangular Systems
  - The LU Factorization
  - Pivoting
- Special Linear Systems
  - Strictly Diagonally Dominant Matrices
  - The LDM^T and LDL^T Factorizations
  - Positive Definite Systems
  - Tridiagonal Systems

Linear Systems of Equations

- For a linear system of equations

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & + & a_{1,n}x_n & = & b_1, \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & + & a_{2,n}x_n & = & b_2, \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{n,1}x_1 & + & a_{n,2}x_2 & + & \cdots & + & a_{n,n}x_n & = & b_n; \end{array}$$

or equivalently, in matrix/vector notation: $(A)_{n\times n}(x)_{n\times 1} = (b)_{n\times 1}$:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, \tag{1}$$

find $x = [x_1, x_2, \ldots, x_n]^T$ such that the relation $Ax = b$ holds.

Gaussian Elimination

- A process of reducing the given linear system to a new linear system in which the unknowns x_i's are systematically eliminated;

- The reduction is done via *elementary row operations*;

- It may be necessary to reorder the equations to accomplish this, i.e., use equation pivoting.

Elementary Row Operations

- Type 1: interchange two rows of a matrix: $(i) \leftrightarrow (j)$;
- Type 2: replacing a row by the same row multiplied by a nonzero constant: $(i) \rightarrow \lambda\,(i)$, $\lambda \in \mathbb{R} \setminus \{0\}$;
- Type 3: replacing a row by the same row plus a constant multiple of another row: $(j) \rightarrow (j) + \lambda\,(i)$, $\lambda \in \mathbb{R}$.

Gaussian Elimination: an Example

$$\underbrace{\left[\begin{array}{ccc|c} 1 & 4 & 7 & 30 \\ 2 & 5 & 8 & 36 \\ 3 & 6 & 10 & 45 \end{array}\right]}_{\text{augmented matrix}} \xrightarrow[(3)\to(3)-(a_{3,1}/a_{1,1})(1)]{(2)\to(2)-(a_{2,1}/a_{1,1})(1)} \left[\begin{array}{ccc|c} 1 & 4 & 7 & 30 \\ 0 & -3 & -6 & -24 \\ 0 & -6 & -11 & -45 \end{array}\right]$$

$$\xrightarrow{(3)\to(3)-(a_{3,2}/a_{2,2})(2)} \left[\begin{array}{ccc|c} 1 & 4 & 7 & 30 \\ 0 & -3 & -6 & -24 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

$$x_3 = 3,$$

$$-3x_2 - 6x_3 = -24 \Longrightarrow x_2 = 2,$$

$$x_1 + 4x_2 + 7x_3 = 30 \Longrightarrow x_1 = 1.$$

Triangular Systems: Forward Substitution

Consider the following 2-by-2 *lower triangular system*:

$$\begin{bmatrix} l_{1,1} & 0 \\ l_{2,1} & l_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

If $l_{1,1}l_{2,2} \neq 0$, then $x_1 = b_1/l_{1,1}$, $x_2 = (b_2 - l_{2,1}x_1)/l_{2,2}$.

The general procedure is obtained by solving the ith equation in $Lx = b$ for x_i:

$$x_i = \frac{\left(b_i - \sum_{j=1}^{i-1} l_{i,j}x_j\right)}{l_{i,i}}.$$

Flop count: $\displaystyle\sum_{i=2}^{n} \left(\underbrace{(i-1)}_{\text{mul}} + \underbrace{(i-2)}_{\text{add}} + \underbrace{1}_{\text{sub}} + \underbrace{1}_{\text{div}} \right) = n^2.$

Triangular Systems: Back Substitution

Consider the following 2-by-2 upper triangular system:

$$\begin{bmatrix} u_{1,1} & u_{1,2} \\ 0 & u_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

If $u_{1,1}u_{2,2} \neq 0$, then $x_2 = b_2/u_{2,2}$, $x_1 = (b_1 - u_{1,2}x_2)/u_{1,1}$.

The general procedure is obtained by solving the ith equation in $Ux = b$ for x_i:

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

Flop count: n^2

The Algebra of Triangular Matrices

Definition. A *unit* triangular matrix is a triangular matrix with ones on the diagonal.

Properties.

- The inverse of an upper (lower) triangular matrix is upper (lower) triangular;
- The product of two upper (lower) triangular matrices is upper (lower) triangular;
- The inverse of a unit upper (lower) triangular matrix is a unit upper (lower) triangular;
- The product of two unit upper (lower) triangular matrices is unit upper (lower) triangular.

The LU Factorization

1. Compute a unit lower triangular L and an upper triangular U such that $A = LU$;
2. Solve $Lz = b$ (forward substitution);
3. Solve $Ux = z$ (back substitution).

Example.

$$\underbrace{\begin{bmatrix} 3 & 5 \\ 6 & 7 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 3 & 5 \\ 0 & -3 \end{bmatrix}}_{U}.$$

For $b = [1, 4]^T$, solving $Lz = b$ yields $z = [1, 2]^T$, and solving $Ux = z$ yields $x = [13/9, -2/3]^T$.

LU Factorization: an Example

Example.

$$\underbrace{\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix}}_{A} \begin{matrix} (2)\to(2)-(a_{2,1}/a_{1,1})(1) \\ (3)\to(3)-(a_{3,1}/a_{1,1})(1) \end{matrix} \Longrightarrow \underbrace{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix}}_{A_1}$$

Note that $M_1 \cdot A = A_1$ where M_1 is the unit lower triangular matrix

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{2,1}}{a_{1,1}} & 1 & 0 \\ -\frac{a_{3,1}}{a_{1,1}} & 0 & 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix}}_{A_1} \xrightarrow{(3)\to(3)-(a_{3,2}/a_{2,2})(2)} \Longrightarrow \underbrace{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix}}_{A_2}$$

Note that $M_2 \cdot A_1 = A_2$ where M_2 is the unit lower triangular matrix

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{3,2}}{a_{2,2}} & 1 \end{bmatrix}$$

Hence, $M_2 M_1 A = A_2$, or equivalently, $A = \underbrace{M_1^{-1} M_2^{-1}}_{L} \underbrace{A_2}_{U}$.

Also,

$$M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{2,1}}{a_{1,1}} & 1 & 0 \\ -\frac{a_{3,1}}{a_{1,1}} & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{2,1}}{a_{1,1}} & 1 & 0 \\ \frac{a_{3,1}}{a_{1,1}} & 0 & 1 \end{bmatrix},$$

$$M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{3,2}}{a_{2,2}} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{a_{3,2}}{a_{2,2}} & 1 \end{bmatrix},$$

$$L = M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{2,1}}{a_{1,1}} & 1 & 0 \\ \frac{a_{3,1}}{a_{1,1}} & \frac{a_{3,2}}{a_{2,2}} & 1 \end{bmatrix}.$$

Gauss Transformation

Suppose $x \in \mathbb{R}^n$ with $x_k \neq 0$. If

$$\tau^T = [\underbrace{0, \ldots, 0}_{k}, \tau_{k+1}, \ldots, \tau_n], \quad \tau_i = \frac{x_i}{x_k} \quad \text{for } (k+1) \leq i \leq n$$

and we define $M_k = I - \tau\, e_k^T$, where $e_k^T = [\underbrace{0, \ldots, 0}_{k-1}, 1, 0, \ldots, 0]$,

$$M_k\, x = \begin{pmatrix} 1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & & 1 & 0 & & 0 \\ 0 & & -\tau_{k+1} & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & -\tau_n & 0 & \ldots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

M_k: Gauss transformation; $\tau_{k+1}, \ldots, \tau_n$: multipliers.

Upper Triangularizing

Assume that $A \in \mathbb{R}^{n\times n}$, Gauss transformation $M_1, \ldots, M_{n-1}$ can usually be found such that $M_{n-1} \cdots M_2 M_1 A = U$ is upper triangular. During the kth step:

- We are confronted with the matrix $A^{(k-1)} = M_{k-1} \cdots M_1 A$ that is upper triangular in columns 1 to $k-1$;
- The multipliers in M_k are based on $a_{k+1,k}^{(k-1)}, \ldots, a_{n,k}^{(k-1)}$. In particular, we need $a_{k,k}^{(k-1)} \neq 0$ to proceed.

The LU Factorization

- Let $M_1, \ldots, M_{n-1}$ be the Gauss transforms such that $M_{n-1} \cdots M_1 A = U$ is upper triangular. If $M_k = I - \tau^{(k)} e_k^T$, then $M_k^{-1} = I + \tau^{(k)} e_k^T$. Hence,

$$A = LU \qquad \text{where } L = M_1^{-1} \cdots M_{n-1}^{-1}.$$

- L is a unit lower triangular matrix since each M_k^{-1} is unit lower triangular (p.9).

- Let $\tau^{(k)}$ be the vector of multipliers associated with M_k then upon termination, $A[k+1..n, k] = \tau^{(k)}$.

The LU Factorization: an Algorithm Description

`Algorithm` **LUFactorization**(A)

`input:` an n-by-n (square) matrix L

`output:` the LU factorization of A, provided it exists

for i from 1 to n do
    for k from $i+1$ to n do
        mult $:= a_{k,i}/a_{i,i}$;
        $a_{k,i} :=$ mult;
        for j from $i+1$ to n do
            $a_{k,j} := a_{k,j} - \text{mult} \times a_{i,j}$;
        od;
    od;
od;
return A.

The LU Factorization: Flop Count

Definition. A flop is a floating-point operation. There is no standard agreement on this terminology. We shall adopt the MATLAB convention, which is to count the total number of operations. The flop count will include the total number of adds + multiplies + divides + subtracts.

- For LU factorization:

$$\sum_{i=1}^{n}\sum_{k=i+1}^{n}\left(1+\sum_{j=i+1}^{n}2\right)=\frac{2}{3}n^3-\frac{1}{2}n^2-\frac{1}{6}n=O(n^3)\ \text{ flops}$$

- Recall that for forward substitution (p.7) and back substitution (p.8): $O(n^2)$ flops.

The LU Factorization: Sufficient Conditions

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

$$A_k = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,k} \\ a_{2,1} & a_{2,2} & \dots & a_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k,1} & a_{k,2} & \dots & a_{k,k} \end{pmatrix}$$

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

Theorem. $A \in \mathbb{R}^{n \times n}$ has an LU factorization if for $1 \leq k \leq n-1$, $\det(A_k) \neq 0$ where the A_k's are the leading principal submatrices of A. If the LU factorization exists and A is nonsingular, then the LU factorization is unique and $\det(A) = u_{1,1} \cdots u_{n,n}$.

Partial Pivoting

- Row interchanges, i.e., elementary row operation of type 1 (see p.5) are needed when one of the pivot elements $a_{k,k}^{(k)} = 0$.
- Row interchanges are often necessary even when the pivot $\neq 0$:
  - If $|a_{k,k}^{(k)}| \ll |a_{j,k}^{(k)}|$ for some j $(k+1 \leq j \leq n)$, the multiplier $m_{j,k} = a_{j,k}^{(k)}/a_{k,k}^{(k)}$ will be very large. Roundoff error that was produced in the computation of $a_{k,l}^{(k)}$ will be multiplied by the large factor $m_{j,k}$ when computing $a_{j,l}^{(k+1)}$;
  - Roundoff error can also be dramatically increased in the back substitution step $x_k = \dfrac{a_{k,n+1}^{(k)} - \sum_{j=k+1}^{n} a_{k,j}^{(k)}}{a_{k,k}^{(k)}}$ when the pivot $a_{k,k}^{(k)}$ is small.

Example. For the linear system

$$\begin{aligned} E_1 &: \quad 0.003000x_1 + 59.14x_2 = 59.17, \\ E_2 &: \quad 5.291x_1 - 6.130x_2 = 46.78. \end{aligned}$$

$(x_1)_E = 10.00$, $(x_2)_E = 1.000$, $(x_1)_A = -10.00$, $(x_2)_A = 1.001$

(Gaussian elimination, four-digit arithmetic rounding).

- $a_{1,1}^{(1)} = 0.003000 \implies m_{2,1} = 5.291/0.003000 \approx 1764.$

Apply $(E_2) \longrightarrow (E_2 - m_{2,1}E_1)$:

$$\begin{aligned} E_1 &: \quad 0.003000x_1 + 59.14x_2 \approx 59.17, \\ E_2 &: \quad -104300x_2 \approx -104400. \end{aligned}$$

Back substitution $\implies x_2 \approx 1.001$. However, since the pivot $a_{1,1}$ is small, $x_1 \approx (59.17 - (59.14)(1.001))/0.003000 = -10.00$ contains the small error 0.001 multiplied by $59.14/0.003000 \approx 20000$.

Pivoting Strategy

- Select the largest element $a_{i,k}^{(k)}$ that is below the pivot $a_{k,k}^{(k)}$: if $\max_{j=k,\ldots,n} \left|a_{j,k}^{(k)}\right| = \left|a_{k^*,k}^{(k)}\right|$, then swap row k^* with row k, and use $a_{k^*,k}^{(k)}$ to form the multiplier.

Example. For the linear system $\{E_1, E_2\}$ on p.21, since $a_{1,1} < a_{2,1}$, rows 1 and 2 are swapped. This leads to the correct solution $x_1 = 10.00$, $x_2 = 1.0000$.

Permutation Matrices

- A *permutation matrix* is just the identity matrix with its rows re-ordered.
- If P is a permutation and A is a matrix, then PA is a row permuted version of A, and AP is a column permuted version of A, e.g.,

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad PA = \begin{bmatrix} a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \end{bmatrix}.$$

- If P is a permutation, then $P^{-1} = P^T$ (P is *orthogonal*).

- An *interchange permutation* is obtained by merely swapping two rows in the identity.

- If $P = E_n \cdots E_1$ and each E_k is the identity with rows k and $p(k)$ interchanged, then the vector $p(1:n)$ is a useful vector encoding of P, e.g., $[4, 4, 3, 4]$ is the vector encoding of P on p.23.

- No floating point arithmetic is involved in a permutation operation. However, permutation matrix operations often involve the irregular movement of data, and can represent a significant computational overhead.

Partial Pivoting: the Basic Idea

$$A = \begin{bmatrix} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{bmatrix}$$

- $(1) \leftrightarrow (3)$, $p[1] = 3$:

$$E_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad E_1 A = \begin{bmatrix} 6 & 18 & -12 \\ 2 & 4 & -2 \\ 3 & 17 & 10 \end{bmatrix},$$

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}, \quad M_1 E_1 A = \begin{bmatrix} 6 & 18 & -12 \\ 0 & -2 & 2 \\ 0 & 8 & 16 \end{bmatrix}.$$

- $(2) \leftrightarrow (3)$, $p[2] = 3$:

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/4 & 1 \end{bmatrix}, M_2 E_2 M_1 E_1 A = \begin{bmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{bmatrix}$$

- In general, upon completion we emerge with $M_{n-1}E_{n-1} \cdots M_1 E_1 A = U$, an upper triangular matrix.

- To solve the linear system $Ax = b$, we
  - Compute $y = M_{n-1}E_{n-1} \cdots M_1 E_1 b$;
  - Solve the upper triangular system $Ux = y$.

All the information necessary to do this is contained in the array A and the vector p.

- Cost. $O(n^2)$ comparisons associated with the search for the pivots. The overall algorithm involves $2n^3/3$ flops.

Scaled Partial Pivoting

- Scale the coefficients before deciding on row exchanges;
- Scale factor: for row i, $S_i = \max_{j=1,2,\ldots,n} |a_{i,j}|$;
- Let $k+1 \leq j \leq n$ be such that

$$|a_{j,k}^{(k)}/S_j| = \max_{i=k+1,\ldots,n} |a_{i,k}^{(k)}/S_i|.$$

If $|a_{j,k}^{(k)}/S_j| > |a_{k,k}^{(k)}/S_k|$ then rows j and k are exchanged.

Example. For

$$A = \begin{bmatrix} 2.11 & -4.21 & 0.921 \\ 4.01 & 10.2 & -1.12 \\ 1.09 & 0.987 & 0.832 \end{bmatrix},$$

$$s_1 = 4.21, \quad s_2 = 10.2, \quad s_3 = 1.09$$

$$\frac{|a_{1,1}|}{s_1} = 0.501, \quad \frac{|a_{2,1}|}{s_2} = 0.393, \quad \frac{|a_{3,1}|}{s_3} = 4.21.$$

- $(1) \leftrightarrow (3)$

$$\begin{bmatrix} 1.09 & 0.987 & 0.832 \\ 4.01 & 10.2 & -1.12 \\ 2.11 & -4.21 & 0.921 \end{bmatrix}.$$

Compute the multipliers $m_{2,1}$, $m_{3,1}$, ...

- Cost. Additional $O(n^2)$ comparisons, and $O(n^2)$ flops (divisions).

Complete Pivoting

- search *all* the entries $a_{i,j}$, $k \le i, j \le n$, to find the entry with the largest magnitude. Both row and column interchanges are performed to bring this entry into the pivot position.

- only recommended for systems where accuracy is essential since it requires an additional $O(n^3)$ comparisons.

An Application: Multiple Right Hand Side

Suppose A is nonsingular and n-by-n, and that B is n-by-p. Consider the problem of finding X (n-by-p) so that $AX = B$. If $X = [x_1, \ldots, x_p]$ and $B = [b_1, \ldots, b_p]$ are column partitions, then

```
Compute PA = LU;
for k from 1 to p do
    Solve Ly = Pb_k;
    Solve Ux_k = y;
od;
```

Note that A is factored just once. If $B = I_n$, then we emerge with a computed A^{-1}.

Strictly Diagonally Dominant Matrices

Definition. An $n \times n$ matrix A is strictly diagonally dominant (sdd) if $\left(|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \right)$ holds for each $i = 1, 2, \ldots, n$.

Theorem. An sdd matrix is nonsingular.

Proof. Suppose there is $x \neq 0$ such that $Ax = 0$. Then there is $1 \leq k \leq n$ such that $|x_k| = \max_{1 \leq j \leq n} |x_j| > 0$. Since x is a solution to $Ax = b$, $a_{k,k}x_k + \sum_{j=1, j\neq k}^{n} a_{k,j}x_j = 0$. Hence,

$$|a_{k,k}| \leq \sum_{j=1, j\neq k}^{n} |a_{k,j}||x_j|/|x_k| \leq \sum_{j=1, j\neq n}^{n} |a_{k,j}|.$$

A contradiction since A is sdd. Hence, $x = 0$ is the only solution to $Ax = b$. Equivalently, A is nonsingular.

Theorem. Let A be an sdd matrix. Then Gaussian elimination can be performed on any linear system of the form $Ax = b$ to obtain its unique solution without row or column interchanges, and the computations are stable to the growth of roundoff errors.

Proof sketch.

$$\begin{array}{cccccc} A^{(1)} & \xrightarrow{\text{type } 3} & A^{(2)} & \dots & \xrightarrow{\text{type } 3} & A^{(n)} \\ \big\uparrow & & \big\downarrow & & & \big\downarrow \\ \text{sdd} & & \text{sdd} & & & \text{sdd} \end{array}$$

LDM^T and LDL^T Factorizations

Theorem. If all the leading principal submatrices of $A \in \mathbb{R}^{n\times n}$ are nonsingular, then there exist unique lower triangular matrices L and M and a unique diagonal matrix $D = \text{diag}(d_1, \ldots, d_n)$ such that $A = LDM^T$.

Proof. By the theorem on p.19, $A = LU$ exists. Set $D = \text{diag}(d_1, \ldots, d_n)$ with $d_i = u_{i,i}$ for $1 \le i \le n$. Since D is nonsingular and $M^T = D^{-1}U$ is *unit* upper triangular, $A = LU = LD(D^{-1}U) = LDM^T$. Uniqueness follows from the uniqueness of the LU factorization.

Theorem. If $A = LDM^T$ is the LDM^T factorization of a nonsingular *symmetric* matrix A, then $L = M$.

Example. For the matrix

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 4 & 5 & 8 \\ 7 & 8 & 10 \end{bmatrix},$$

The LU factorization of A is

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & \frac{20}{11} & 1 \end{bmatrix}, \; U = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -11 & -20 \\ 0 & 0 & -\frac{29}{11} \end{bmatrix}.$$

Set

$$D = \mathrm{diag}(U) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -11 & 0 \\ 0 & 0 & -\frac{29}{11} \end{bmatrix}.$$

Then the matrix M^T in the LDM^T factorization of A is

$$\mathrm{M}^T = \begin{bmatrix} 1 & 4 & 7 \\ 0 & 1 & \frac{20}{11} \\ 0 & 0 & 1 \end{bmatrix}.$$

Since A is symmetric, $M^T = L^T$ and we have the LDL^T factorization of A.

Remark. It is possible to compute the matrices L, D, and M directly, instead of computing them via the LU factorization.

Positive Definite Matrices

Definition. A matrix A is positive definite if $x^T A x > 0$ for every nonzero n-dimensional column vector x.

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

a. A is nonsingular;

b. $a_{i,i} > 0$, for each $1 \leq i \leq n$;

c. $\max_{1 \leq k,j \leq n} |a_{k,j}| \leq \max_{1 \leq i \leq n} |a_{i,i}|$;

d. $(a_{i,j})^2 < a_{i,i} a_{j,j}$, for each $i \neq j$.

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

Example. For

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

- $\det A_1 = \det \begin{bmatrix} 2 \end{bmatrix} = 2 > 0$,
- $\det A_2 = \det \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = 3 > 0$,
- $\det A_3 = \det A = 4 > 0$.

Since A is also symmetric, A is positive definite.

Theorem. A symmetric matrix A is positive definite if and only if Gaussian elimination without row exchanges can be performed on the linear system $Ax = b$ with all the pivot elements positive. Moreover, in this case, the computations are stable with respect to the growth of roundoff errors.

Theorem. (**Cholesky Factorization**) If $A \in \mathbb{R}^{n \times n}$ is positive definite, then there exists a unique lower triangular $G \in \mathbb{R}^{n \times n}$ with positive diagonal entries such that $A = GG^T$.

Proof. By the second theorem of p.33, there exists a unit lower triangular L and a diagonal $D = \text{diag}(d_1, \ldots, d_n)$ such that $A = LDL^T$. Since the d_k are positive, the matrix $G = L\text{diag}(\sqrt{d}_1, \ldots, \sqrt{d}_n)$ is real lower triangular with positive diagonal entries. It also satisfies $A = GG^T$. Uniqueness follows from the uniqueness of LDL^T factorization.

Example. For the positive definite matrix A on p.37, the matrices L and D in the LDL^T factorization for A are

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & -2/3 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3/2 & 0 \\ 0 & 0 & 4/3 \end{bmatrix}.$$

Hence, the matrix $G = L\text{diag}(\sqrt{d_1}, \sqrt{d_2}, \sqrt{d_3})$ in the GG^T factorization of A is

$$G = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ -1/2\sqrt{2} & 1/2\sqrt{6} & 0 \\ 0 & -1/3\sqrt{6} & 2/3\sqrt{3} \end{bmatrix}.$$

Tridiagonal Matrices

- Matrices of the form

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & \dots & \dots & \dots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & & & & \vdots \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & 0 \\ \vdots & & & & & & a_{n-1,n} \\ 0 & \dots & \dots & \dots & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

Now suppose A can be factored into the triangular matrices L and U. Suppose that the matrices can be found in the form

$$L = \begin{pmatrix} l_{1,1} & 0 & \dots & \dots & 0 \\ l_{2,1} & l_{2,2} & & & \vdots \\ 0 & & & & \vdots \\ \vdots & & & & 0 \\ 0 & \dots & 0 & l_{n,n-1} & l_{n,n} \end{pmatrix}, \; U = \begin{pmatrix} 1 & u_{1,2} & 0 & \dots & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & & & 0 \\ \vdots & & & & u_{n-1,n} \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix}.$$

The zero entries of A are automatically generated by LU.

Multiplying $A = LU$, we also find the following conditions:

$$\begin{aligned}
a_{1,1} &= l_{1,1}; & (2)\\
a_{i,i-1} &= l_{i,i-1}, \quad \text{for each } i = 2, 3, \ldots, n; & (3)\\
a_{i,i} &= l_{i,i-1}u_{i-1,i} + l_{i,i}, \text{ for each } i = 2, 3, \ldots, n; & (4)\\
a_{i,i+1} &= l_{i,i}u_{i,i+1},, \text{ for each } i = 1, 2, \ldots, n-1. & (5)
\end{aligned}$$

This system is straightforward to solve: (2) and (3) give us $l_{1,1}$ and the off-diagonal entries of L, (4) and (5) are used alternately to obtain the remaining entries of L and U.

This solution technique is often referred to as *Crout factorization.*

Cost. $(5n-4)$ multiplications/divisions; $(3n-3)$ additions/subtractions.