Chapter 3 Methods for Solving Linear Systems $\mathrm{Ax} = \mathrm{b}$

Baodong LIU
baodong@sdu.edu.cn

- **the Linear System of Equations(LSEs):**

$$(I)\begin{cases} E_1 : a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ E_2 : a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ E_n : a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

- Direct techniques are methods that gives an answer in a fixed number of steps, and subject only to the roundoff error.

3.1 Linear System of Equations

Three operations or **mathematical transposition** to simplify the linear system:

1. **Multiplied operation–数乘**:
   Equation E_i can be multiplied by any nonzero constant λ

$$(\lambda E_i) \to E_i.$$

2. **Multiplied and added operation–倍加**:
   Equation E_j can be multiplied by nonzero contant, and added to Equation E_i in place of E_i, denoted by

$$(\lambda E_j + E_i) \to E_i.$$

3. **Transposition–交换**:
   Equation E_i and E_j can be transposed in order, denoted by

$$E_i \leftrightarrow E_j.$$

Example: To solve the equations:

$$
\begin{array}{lrcrcrcrcr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2: & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\
E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4,
\end{array}
$$

Step 1 Using the equation E_1 to eliminate the coefficients of x_1 from E_2, E_3, E_4 by performing operations:

$$(E_2 - 2E_1) \to (E_2), (E_3 - 3E_1) \to (E_3), (E_4 + E_1) \to (E_4)$$

and the resulting system is

$$\begin{array}{lrrrrrrrrr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\
E_3: & & - & 4x_2 & - & x_3 & - & 7x_4 & = & -15, \\
E_4: & & & 3x_2 & + & 3x_3 & + & 2x_4 & = & 8,
\end{array}$$

Step 2 Using the equation E_2 to eliminate the coefficients of x_2 from E_3, E_4 by performing operations:

$$(E_3 - 4E_2) \to (E_3), (E_4 + 3E_2) \to (E_4),$$

resulting the system

$$\begin{array}{lrcrcrcrcr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\
E_3: & & & & & 3x_3 & + & 13x_4 & = & 13, \\
E_4: & & & & & & - & 13x_4 & = & -13,
\end{array}$$

Step 3 The system is now transposed in the triangular form and can be solved by backward–substitution process.

- By the equation E_4 implies $x_4 = 1$
- E_3 can be solved for x_3 to gives

$$x_3 = \frac{1}{3}(13 - 13x_4) = 0,$$

- E_2 gives

$$x_2 = -(-7 + 5x_4 + x_3) = 2$$

- Continuing to E_1 for x_1 gives

$$x_1 = 4 - 3x_3 - x_2 = -1$$

Notation:

Let $\mathbf{A}$ be an $n \times m (n$ by $m)$ matrix, and

$$\mathbf{A} = (a_{ij})_{n\times m} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

and a_{ij} refers to the entry at the intersection of the ith row and jth column.

and

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

be n-dimensional **column vector**, and

$$\mathbf{y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_m \end{pmatrix}$$

be m-dimensional **row vector**.

Now, the linear system of equations (LSEs) can be rewritten as in matrix form:

$$\mathbf{Ax} = \mathbf{b}$$

where, $\mathbf{A}$ is the coefficient $n \times n$ matrix, $\mathbf{x}$ and $\mathbf{b}$ are n-dimensional column vectors.

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

To solve LSEs, we construct the **Augmented Matrix–增广矩阵**

$$\tilde{A} = [A, \mathbf{b}] = \left(\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right)$$

Gaussian Elimination with Backward Substitution Method

Step 1. Assume that $a_{11} \neq 0$, perform the operation

$$(E_j - (a_{j1}/a_{11})E_1) \to E_j$$

to eliminate the coefficients $a_{21}, a_{31}, \cdots, a_{n1}$ of x_1 for each $j = 2, 3, \cdots, n$.

$$\begin{array}{c} \downarrow \\ \to \\ \to \\ \to \end{array} \left(\begin{array}{ccccc:c} \mathbf{a_{11}} & a_{12} & \cdots & a_{1n} & \vdots & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & \vdots & b_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & \vdots & b_n \end{array} \right)$$

备注： 该步算法共需要计算$(n+1)(n-1)$ 次乘法（或除法）

The resulting matrix has the form.

$$\tilde{A}^{(1)} = [A^{(1)}, \mathbf{b}^{(1)}] = \left(\begin{array}{ccccc|c} a_{11} & a_{12} & \cdots & a_{1n} & \vdots & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & \vdots & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & \vdots & b_n^{(1)} \end{array} \right)$$

where for each $i = 2, 3, \cdots, n$,

$$\begin{aligned} a_{i1}^{(1)} &= 0; \\ a_{ij}^{(1)} &= a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}, j = 2, 3, \cdots, n; \\ b_i^{(1)} &= b_i - \frac{a_{i1}}{a_{11}} b_1; \end{aligned}$$

Step 2. For $\tilde{A}^{(1)}$, suppose that $a_{22}^{(1)} \neq 0$, do operations

$$(E_i - (a_{i2}^{(1)}/a_{22}^{(1)})E_2) \to E_i, i = 3, 4, \cdots, n$$

to eliminate the coefficients

$$a_{32}^{(1)}, a_{42}^{(1)}, \cdots, a_{n2}^{(1)}.$$

Thus we obtain

$$\tilde{A}^{(2)} = [A^{(2)}, \mathbf{b}^{(2)}] = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \vdots & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & \vdots & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & \vdots & b_3^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{nn}^{(2)} & \vdots & b_n^{(2)} \end{pmatrix}$$

where for each $i = 3, \cdots, n$,

$$
\begin{aligned}
a_{i2}^{(2)} &= 0; \\
a_{ij}^{(2)} &= a_{ij}^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} a_{2j}^{(1)}, j = 3, \cdots, n; \\
b_i^{(2)} &= b_i^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} b_2^{(1)};
\end{aligned}
$$

备注： 该步算法共需要计算$n(n-2)$ 次乘法（或除法）

Step $k-1$: suppose that we have done $k-1$ steps, and get that

$$\tilde{A}^{(k-1)} = \left(\begin{array}{ccccccc|c}
a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} & b_1 \\
 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & a_{2,k+1} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\
 & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\
 & & & a_{k+1,k}^{(k-1)} & a_{k+1,k+1}^{(k-1)} & \cdots & a_{k+1,n}^{(k-1)} & b_{k+1}^{(k-1)} \\
 & & & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{nk}^{(k-1)} & a_{n,k+1}^{k-1} & \cdots & a_{nn}^{(k-1)} & b_n^{(k-1)}
\end{array}\right)$$

Step k If $a_{kk}^{(k-1)} \neq 0$, then do the kth step

$$(E_i - (a_{ik}^{(k-1)} / a_{kk}^{(k-1)}) E_k) \to E_i, i = k+1, k+2, \cdots, n$$

to eliminate the coefficients

$$a_{k+1,k}^{(k-1)}, a_{k+2,k}^{(k-1)}, \cdots, a_{n,k}^{(k-1)},$$

and obtain the new matrix form

$$
\begin{aligned}
\tilde{A}^{(k)} &= [A^{(k)}, \mathbf{b}^{(k)}] \\
&= \left(\begin{array}{ccccccc:c}
a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} & b_1 \\
 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & a_{2,k+1} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\
 & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\
 & & & 0 & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\
 & & & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & 0 & a_{n,k+1}^{k} & \cdots & a_{nn}^{(k)} & b_n^{(k)}
\end{array}\right)
\end{aligned}
$$

备注： 该步算法共需要计算$(k+1)(k-1)$ 次乘法（或除法）

where for each $i = k+1, \cdots, n$,

$$
\begin{aligned}
a_{ik}^{(2)} &= 0; \\
a_{ij}^{(k)} &= a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}, j = k+1, \cdots, n; \\
b_{i}^{(2)} &= b_{i}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_{k}^{(k-1)};
\end{aligned}
$$

Step $n-1$. After $(n-1)$th elimination process, we obtain

$$\tilde{A}^{(n-1)} = [A^{(n-1)}, \mathbf{b}^{(n-1)}]$$

$$= \left(\begin{array}{ccccccc|c}
a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} & b_1 \\
 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & a_{2,k+1} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\
 & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\
 & & & & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\
 & & & & & \ddots & \vdots & \vdots \\
 & & & & & & a_{nn}^{(n-1)} & b_n^{(n-1)}
\end{array}\right)$$

备注：

1. 该步算法共需要计算3×1 次乘法（或除法）
2. 累计计算复杂度为

$$1 \times 3 + 2 \times 4 + \cdots + (n-1)(n+1) = \frac{n^3 - 3n + 2}{3} = O(n^3)$$

- Thus the corresponding linear system of equations is transposed to the new linear system of equations with upper-triangular form.

$$(II)\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \ddots \quad \vdots \\ a_{nn}x_n = b_n \end{cases}$$

- Thus the **backward substitution** can be performed for solving the new LSEs.

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \\ x_j &= \frac{b_j - \sum_{k=j+1}^{n} a_{jk}x_k}{a_{jj}}, j = n-1, n-2, \cdots, 1 \end{aligned}$$

Notes on Gaussian Elimination Method:

- The Gaussian Elimination Procedure will fail if one of the elements

$$a_{11}, a_{22}^{(1)}, \cdots, a_{kk}^{(k-1)}, \cdots, a_{nn}^{(n-1)}$$

is zero, because in this case, the step

$$\left(E_i - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} E_k \right) \rightarrow (E_i)$$

either cannot be performed, or the backward substitution cannot be accomplished.

- This does not mean that the linear system has no solution, but rather that the technique for finding the solution must be altered.

Notes on Gaussian Elimination Method:

- The Gaussian Elimination Procedure will fail if one of the elements

$$a_{11}, a_{22}^{(1)}, \cdots, a_{kk}^{(k-1)}, \cdots, a_{nn}^{(n-1)}$$

is zero, because in this case, the step

$$\left(E_i - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} E_k \right) \rightarrow (E_i)$$

either cannot be performed, or the backward substitution cannot be accomplished.

- This does not mean that the linear system has no solution, but rather that the technique for finding the solution must be altered.

- To continue the Gaussian Elimination Procedure in the case that some $a_{kk}^{(k-1)} = 0$ for $k = 1, 2, \cdots, n-1$.
- The kth column of $\tilde{A}_{kk}^{(k-1)}$ from the k row to the nth row is searched for the first nonzero entry, suppose that $a_{pk}^{(k-1)} \neq 0$ for some p, with $k+1 \leq p \leq n$, then the operation $(E_k) \leftrightarrow (E_p)$ is performed to obtain new equivalent matrix form, and then do next elimination step with new matrix, otherwise if all

$$a_{pk}^{(k-1)} = 0, k+1 \leq p \leq n,$$

then by the solution theory of linear system, the linear system does not have a unique solution.

- To continue the Gaussian Elimination Procedure in the case that some $a_{kk}^{(k-1)} = 0$ for $k = 1, 2, \cdots, n-1$.
- The kth column of $\tilde{A}_{kk}^{(k-1)}$ from the k row to the nth row is searched for the first nonzero entry, suppose that $a_{pk}^{(k-1)} \neq 0$ for some p, with $k+1 \leq p \leq n$, then the operation $(E_k) \leftrightarrow (E_p)$ is performed to obtain new equivalent matrix form, and then do next elimination step with new matrix, otherwise if all

$$a_{pk}^{(k-1)} = 0, k+1 \leq p \leq n,$$

then by the solution theory of linear system, the linear system does not have a unique solution.

ALGORITHM 6.1:

Input: N–dimension, $A(N,N), B(N)$

Output: Solution $x(N)$ or Message that LESs has no unique solution.

Step 1 For $k = 1, 2, \cdots, N-1$, do step 2-4.

Step 2 Set p be the smallest integer with $k \leq p \leq N$ and $Ap, k \neq 0$. If no p can be found, output:"no unique solution exists"; stop.

Step 3 If $p \neq k$, do transposition $E_p \leftrightarrow E_k$.

Step 4 For $i = k+1, \cdots, N$

- 1. Set $m_{i,k} = A(i,k)/A(k,k)$
- 2. Set $B(i) = B(i) - m_{i,k}B(k)$
- 3. For $j = k+1, \cdots, N$, set $A(i,j) = A(i,j) - m_{i,k}A(k,j)$;

ALGORITHM 6.1: Continued

Step 5 If $A(N,N) \neq 0$,set $x(N) = B(N)/A(N,N)$;
Else, output:"no unique solution exists."

Step 6 For $i = N-1, N-2, \cdots, 1$. SET

$$X(i) = \left[B(i) - \sum_{j=i+1}^{N} A(i,j)x(j)\right]/A(i,i)$$

Step 7 Output the solution $x(N)$

3.2 Pivoting Strategies (主元消去法)

- When $a_{kk}^{(k-1)} = 0$ in Gaussian Elimination method, we always need to make a row interchange.
- To reduce roundoff error, it is necessary to perform row interchange even when $a_{kk}^{(k-1)} \neq 0$.
- If $a_{kk}^{(k-1)} \neq 0$. is much small in magnitude compared to $a_{ik}^{(k-1)}, i = k+1, \cdots, n$, the multiplier

$$m_{jk} = \frac{a_{jk}^{(k-1)}}{a_{kk}^{(k-1)}}$$

or when performing the backward substitution for

$$x_k = \frac{b_k^{(k-1)} - \sum_{j=k+1}^{n} a_{kj}^{(k-1)} x_j}{a_{kk}^{(k-1)}}$$

will have magnitude much larger than 1.

- There may occur larger roundoff error because of the division by $a_{kk}^{(k-1)}$ in elimination and substitution operations.
- To avoid this, an algorithm of Pivoting technique is introduced.

3.2.1 Gaussian Elimination with Maximal Column (partial) Pivoting Technique—列主元消去法:

Algorithm description:

- For $k = 1, 2, \cdots, n-1$, Choose the smallest integer $p, p \geq k$, such that

$$|a_{pk}^{(k-1)}| = \max_{k \leq i \leq n} |a_{ik}^{(k-1)}|.$$

- If $p \neq k$, make interchange between the rows E_p and E_k: $(E_p) \leftrightarrow (E_k)$, then do Gaussian elimination to make the elements

$$a_{k+1,k}^{(k-1)}, \cdots, a_{n,k}^{(k-1)}$$

to be zeros.

3.2.2 Gaussian Elimination with Maximal Row Pivoting Technique—行主元消去法

Algorithm description:

- For $k = 1, 2, \cdots, n-1$, Choose the smallest integer

$$|a_{kp}^{(k-1)}| = \max_{k \le j \le n} |a_{kj}^{(k-1)}|.$$

- If $p \neq k$, make interchange between pth and kth columns, and $x_k \leftrightarrow x_p$ then do Gaussian elimination to make the elements $a_{k+1,k}^{(k-1)}, \cdots, a_{n,k}^{(k-1)}$ to be zeros.

3.2.3 Gaussian Elimination with Pivoting Technique–全主元消去法

Algorithm description:

- For $k = 1, 2, \cdots, n-1$, Choose the maximal element

$$|a_{pq}^{(k-1)}| = \max_{k \le i,j \le n} |a_{ij}^{(k-1)}|.$$

- If $p \ge k$ and $q \ge k$, make interchanges between the q and kth columns, and then $E_p \leftrightarrow E_k$, do Gaussian elimination to make the elements $a_{k+1,k}^{(k-1)}, \cdots, a_{n,k}^{(k-1)}$ to be zero.

3.2.4 Gaussian Elimination with scaled Partial Pivoting Technique—按比例列主元消去法

- First, define the scaled factors s_i for each row vector $i = 1, 2, \cdots, n$,

$$s_i = \max_{1 \le j \le n} |a_{ij}|.$$

- If for some i we have $s_i = 0$. Then the system has no unique solution since all elements in the ith row are zero.
- If no this kind of cases, then choose the integer p, such that

$$\frac{a_{p1}}{s_p} = \max_{1 \le i \le n} \frac{a_{i1}}{s_i}.$$

- Performing the interchange $E_p \leftrightarrow E_1$,then applying Gaussian Elimination Technique to eliminate the elements to be zero in the first column after a_{11}
- In a similar manner, at the step k, find the small integer $r, r \geq k$, such that

$$\frac{a_{rk}}{s_r} = \max_{k \leq i \leq n} \frac{a_{ik}}{s_i}.$$

  then do interchange $E_r \leftrightarrow E_k$, and applying Gaussian Elimination Technique to eliminate the elements to be zero in the kth column after a_{kk}

Example:

- Solve the linear system using three-digit rounding arithmetic.

$$\begin{array}{rrcrcrcr} E_1: & 2.11x_1 & - & 4.21x_2 & + & 0.921x_3 & = & 2.01, \\ E_2: & 4.01x_1 & + & 10.2x_2 & - & 1.12x_3 & = & -3.09, \\ E_3: & 1.09x_1 & + & 0.987x_2 & + & 0.832x_3 & = & 4.21. \end{array}$$

- The augmented matrix is in the form

$$[A, b] = \left(\begin{array}{ccc|c} 2.11 & -4.21 & 0.921 & 2.01 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 1.09 & 0.987 & 0.832 & 4.21 \end{array} \right)$$

Partial Pivoting method:

Step 1 $\max_{1\leq i\leq 3}|a_{i1}| = a_{21} = 4.01$, making row interchange $(E_2) \leftrightarrow (E_1)$, get new matrix form

$$A_1 = \left(\begin{array}{ccc:c} 4.01 & 10.2 & -1.12 & -3.09 \\ 2.11 & -4.21 & 0.921 & 2.01 \\ 1.09 & 0.987 & 0.832 & 4.21 \end{array}\right)$$

Performing Gaussian Elimination, gives

$$A_2 = \left(\begin{array}{ccc:c} 4.01 & 10.2 & -1.12 & -3.09 \\ 0 & -9.58 & 1.51 & 3.64 \\ 0 & -1.79 & 1.14 & 5.05 \end{array}\right)$$

Step 2 Comparing $a_{22}, a_{3,2}$, get the absolute maximum element $|a_{22}| = 9.58$, no row interchange needs to interchange, do Gaussian Elimination to make a_{32} to be zero

$$A_3 = \left(\begin{array}{ccc|c} 4.01 & 10.2 & -1.12 & -3.09 \\ 0 & -9.58 & 1.51 & 3.64 \\ 0 & 0 & 0.858 & 4.37 \end{array} \right)$$

Step 3 Making backward substitution, we get the solution

$$x_3 = 5.09, x_2 = 0.422, x_1 = 0.422.$$

Scaling Partial Pivoting Method:

$$A_1 = [A, b] = \left(\begin{array}{cccc} 2.11 & \textbf{-4.21} & 0.921 & 2.01 \\ 4.01 & \textbf{10.2} & -1.12 & -3.09 \\ \textbf{1.09} & 0.987 & 0.832 & 4.21 \end{array} \right)$$

Step1. Since $s_1 = 4.21, s_2 = 10.2, s_3 = 1.09$, so

$$\frac{|a_{11}|}{s_1} = \frac{2.11}{4.21} = 0.501, \frac{|a_{21}|}{s_2} = \frac{4.01}{10.2} = 0.393,$$

$$\frac{|a_{31}|}{s_3} = \frac{1.09}{1.09} = 1.$$

Making row interchange $(E_3) \leftrightarrow (E_1)$, obtain

$$A_2 = \left(\begin{array}{cccc} 1.09 & 0.987 & 0.832 & 4.21 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 2.11 & -4.21 & 0.921 & 2.01 \end{array} \right)$$

Performing Gaussian Elimination to eliminate a_{21}, a_{31}, obtain

$$A_3 = \left(\begin{array}{ccc:c} 1.09 & 0.987 & 0.832 & 4.21 \\ 0 & \mathbf{6.57} & -4.18 & -18.6 \\ 0 & \mathbf{-6.12} & -0.689 & -6.16 \end{array}\right)$$

Step2. It is easy to see that

$$\frac{|a_{22}|}{s_2} = \frac{6.57}{10.2} = 0.644 < \frac{|a_{32}|}{s_3} = \frac{6.12}{1.09} = 5.61.$$

so Making row interchange $(E_3) \leftrightarrow (E_2)$, obtain

$$A_4 = \left(\begin{array}{ccc:c} 1.09 & 0.987 & 0.832 & 4.21 \\ 0 & \mathbf{-6.12} & -0.689 & -6.16 \\ 0 & \mathbf{6.57} & -4.18 & -18.6 \end{array}\right)$$

Performing Gaussian Elimination to eliminate a_{32}, obtain

$$A_5 = \left(\begin{array}{ccc|c} 1.09 & 0.987 & 0.832 & 4.21 \\ 0 & -6.12 & -0.689 & -6.16 \\ 0 & 0 & -4.92 & -25.2 \end{array} \right)$$

Step 3 Making backward substitution, we get the solution

$$x_3 = 5.12, x_2 = 0.430, x_1 = 0.431.$$

3.5 Matrix Factorization

To solve the LSEs: $\mathbf{Ax} = \mathbf{b}$, if $\mathbf{A}$ has been factored into the triangular form $\mathbf{A} = \mathbf{LU}$, with the form

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

- Then the system can be written as $\mathbf{LUx} = \mathbf{b}$, and we can solve for $\mathbf{x}$ more easily by using a **two-step process**:
  1. First we let
  $$\mathbf{y} = \mathbf{Ux}$$
  $$\mathbf{Ly} = \mathbf{b}$$
  2. solve the system with forward substitution method
  $$\mathbf{Ly} = \mathbf{b}$$
  for $\mathbf{y}$.
  3. Once $\mathbf{y}$ is known, the same as the backward substitution method can be used to solve linear system
  $$\mathbf{Ux} = \mathbf{y}$$
  to determine $\mathbf{x}$.

$\mathbf{LU}$ factorization

- To examine which matrices have an $\mathbf{LU}$ factorization and find how it is determined, first suppose that Gaussian elimination can be performed on the system $\mathbf{Ax} = \mathbf{b}$ without **row interchanges**.
- The first step in the Gaussian elimination process consists of performing, for each $j = 2, 3, \cdots, n$, the operations

$$(E_j - m_{j,1}E_1) \rightarrow (E)_j, \texttt{where} \;\; m_{j,1} = \frac{a_{j1}}{a_{11}}$$

The first Gaussian transformation matrix

- It is simultaneously accomplished by multiplying the original matrix $\mathbf{A}$ on the left by the matrix

$$\mathbf{M}^{(1)} = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -m_{21} & 1 & 0 & \ddots & & 0 \\ -m_{31} & 0 & 1 & & & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ -m_{n1} & 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

  where

$$m_{j,1} = \frac{a_{j1}}{a_{11}}, j = 2, 3, \cdots, n$$

- We denote the product of this matrix with $\mathbf{A}^{(1)} \equiv \mathbf{A}$ by $\mathbf{A}^{(2)}$ and with $\mathbf{b}$ by $\mathbf{b}^{(2)}$, so

$$\mathbf{A}^{(2)}\mathbf{x} = \mathbf{M}^{(1)}\mathbf{A}\mathbf{x} = \mathbf{M}^{(1)}\mathbf{b} = \mathbf{b}^{(2)}.$$

the Second Gaussian transformation matrix

- In a similar manner we construct $\mathbf{M}^{(2)}$

$$\mathbf{M}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \ddots & & 0 \\ 0 & -m_{32} & 1 & & & 0 \\ \vdots & \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & -m_{n2} & 0 & \cdots & 0 & 1 \end{pmatrix}$$

where

$$m_{j,2} = \frac{a_{j2}^{(2)}}{a_{22}^{(2)}}, j = 3, 4, \cdots, n$$

- Let

$$\begin{aligned} \mathbf{A}^{(3)}\mathbf{x} &= \mathbf{M}^{(2)}\mathbf{A}^{(2)}\mathbf{x} = \mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A}^{(1)}\mathbf{x} = \mathbf{M}^{(2)}\mathbf{b}^{(2)} \\ &= \mathbf{b}^{(3)} = \mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{b}^{(1)}. \end{aligned}$$

In general, with $\mathbf{A}^{(k)}\mathbf{x} = \mathbf{b}^{(k)}$ already formed, multiply by the kth Gaussian transformation matrix

$$\mathbf{M}^{(k)} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \ddots & & 0 \\ 0 & \ddots & 1 & & & 0 \\ \vdots & & -m_{k+1,k} & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{n,k} & \cdots & 0 & 1 \end{pmatrix}$$

where

$$m_{j,k} = \frac{a_{jk}^{(k)}}{a_{kk}^{(k)}}, j = k+1, k+2, \cdots, n$$

to obtain

$$\begin{aligned} \mathbf{A}^{(k+1)}\mathbf{x} &= \mathbf{M}^{(k)} A^{(k)}\mathbf{x} = \mathbf{M}^{(k)} \cdots \mathbf{M}^{(1)}\mathbf{A}\mathbf{x} \\ &= \mathbf{M}^{(k)}\mathbf{b}^{(k)} = \mathbf{b}^{(k+1)} = \mathbf{M}^{(k)} \cdots \mathbf{M}^{(1)}\mathbf{b}. \end{aligned}$$

- The process ends with the formation of $\mathbf{A}^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$, where $\mathbf{A}^{(n)}$ is the upper triangular matrix

$$\mathbf{A}^{(n)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & a_{2,k+1}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots & \vdots & \vdots & \vdots \\ & & & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} \\ & & & & & \ddots & \vdots \\ & & & & & & a_{nn}^{(n)} \end{pmatrix}$$

- given by

$$\mathbf{A}^{(n)} = \mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)} \cdots \mathbf{M}^{(1)}\mathbf{A}.$$

- Let $\mathbf{U} = \mathbf{A}^{(n)} = \mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$
- Since $\mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$ are non-singular matrices, then $\mathbf{U}$ is an upper-triangular matrix.
- This means that

$$\mathbf{U} = \mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)} \cdots \mathbf{M}^{(1)}\mathbf{A}.$$

or

$$[\mathbf{M}^{(1)}]^{-1} \cdots [\mathbf{M}^{(n-2)}]^{-1}[\mathbf{M}^{(n-1)}]^{-1}\mathbf{U} = \mathbf{A}.$$

- Let $\mathbf{U} = \mathbf{A}^{(n)} = \mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$
- Since $\mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$ are non-singular matrices, then $\mathbf{U}$ is an upper-triangular matrix.
- This means that

$$\mathbf{U} = \mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)} \cdots \mathbf{M}^{(1)}\mathbf{A}.$$

or

$$[\mathbf{M}^{(1)}]^{-1} \cdots [\mathbf{M}^{(n-2)}]^{-1}[\mathbf{M}^{(n-1)}]^{-1}\mathbf{U} = \mathbf{A}.$$

- Let $\mathbf{U} = \mathbf{A}^{(n)} = \mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$
- Since $\mathbf{M}^{(n-1)}, \mathbf{M}^{(n-2)}, \cdots, \mathbf{M}^{(1)}$ are non-singular matrices, then $\mathbf{U}$ is an upper-triangular matrix.
- This means that

$$\mathbf{U} = \mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)} \cdots \mathbf{M}^{(1)}\mathbf{A}.$$

or

$$[\mathbf{M}^{(1)}]^{-1} \cdots [\mathbf{M}^{(n-2)}]^{-1}[\mathbf{M}^{(n-1)}]^{-1}\mathbf{U} = \mathbf{A}.$$

- Let

$$\mathbf{L}^{(k)} = [\mathbf{M}^{(k)}]^{-1} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \ddots & & 0 \\ 0 & \ddots & 1 & & & 0 \\ \vdots & & m_{k+1,k} & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \ddots & \vdots \\ 0 & \cdots & m_{n,k} & \cdots & 0 & 1 \end{pmatrix}$$

- So The lower-triangular matrix $\mathbf{L}$ in the factorization of $\mathbf{A}$ is the product of the matrices $\mathbf{L}^{(k)}$:

$$\mathbf{L} = \mathbf{L}^{(1)}\mathbf{L}^{(2)}\cdots\mathbf{L}^{(n-1)} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{pmatrix}$$

Theorem 3.18

If Gaussian elimination can be performed on the linear system $\mathbf{Ax} = \mathbf{b}$ **without row interchanges**, then the matrix $\mathbf{A}$ can be factored into the product of a lower-triangular $\mathbf{L}$ and an upper-triangular matrix $\mathbf{U}$,

$$\mathbf{A} = \mathbf{LU}$$

where

$$\mathbf{U} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & \ddots & \vdots \\ & & & & a_{nn}^{(n)} \end{pmatrix}, \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{pmatrix}$$

Some Remarks on LU Factorization:— 直接求解如何设计?

$$\mathbf{LU} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} = \mathbf{A}.$$

Algorithm: Direct LU Factorization

To factor the $n \times n$ matrix $\mathbf{A} = (a_{ij})$ into the product of the lower-triangular matrix $\mathbf{L} = (l_{ij})$ and the upper-triangular matrix $\mathbf{U} = (u_{ij})$:

INPUT:

- dimension n; the entries $a_{ij}, 1 \leq i, j \leq n$ of $\mathbf{A}$;
- the diagonal $l_{11} = l_{22} = \cdots = l_{nn} = 1$ of $\mathbf{L}$
- the diagonal $u_{11} = \cdots = u_{nn} = 1$ of $\mathbf{U}$

OUTPUT: the entries

$$l_{ij}, 1 \leq j \leq i, 1 \leq i \leq n \mathtt{of} \mathbf{L}$$

and the entries

$$u_{ij}, i \leq j \leq n, 1 \leq i \leq n \mathtt{of} \mathbf{U}.$$

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 $j = 2, \cdots, n$, set

$$\begin{aligned} u_{1j} &= a_{1j}/l_{11}(\texttt{First row of } \mathbf{U}.); \\ l_{j1} &= a_{j1}/u_{11}.(\texttt{First column of } \mathbf{L}) \end{aligned}$$

Step 3 For $i = 2, \cdots, n-1$ 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 $j = i+1, \cdots, n$, set

$$u_{ij} = \frac{1}{l_{ii}}[a_{ij} - \sum_{k=1}^{i-1} l_{ik}\, u_{kj}], (i\texttt{th row of } \mathbf{U})$$

$$l_{ji} = \frac{1}{u_{ii}}[a_{ji} - \sum_{k=1}^{i-1} l_{jk}\, u_{ki}], (i\texttt{th column of } \mathbf{L})$$

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 $\mathbf{A} = \mathbf{LU}$ but $\mathbf{A}$ is singular.)

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

3.6 Special Types of Matrices

In this section, we will discuss **two special matrices**, which **Gaussian Elimination** can be performed **without row interchanges**.

Definition 3.18

The $n \times n$ matrix $\mathbf{A}$ is said to be **strictly diagonally dominant**(严格对角占优) when

$$|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}|$$

holds for each $i = 1, 2, 3, \cdots, n$.

Example 1: Consider two matrices

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

- It can be seen that $\mathbf{A}$ is a **nonsymmetric matrix** and **strictly diagonally dominant**. But $\mathbf{A}^T$ is not strictly diagonally dominant
- The **symmetric matrix B** is not strictly diagonally dominant, nor, of course, $\mathbf{B}^T = \mathbf{B}$ is also not.

Theorem 3.19

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

Definition 3.20

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

- **Note that:** not all authors require symmetry of a positive definite matrix.

Theorem 3.21

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

1. $\mathbf{A}$ is nonsingular
2. $a_{ii} > 0$ for each $i = 1, 2, \cdots, n$.
3. $\max\limits_{1 \le k,j \le n} |a_{k,j}| \le \max\limits_{1 \le i \le n} |a_{ii}|$
4. $(a_{ij})^2 < a_{ii}\, a_{jj}$ for each $i \neq j$

Definition 3.22

A leading principal submatrix of a matrix $\mathbf{A}$ is a matrix of the form

$$\mathbf{A}_k = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}$$

Theorem 3.23

A symmetric matrix $\mathbf{A}$ is positive definite if and only if each of its leading principal submatrices has a positive determinant.

Theorem 3.24

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

- Some interesting facts that are uncovered in constructing the proof of Theorem 3.24 are presented in the following corollaries.

Corollary 3.25

The matrix $\mathbf{A}$ is **positive definite** if and only if $\mathbf{A}$ can be factored in the form $\mathbf{LDL}^T$, where

- $\mathbf{L}$ is lower triangular with 1s on its diagonal
- $\mathbf{D}$ is a diagonal matrix with positive diagonal entries.

Corollary 3.26

The matrix $\mathbf{A}$ is **positive definite** if and only if $\mathbf{A}$ can be factored in the form $\mathbf{LL}^T$, where $\mathbf{L}$ is lower triangular with nonzero diagonal entries.

LL^T Factorization—Choleski Algorithm

For a $n \times n$ symmetric and positive definite matrix $\mathbf{A}$ with the form

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{pmatrix}$$

where $\mathbf{A}^T = \mathbf{A}$.

- To factor the positive definite matrix $\mathbf{A}$ into the form $\mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix with form as follows

$$\mathbf{L} = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix},$$

- The problem is to determine the elements $l_{ij}, j = 1, 2, \cdots, i$, for each $i = 1, 2, \cdots, n$.

- Let us first view the relationship by the definition of equal matrices

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{pmatrix} = \mathbf{L}\mathbf{L}^T$$

$$= \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & {}_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & \cdots & l_{n1} \\ 0 & l_{22} & l_{32} & \cdots & l_{n2} \\ 0 & 0 & l_{33} & \cdots & l_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{pmatrix}$$

- First, determining l_{11} by

$$l_{11} = \sqrt{a_{11}}$$

and $l_{i1}, i = 2, 3, \cdots, n$, by

$$l_{i1} l_{11} = a_{i1}, \texttt{thus} \quad l_{i1} = \frac{a_{1i}}{l_{11}}.$$

- Second to determine l_{22}, by

$$l_{21}^2 + l_{22}^2 = a_{22}$$

thus

$$l_{22} = [a_{22} - l_{21}^2]^{1/2}.$$

and then the second column $l_{i2}, i = 3, 4, \cdots, n$ of matrix L satisfies

$$l_{21} l_{i1} + l_{22} l_{i2} = a_{2i}, i = 3, 4, \cdots, n.$$

so we obtain

$$l_{i2} = \frac{a_{2i} - l_{21} l_{i1}}{l_{22}}, i = 3, 4, \cdots, n$$

- With similar idea, we can drive for any $k, k = 2, 3, \cdots, n-1$

$$l_{kk} = [a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2]^{1/2},$$

and for $i = k+1, k+2, \cdots, n$,

$$l_{ik} = \frac{a_{ki} - \sum_{j=1}^{k-1} l_{kj}\, l_{ij}}{l_{kk}}$$

- Specially, when $k = n$, we have

$$l_{nn} = [a_{nn} - \sum_{j=1}^{n-1} l_{nj}^2]^{1/2}.$$

Choleski's Algorithm

To factor the positive definite $n \times n$ matrix $\mathbf{A}$ into $\mathbf{L}\mathbf{L}^T$, where L is lower triangular:

INPUT the dimension n; entries a_{ij}, for $i = 1, 2, \cdots, n, j = 1, 2, \cdots, i$ of $\mathbf{A}$.

OUTPUT the entries l_{ij}, for $j = 1, 2, \cdots, i$ and $i = 1, 2, \cdots, n$ of $\mathbf{L}$

Step 1 Set $l_{11} = \sqrt{a_{11}}$.

Step2 For $j = 2, \cdots, n$, set $l_{j1} = a_{1j}/l_{11}$.

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

Step 4 Set $l_{ii} = [a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2]^{1/2}$.

Step 5 For $j = i+1, i+2, \cdots, n$, set $l_{ji} = \dfrac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk}}{l_{ii}}$

Step 6 Set $l_{nn} = [a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2]^{1/2}$.

Step 7 OUTPUT l_{ij} for $j = 1, 2, \cdots, i$ and $i = 1, 2, \cdots, n$.
STOP!

$\mathbf{LDL}^T$ Factorization

To factor the positive definite matrix $\mathbf{A}$ into the form $\mathbf{LDL}^T$, where

- $\mathbf{L}$ is a lower triangular matrix with form:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & {}_{n2} & l_{n3} & \cdots & 1 \end{pmatrix},$$

- $\mathbf{D}$ is a diagonal matrix with positive entries on the diagonal, which can be formed as follows

$$\mathbf{D} = \begin{pmatrix} d_{11} & 0 & 0 & \cdots & 0 \\ 0 & d_{22} & 0 & \cdots & 0 \\ 0 & 0 & d_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_{nn} \end{pmatrix}$$

By the relationship, we can see that

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{pmatrix} = \mathbf{LDL}^T$$

$$= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_{11} & 0 & 0 & \cdots & 0 \\ 0 & d_{22} & 0 & \cdots & 0 \\ 0 & 0 & d_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_{nn} \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} & \cdots \\ 0 & 1 & l_{32} & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \cdots \end{pmatrix}$$

$$= \begin{pmatrix} d_{11} & 0 & 0 & \cdots & 0 \\ l_{21}\,d_{11} & d_{22} & 0 & \cdots & 0 \\ l_{31}\,d_{11} & l_{32}\,d_{22} & d_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1}\,d_{11} & l_{n2}\,d_{22} & l_{n3} & \cdots & d_{nn} \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ 0 & 1 & l_{32} & \cdots & l_{n2} \\ 0 & 0 & 1 & \cdots & l_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

- So we can easily get that

$$d_{11} = a_{11}$$

for $j = 2, \cdots, n$,

$$l_{j1} = a_{1j} / d_{11}.$$

- For each i, $i = 2, 3, \cdots, n$

$$d_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 d_{kk}$$

and

$$l_{ji} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk}}{d_{ii}}$$

for each $j = i + 1, i + 2, \cdots, n$.

LDL^T Algorithm

To factor the positive definite $n \times n$ matrix $\mathbf{A}$ into $\mathbf{LDL}^T$, where $\mathbf{L}$ is lower triangular with 1s along the diagonal:

INPUT the dimension n; entries a_{ij}, for $i = 1, 2, \cdots, n, j = 1, 2, \cdots, i$ of $\mathbf{A}$.

OUTPUT the entries l_{ij}, for $j = 1, 2, \cdots, i-1$ and $i = 1, 2, \cdots, n$ of $\mathbf{L}$, and d_i for $1 \leq i \leq n$ of $\mathbf{D}$.

Step 1 Set $d_{11} = a_{11}$.

Step 2 For $j = 2, \cdots, n$, set $l_{j1} = a_{1j}/d_{11}$.

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

Step 4 Set $d_{ii} = a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2 d_{jj}$.

Step 5 For $j = i+1, i+2, \cdots, n$, set

$$l_{ji} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk}}{d_{ii}}$$

Step6 OUTPUT l_{ij} for $1 \leq j \leq i-1, 1 \leq i \leq n$ and $d_{ii}, 1 \leq i \leq n$.STOP!

- To solve the linear system $\mathbf{Ax} = \mathbf{b}$ with the $\mathbf{LDL}^T$ factorization method of $\mathbf{A}$.
- Let
$$\mathbf{Ly} = \mathbf{b}, \mathbf{Dz} = \mathbf{y}, \mathbf{L}^T\mathbf{x} = \mathbf{z}.$$
- Solve the linear system $\mathbf{LY} = \mathbf{b}$ with listed algorithm

  Step 1 Set $y_1 = b_1$.

  Step 2 Set $y_i = b_i - \sum_{j=1}^{i-1} l_{ij}\, y_j,\ i = 2, \cdots, n.$
- Solve $\mathbf{Dz} = \mathbf{y}$, Set
$$z_i = \frac{y_i}{d_{ii}},\ \ i = 1, 2, \cdots, n.$$
- Finally, solve $\mathbf{L}^T\mathbf{x} = \mathbf{z}$.

  Step 4 Set $x_n = z_n$.

  Step 5 Set $x_i = z_i - \sum_{j=i+1}^{n} l_{ji} x_j,\ \ i = n-1, \cdots, 1$

  Step 6 OUTPUT $(x_i, i = 1, \cdots, n)$; STOP.

Using Choleski $\mathbf{LL}^T$ factorization to solve $\mathbf{Ax} = \mathbf{b}$

- This system $\mathbf{Ax} = \mathbf{LL}^T\mathbf{x} = \mathbf{b}$ can be factorized into two subsystem: $\mathbf{Ly} = \mathbf{b}, \mathbf{L}^T\mathbf{x} = \mathbf{y}$.

  Step 1 Set $y_1 = b_1/l_{11}$.

  Step 2 For $i = 2, 3, \cdots, n$, set

$$y_i = (b_i - \sum_{j=1}^{i-1} l_{ij}\, y_j)/l_{ii}.$$

  Step 3 Set $x_n = y_n/l_{nn}$.

  Step 4 For $i = n-1, \cdots, 1$, set

$$x_i = (y_i - \sum_{j=i+1}^{n} l_{ji} x_j)/l_{ii}.$$

  Step 5 OUTPUT $(x_i, i = 1, \cdots, n)$; STOP.

Tri-diagonal Linear System– 三对角矩阵

Definition 6.28

An $n \times n$ matrix $\mathbf{A}$ is called a **band matrix(带状矩阵)**, if integers p and q, with $1 < p, q < n$, exist having the property that $a_{ij} = 0$ whenever $i + p \leq j$ or $j + q \leq i$. The **bandwidth（带宽）** of a band matrix is defined as $w = p + q - 1$.

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1p} & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ a_{q1} & & \ddots & & \ddots & 0 \\ 0 & \ddots & & \ddots & & a_{n-p+1,n} \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,n-q+1} & \cdots & a_{nn} \end{bmatrix}$$

- The matrix of bandwidth 3, occurring when $p = q = 2$, and is called **tridiagonal–三对角矩阵** with the form

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

- The matrix of bandwidth 5, occurring when $p = q = 4$.

$\mathbf{LU}$ Factorization for Tridiagonal Matrix

Suppose that the matrices $\mathbf{L}, \mathbf{U}$ can be found in the form

$$\mathbf{L} = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \ddots & & & \vdots \\ 0 & l_{32} & l_{33} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & l_{n,n-1} & l_{nn} \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 1 & u_{12} & 0 & \cdots & \cdots & 0 \\ 0 & 1 & u_{23} & \ddots & & \vdots \\ \vdots & 0 & 1 & u_{34} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 & u_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

In similar manner, we can gives that the entries in $\mathbf{L}, \mathbf{U}$,

$$\begin{aligned} a_{11} &= l_{11}, \\ a_{i,i-1} &= l_{i,i-1}, i = 2, 3, 4, \cdots, n \\ a_{ii} &= l_{i,i-1} u_{i-1,i} + l_{i,i}, i = 2, 3, \cdots, n \\ a_{i,i+1} &= l_{ii} u_{i,i+1}, i = 2, 3, \cdots, n \end{aligned}$$

Algorithm for Tridiagonal Linear System with Crout Factorization(LU)

INPUT the dimension n, the entries of $\mathbf{A}, \mathbf{b}$

OUTPUT the solution $x_1, x_2, \cdots, x_n$.

Solving $\mathbf{Lz} = \mathbf{b}$

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

STEP 2 For $i = 2, \cdots , , n-1$, set

$$\begin{aligned} l_{i,i-1} &= a_{i,i-1}; \\ l_{ii} &= a_{ii} - l_{i,i-1}u_{i-1,i}; \\ u_{i,i+1} &= a_{i,i+1}/l_{ii} \\ z_i &= (a_{i,n+1} - l_{i,i-1}z_{i-1})/l_{ii}.(\mathbf{Lz} = \mathbf{b}) \end{aligned}$$

STEP 3 set

$$\begin{aligned} l_{n,n-1} &= a_{n,n-1}; \\ l_{nn} &= a_{nn} - l_{n,n-1}u_{n-1,n}; \\ z_n &= (a_{n,n+1} - l_{n,n-1}z_{n-1})/l_{nn}. \end{aligned}$$

Solving $\mathbf{Ux} = \mathbf{z}$

STEP 4 set $x_n = z_n$

STEP 5 For $i = n-1, \cdots, 1$, set

$$x_i = z_i - u_{i,i+1} x_{i+1}.$$

STEP 6 OUTPUT: $(x_1, x_2, \cdots, x_n)$, STOP.

思考问题：

- 算法的复杂性估计：每个算法的计算次数（乘除法或加减法）？
- 大型带状矩阵为稀疏矩阵，如何存储才能有效地减少存储空间？以及如何查询？
- 原始矩阵与计算矩阵如何存储才能减少存储空间？
- 当n 足够大时，除了用矩阵的直接分解方法求解方程组，还有没有好的方法？
- 带状矩阵多出现在偏微分方程数值计算方法中，是常见的矩阵形式.

思考问题：

- 算法的复杂性估计：每个算法的计算次数（乘除法或加减法）？
- 大型带状矩阵为稀疏矩阵，如何存储才能有效地减少存储空间？以及如何查询？
- 原始矩阵与计算矩阵如何存储才能减少存储空间？
- 当n 足够大时，除了用矩阵的直接分解方法求解方程组，还有没有好的方法？
- 带状矩阵多出现在偏微分方程数值计算方法中，是常见的矩阵形式.

思考问题：

- 算法的复杂性估计：每个算法的计算次数（乘除法或加减法）？
- 大型带状矩阵为稀疏矩阵，如何存储才能有效地减少存储空间？以及如何查询？
- 原始矩阵与计算矩阵如何存储才能减少存储空间？
- 当n 足够大时，除了用矩阵的直接分解方法求解方程组，还有没有好的方法？
- 带状矩阵多出现在偏微分方程数值计算方法中，是常见的矩阵形式.

思考问题：

- 算法的复杂性估计：每个算法的计算次数（乘除法或加减法）？
- 大型带状矩阵为稀疏矩阵，如何存储才能有效地减少存储空间？以及如何查询？
- 原始矩阵与计算矩阵如何存储才能减少存储空间？
- 当n 足够大时，除了用矩阵的直接分解方法求解方程组，还有没有好的方法？
- 带状矩阵多出现在偏微分方程数值计算方法中，是常见的矩阵形式.

思考问题：

- 算法的复杂性估计：每个算法的计算次数（乘除法或加减法）？
- 大型带状矩阵为稀疏矩阵，如何存储才能有效地减少存储空间？以及如何查询？
- 原始矩阵与计算矩阵如何存储才能减少存储空间？
- 当n 足够大时，除了用矩阵的直接分解方法求解方程组，还有没有好的方法？
- 带状矩阵多出现在偏微分方程数值计算方法中，是常见的矩阵形式.

3.7 Iterative Methods for Linear Equations

- In this section, we will discuss iterative methods to solve linear systems
- To do this, some problems will occur:
  1. How to construct the iterative scheme?
  2. How to choose the initial vectors?
  3. Does the scheme converge? If so, how to measure the convergent rate?
  4. Conditions which are necessary for LEs.
- Let $\mathbb{R}^n$ denote the set of all n-dimensional column vectors with real-number components.
- To define a distance in $\mathbb{R}^n$, we use the notion of a norm.

Definition 3.27

A **vector norm** on $\mathbb{R}^n$ is a function, $\|\cdot\|$, from $\mathbb{R}^n$ into $\mathbb{R}$ with the following properties:

(i) $\|\mathbf{x}\| \geq 0$ for all $x \in \mathbb{R}^n$,

(ii) $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = (0, 0, \cdots, 0)^T \equiv 0$.

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ for all $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,

(iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Two classes of useful norms: l_2 norm and l_∞ norm

Definition 3.28

The l_2 and l_∞ norm for vectors $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ are defined by

$$\|\mathbf{x}\|_2 = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2} \quad \text{and} \quad \|\mathbf{x}\|_\infty = \max_{1 \le i \le n} |x_i|.$$

In general, the l_2 norm is called **Euclidean Norm**.

To show that the definition of l_∞ norm satisfies the properties of definition 3.27

- It is easy to see that, l_∞ norm follows from the similar results for absolute values, and satisfy the properties (i)-(iii) obviously.
- For property (iv), we can see that, if $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$, then

$$\begin{aligned} \|\mathbf{x}+\mathbf{y}\|_\infty &= \max_{1\le i\le n}\{|x_i+y_i|\} \\ &\le \max_{1\le i\le n}\{|x_i|+|y_i|\} \\ &= \|\mathbf{x}\|_\infty + \|\mathbf{y}\|_\infty. \end{aligned}$$

To show that the definition of l_2 norm satisfies the properties of definition 3.27

- For l_2 norm, the properties of (i)-(iii) of definition 7.1 are satisfied obviously,
- Next we prove it satisfies the property (iv) also
- To prove this, we need a famous inequality.

Theorem 3.29(Cauchy-Buniakowsky-Schwarz Inequality)

For each $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$ in $\mathbb{R}^n$, there has

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

Proof: Suppose $\lambda \in \mathbb{R}$, and

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)^T \neq 0$$

and

$$\mathbf{y} = (y_1, y_2, \cdots, y_n)^T \neq 0$$

in $\mathbb{R}^n$, thus according to the definition of l_2 norm, we obtain

$$\begin{aligned} 0 &\leq \|\mathbf{x} - \lambda \mathbf{y}\|_2^2 \\ &= \sum_{i=1}^{n} (x_i - \lambda y_i)^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2\lambda \sum_{i=1}^{n} x_i y_i + \lambda^2 \sum_{i=1}^{n} y_i^2. \end{aligned}$$

Thus

$$2\lambda \sum_{i=1}^{n} x_i y_i \leq \sum_{i=1}^{n} x_i^2 + \lambda^2 \sum_{i=1}^{n} y_i^2$$

Let

$$\lambda = \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2} \neq 0$$

and substitutes into previous inequality.

$$\begin{aligned} 2\frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2} \sum_{i=1}^{n} x_i y_i &\leq \sum_{i=1}^{n} x_i^2 + \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{y}\|_2^2} \sum_{i=1}^{n} y_i^2 \\ &= \|\mathbf{x}\|_2^2 + \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{y}\|_2^2} \|\mathbf{y}\|_2^2 = 2\|\mathbf{x}\|_2^2 \end{aligned}$$

and divided by λ on each side to produce

$$\sum_{i=1}^{n} x_i y_i \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 = \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \left\{\sum_{i=1}^{n} y_i^2\right\}^{1/2}$$

If $x_i y_i < 0$, replace x_i by $-x_i$, and call the new vector $\tilde{\mathbf{x}} = (\tilde{x}_i) = (-x_i)$. Then $\|\tilde{\mathbf{x}}\|_2 = \|\mathbf{x}\|_2$ and

$$\begin{aligned} \sum_{i=1}^{n} |x_i y_i| &= \sum_{i=1}^{n} \tilde{x} y_i \leq \|\tilde{\mathbf{x}}\|_2 \|\mathbf{y}\|_2 \\ &= \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \left\{\sum_{i=1}^{n} y_i^2\right\}^{1/2}. \blacksquare\blacksquare\blacksquare \end{aligned}$$

For property (iv), using this theorem we have

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

Measurement for the distance between two arbitrary vector:

Definition 3.30

If $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$ are two vectors in $\mathbb{R}^n$, the l_2 and l_∞ distances between $\mathbf{x}$ and $\mathbf{y}$ are defined by

$$\|\mathbf{x} - \mathbf{y}\|_2 = \left\{ \sum_{i=1}^{n} (x_i - y_i)^2 \right\}^{1/2}$$

and

$$\|\mathbf{x} - \mathbf{y}\|_\infty = \max_{1 \le i \le n} |x_i - y_i|.$$

Convergence of a sequence of vectors:

Definition 3.31

A sequence $\{\mathbf{x}^{(k)}\}_0^\infty$ of vectors in $\mathbb{R}^n$ is said to converge to $\mathbf{x}$ with respect to the norm $\|\cdot\|$, if, given any $\epsilon > 0$, there exists an integer $N(\epsilon)$, such that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| \leq \epsilon \quad \text{for all} \quad k \geq N(\epsilon)$$

Convergence for a sequence of vectors:

Theorem 3.32

The sequence $\{\mathbf{x}^{(k)}\}_0^\infty$ of vectors converges to $\mathbf{x}$ in $\mathbb{R}^n$ with respect to the norm $\|\cdot\|_\infty$, if and only if

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \quad i = 1, 2, \cdots, n.$$

Proof:先证必要性

- Suppose that the sequence $\{\mathbf{x}^{(k)}\}_0^\infty$ of vectors converges to $\mathbf{x}$ with respect to the norm $\|\cdot\|_\infty$
- Thus given $\epsilon > 0$, there exists an integer $N(\epsilon)$, such that for all $k \geq N(\epsilon)$,

$$\max_{1 \leq i \leq n} |x_i^{(k)} - x_i| = \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty < \epsilon$$

  holds.
- This inequality implies that

$$|x_i^{(k)} - x_i| < \epsilon \text{ for each } i = 1, 2, \cdots, n.$$

- So $\lim_{k \to \infty} x_i^{(k)} = x_i$, for each $i = 1, 2, \cdots, n$.

再证充分性

- Conversely, suppose that $\lim_{k\to\infty} x_i^{(k)} = x_i$, for each $i = 1, 2, \cdots, n$.
- Thus for a given positive number $\epsilon > 0$, there exists an integer $N_i(\epsilon)$, such that

$$|x_i^{(k)} - x_i| < \epsilon$$

  for i, $i = 1, 2, \cdots, n$.
- Let $N(\epsilon) = \max_{1\le i\le n} N_i(\epsilon)$, then for all $k \ge N(\epsilon)$, $|x_i^{(k)} - x_i| < \epsilon$ hold.
- Thus we have

$$\max_{1\le i\le n} |x_i^{(k)} - x_i| = \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty < \epsilon.$$

- This implies that the sequence $\{\mathbf{x}^{(k)}\}_0^\infty$ of vectors converges to $\mathbf{x}$.■

Theorem 3.33

For each $\mathbf{x} \in \mathbb{R}^n$, we have

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty$$

Proof:

- Let x_j be a coordinate of $\mathbf{x}$ such that

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

- Then

$$\|\mathbf{x}\|_\infty^2 = |x_j|^2 = x_j^2 \leq \sum_{i=1}^n x_i^2 \leq \sum_{i=1}^n x_j^2 = nx_j^2 = n\|\mathbf{x}\|_\infty^2.$$

- Thus

$$\|\mathbf{x}\|_\infty^2 \leq \{\sum_{i=1}^n x_i^2\}^{1/2} = \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty. \blacksquare\blacksquare\blacksquare$$

Note:

- It can be shown that all norms on $\mathbb{R}^n$ are equivalent with respect to convergence;
- That is: if $\|\cdot\|$ and $\|\cdot\|'$ are any two norms on $\mathbb{R}^n$ and $\{\mathbf{x}^k\}_{k=1}^{\infty}$ has the limit $\mathbf{x}$ with respect to $\|\cdot\|$, then $\{\mathbf{x}^k\}_{k=1}^{\infty}$ also has the limit $\mathbf{x}$ with respect to $\|\cdot\|'$.

Definition 3.34

A **matrix norm** on the set of all $n \times n$ matrices is a real-valued function, $\|\cdot\|$, defined on this set, satisfying for all $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ and all real numbers α:

1. $\|\mathbf{A}\| \geq 0$;
2. $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A}$ is $\mathbf{0}$;
3. $\|\alpha\mathbf{A}\| = \alpha\|\mathbf{A}\|$;
4. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$;
5. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$.

- A **distance** between $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ with respect to this matrix norm is $\|\mathbf{A} - \mathbf{B}\|$.

Theorem 3.35

If $\|\cdot\|$ is a vector norm on $\mathbb{R}^n$, then

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|$$

is a matrix norm.

- This is called the **natural**, or **induced, matrix norm** associated with the vector norm.
- In this text, all matrix norms will be assumed to be natural matrix norms unless specified otherwise.
- The following corollary is often used to bound a value of $\|\mathbf{A}\mathbf{x}\|$.

Corollary 3.36

For any vector $\mathbf{x} \neq 0$, matrix $\mathbf{A}$,and any natural norm $\|\cdot\|$, we have

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|.$$

Proof:

- First note that for $\mathbf{x} \neq 0$, the vector $\|\frac{\mathbf{x}}{\|\mathbf{x}\|}\| = 1$. Then we have

$$\left\|\mathbf{A}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)\right\| \leq \|\mathbf{A}\|.$$

- Since $\|\mathbf{x}\|$ is a nonzero real number, which implies that

$$\mathbf{A}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) = \frac{1}{\|\mathbf{x}\|}\mathbf{A}\mathbf{x}$$

- Hence

$$\frac{1}{\|\mathbf{x}\|}\|\mathbf{A}\mathbf{x}\| = \left\|\frac{1}{\|\mathbf{x}\|}A\mathbf{x}\right\| = \left\|\mathbf{A}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)\right\| \leq \|\mathbf{A}\|,$$

which implies that

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\| \blacksquare\blacksquare\blacksquare$$

Proof of Theorem 3.35:

- For $\forall \mathbf{x} \in \mathbb{R}^n$, since $\mathbf{A}$ is $n \times n$ matrix, and $\mathbf{x}$ is n-dimensional column vector, so $\mathbf{Ax} \in \mathbb{R}^n$, thus $\|\mathbf{Ax}\| \geq 0, \forall \mathbf{x} \in \mathbb{R}^n$. Further, $\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\| \geq 0$.
- $\|\mathbf{A}\| = 0 \Leftrightarrow \|\mathbf{Ax}\| = 0, \texttt{for any} \|\mathbf{x}\| = 1 (\neq 0) \Leftrightarrow \mathbf{A} = \mathbf{0}$.
- $\|\alpha\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\alpha\mathbf{Ax}\| = |\alpha| \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\| = |\alpha|\|\mathbf{A}\|$.
-

$$
\begin{aligned}
\|\mathbf{A}+\mathbf{B}\| &= \max_{\|\mathbf{x}\|=1} \|(\mathbf{A}+\mathbf{B})\mathbf{x}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}+\mathbf{Bx}\| \\
&\leq \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\| + \max_{\|\mathbf{x}\|=1} \|\mathbf{Bx}\| = \|\mathbf{A}\| + \|\mathbf{B}\| \\
\|\mathbf{AB}\| &= \max_{\|\mathbf{x}\|=1} \|\mathbf{A}(\mathbf{Bx})\| \leq \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\|\|\mathbf{Bx}\| \\
&\leq \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\|\|\mathbf{B}\|\|\mathbf{x}\| = \|\mathbf{A}\|\|\mathbf{B}\|
\end{aligned}
$$

Most useful matrix norms:

- $\|\mathbf{A}\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty,$
- $\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$

Theorem 3.37

If $\mathbf{A} = (a_{ij})$ is an $n \times n$ matrix, then

$$\|\mathbf{A}\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|.$$

Proof of Theorem 3.37:

- First we show that

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

- Let $\mathbf{x}$ be an n-dimensional column vector with

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

- Since $\mathbf{Ax}$ is also an n-dimensional column vector,

$$\begin{aligned} \|\mathbf{Ax}\|_\infty &= \max_{1\leq i\leq n} |(\mathbf{Ax})_i| = \max_{1\leq i\leq n} |\sum_{j=1}^{n} a_{ij}\, x_j| \\ &\leq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}| \max_{1\leq j\leq n} |x_j| \end{aligned}$$

Continued

- Since $\|\mathbf{x}\|_\infty = 1$, we have

$$\|\mathbf{Ax}\|_\infty \leq \max_{1\leq i\leq n}(\sum_{j=1}^{n}|a_{ij}|)\|\mathbf{x}\|_\infty = \max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|.$$

- Consequently,

$$\|\mathbf{A}\|_\infty = \max_{\|\mathbf{x}\|_\infty=1}\|\mathbf{Ax}\|_\infty \leq \max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|.$$

- Now we need to show the opposite inequality, that

$$\|\mathbf{A}\|_\infty \geq \max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|.$$

- Let p be an integer with

$$\sum_{j=1}^{n} |a_{pj}| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

and $\mathbf{x}$ be the vector with components

$$x_j = \begin{cases} 1, \texttt{if} & a_{pj} \ge 0 \\ -1, \texttt{if} & a_{pj} < 0 \end{cases}$$

- Then $\|\mathbf{x}\|_\infty = 1$ and $a_{pj} x_j = |a_{pj}|$, for all $j = 1, 2, \cdots, n$, so

$$\begin{aligned} \|\mathbf{Ax}\|_\infty &= \max_{1 \le i \le n} |\sum_{j=1}^{n} a_{ij} x_j| \ge |\sum_{j=1}^{n} a_{pj} x_j| \\ &= \left|\sum_{j=1}^{n} |a_{pj}|\right| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \end{aligned}$$

This result implies that

$$\|\mathbf{A}\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|\mathbf{A}\mathbf{x}\|_\infty \geq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|$$

which together with inequality previous, gives

$$\|\mathbf{A}\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.\blacksquare\blacksquare\blacksquare$$

Definition 3.38

If $\mathbf{A}$ is a square matrix, the polynomial defined by

$$p(\mathbf{A}) = \det(\mathbf{A} - \lambda \mathbf{I})$$

is called the **characteristic polynomial** of $\mathbf{A}$.

Notes:

1. $p(\mathbf{A})$ is an nth-degree polynomial;
2. $p(\mathbf{A})$ has at most n distinct zeros, some of which may be complex.
3. If λ is a zero of p, then, since

$$\det(A - \lambda I) = 0,$$

this implies that

$$(\mathbf{A} - \lambda I)\mathbf{x} = \mathbf{0}$$

has nonzero solution.

Definition 3.39

- If p is the **characteristic polynomial** of the matrix $\mathbf{A}$, the zeros of p are called **eigenvalues**, or **characteristic values**, of the matrix $\mathbf{A}$.
- If λ is a eigenvalue of $\mathbf{A}$ and $\mathbf{x} \neq \mathbf{0}$ has the property that

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0},$$

then $\mathbf{x}$ is called an **eigenvector**, or **characteristic vector**, of $\mathbf{A}$ corresponding to the eigenvalue λ

If $\mathbf{x}$ is an eigenvector associated with the eigenvalue $\mathbf{A}$, then $\mathbf{Ax} = \lambda\mathbf{x}$, so the matrix $\mathbf{A}$ takes the vector $\mathbf{x}$ into a scalar multiple of itself.

Definition 3.40

The **spectral radius** $\rho(\mathbf{A})$ of a matrix $\mathbf{A}$ is defined by

$$\rho(\mathbf{A}) = \max|\lambda|,$$

where λ is an eigenvalue of $\mathbf{A}$.

Note that if λ is a complex number and $\lambda = \alpha + \beta i$, then we have $|\lambda| = (\alpha^2 + \beta^2)^{1/2}$.

Theorem 3.41

If $\mathbf{A}$ is an $n \times n$ matrix, then

(i) $\|\mathbf{A}\|_2 = [\rho(\mathbf{A}^T\mathbf{A})]^{1/2}$,

(ii) $\rho(\mathbf{A}) \leq \|\mathbf{A}\|$, for any natural norm $\|\cdot\|$.

Proof of theorem 3.41:

- The proof of part (i) requires more information concerning eigenvalues than We presently have available.
- To prove part (ii), suppose λ is an eigenvalue of $\mathbf{A}$ with eigenvector $\mathbf{x}$ where $\|\mathbf{x}\| = 1$.
- Since $\mathbf{Ax} = \lambda\mathbf{x}$, for any natural norm

$$|\lambda| = |\lambda| \cdot \|\mathbf{x}\| = \|\lambda\mathbf{x}\| = \|\mathbf{Ax}\| \leq \|\mathbf{A}\|\|\mathbf{x}\| = \|\mathbf{A}\|.$$

- Thus,

$$\rho(\mathbf{A}) = \max|\lambda| \leq \|\mathbf{A}\|.\blacksquare\blacksquare\blacksquare$$

Example:

$$\texttt{Let}\quad \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{pmatrix}$$

$$\texttt{then}\quad \mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 2 & -1 \\ 2 & 6 & 4 \\ -1 & 4 & 5 \end{pmatrix}$$

To calculate $\rho(\mathbf{A}^T\mathbf{A})$ we need the eigenvalues of $\mathbf{A}^T\mathbf{A}$. If

$$\begin{aligned} 0 &= \det(\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I}) = \det \begin{pmatrix} 3 & 2 & -1 \\ 2 & 6 & 4 \\ -1 & 4 & 5 \end{pmatrix} \\ &= -\lambda^3 + 14\lambda^2 - 42\lambda = -\lambda(\lambda^2 - 14\lambda + 42) \end{aligned}$$

then $\lambda = 0,\ \texttt{or}\ \ \lambda = 7 \pm \sqrt{7}$, so

$$\|\mathbf{A}\|_2 = \sqrt{\rho(\mathbf{A}^T\mathbf{A})} = \sqrt{\max\{0, 7+\sqrt{7}, 7-\sqrt{7}\}} = \sqrt{7+\sqrt{7}} \approx 3.106.$$

Matrix Convergence

Definition 3.42

We call an $n \times n$ matrix $\mathbf{A}$ convergent if

$$\lim_{k \to \infty} (\mathbf{A}^k)_{ij} = 0,$$

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

Example: Let

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

Computing the power of $\mathbf{A}$, we obtain

$$\mathbf{A}^2 = \begin{pmatrix} \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}, \mathbf{A}^3 = \begin{pmatrix} \frac{1}{8} & 0 \\ \frac{3}{16} & \frac{1}{8} \end{pmatrix}, \mathbf{A}^4 = \begin{pmatrix} \frac{1}{16} & 0 \\ \frac{1}{8} & \frac{1}{16} \end{pmatrix}, \cdots$$

and in general,

$$\mathbf{A}^k = \begin{pmatrix} (\frac{1}{2})^k & 0 \\ \frac{k}{2^{k+1}} & (\frac{1}{2})^k \end{pmatrix}.$$

Since

$$\lim_{k\to\infty} \left(\frac{1}{2}\right)^k = 0, \lim_{k\to\infty} \frac{k}{2^{k+1}} = 0,$$

so $\mathbf{A}$ is a convergent matrix.

Theorem 3.43

The following statements are equivalent.

1. $\mathbf{A}$ is a convergent matrix.
2. $\lim_{n\to\infty} \|\mathbf{A}^n\| = 0$, for some natural norm.
3. $\lim_{n\to\infty} \|\mathbf{A}^n\| = 0$, for all natural norm.
4. $\rho(\mathbf{A}) < 1$.
5. $\lim_{n\to\infty} \mathbf{A}^n\mathbf{x} = \mathbf{0}$, for every $\mathbf{x}$.

Iterative Techniques for Solving Linear Systems

- An iterative technique to solve the linear system

$$\mathbf{Ax} = \mathbf{b}$$

starts with an **initial approximation** $\mathbf{x}^{(0)}$ to the solution $\mathbf{x}$ and generates a sequence of vectors $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that **converges to** $\mathbf{x}$.

- Iterative techniques involve a process that **converts** the system $\mathbf{Ax} = \mathbf{b}$ **into** an equivalent system of the form

$$\mathbf{x} = \mathbf{Tx} + \mathbf{c}$$

for some fixed matrix $\mathbf{T}$ and vector $\mathbf{c}$.

Iterative Techniques for Solving Linear Systems

- An iterative technique to solve the linear system

$$\mathbf{Ax} = \mathbf{b}$$

starts with an **initial approximation** $\mathbf{x}^{(0)}$ to the solution $\mathbf{x}$ and generates a sequence of vectors $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that **converges to** $\mathbf{x}$.

- Iterative techniques involve a process that **converts** the system $\mathbf{Ax} = \mathbf{b}$ **into** an equivalent system of the form

$$\mathbf{x} = \mathbf{Tx} + \mathbf{c}$$

for some fixed matrix $\mathbf{T}$ and vector $\mathbf{c}$.

- After the initial vector $\mathbf{x}^{(0)}$ is selected, the sequence of approximate solution vectors is generated by computing

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}$$

for each $k = 1, 2, 3, \cdots$.

- **Remarks:**
  1. Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques such as the Gaussian elimination method.
  2. For large systems with a high percentage of zero entries, however, these techniques are efficient in terms of both computer storage and computational time.

- After the initial vector $\mathbf{x}^{(0)}$ is selected, the sequence of approximate solution vectors is generated by computing

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}$$

for each $k = 1, 2, 3, \cdots$.

- **Remarks:**
  1. Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques such as the Gaussian elimination method.
  2. For large systems with a high percentage of zero entries, however, these techniques are efficient in terms of both computer storage and computational time.

Iterative Method I: Jacobi Iterative Method

Problem: To solve the linear system of equation in the form

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1i}x_i + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2i}x_i + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i + \cdots + a_{in}x_n = b_i \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{ni}x_i + \cdots + a_{nn}x_n = b_n \end{cases}$$

- If $a_{ii} \neq 0,\ \ i = 1, 2, \cdots, n$, we can rewrite above equation as the follows

$$\begin{cases} x_1 = \frac{1}{a_{11}}[b_1 \qquad\qquad - a_{12}x_2 - \cdots - a_{1i}x_i - \cdots - a_{1n}x_n] \\ x_2 = \frac{1}{a_{22}}[b_2 - a_{21}x_1 \qquad\qquad - a_{23}x_3 - \cdots - a_{2i}x_i - \cdots - a_{2n}x_n] \\ \vdots \\ x_i = \frac{1}{a_{ii}}[b_i - a_{i1}x_1 - \cdots - a_{i,i-1}x_{i-1} \qquad\qquad - a_{i,i+1}x_{i+1} - \cdots - a_{in}x_n] \\ \vdots \\ x_n = \frac{1}{a_{nn}}[b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{ni}x_i - \cdots - a_{n,n-1}x_{n-1} \qquad\qquad] \end{cases}$$

- Jacobi Iterative Method consists of solving the ith equation in $\mathbf{Ax} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

$$x_i = \sum_{j=1, j\neq i}^{n} \left(-\frac{a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \quad \texttt{for} \quad i = 1, 2, \cdots, n$$

Jacobi Iterative Method

- Given $\mathbf{x}^{(0)}$, generating each $x_i^{(k)}$ from components of $\mathbf{x}^{(k-1)}$ for $k \geq 1$ by

$$x_i^{(k)} = \frac{\sum_{j=1, j\neq i}^{n} (-a_{ij} x_j^{(k-1)}) + b_i}{a_{ii}}, i = 1, 2, \cdots, n. \qquad (1)$$

- In details

$$\begin{aligned}
x_1^{(k)} &= \tfrac{1}{a_{11}}[\quad -a_{12}x_2^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1] \\
x_2^{(k)} &= \tfrac{1}{a_{22}}[-a_{21}x_1^{(k-1)} \quad - a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2] \\
&\vdots \\
x_i^{(k)} &= \tfrac{1}{a_{ii}}[-a_{i1}x_1^{(k-1)} - \cdots - a_{i,i-1}x_{i-1}^{(k-1)} \quad - a_{i,i+1}x_{i+1}^{(k-1)} - \cdots - a_{in}x_n^{(k-1)} + b_i] \\
&\vdots \\
x_n^{(k)} &= \tfrac{1}{a_{nn}}[-a_{n1}x_1^{(k-1)} - a_{n2}x_2^{(k-1)} - \cdots - a_{n,n-1}x_{n-1}^{(k-1)} \quad + b_n]
\end{aligned}$$

for $k = 1, 2, \cdots$

- The method is written in the form

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

  by **splitting** $\mathbf{A}$ into its diagonal and off-diagonal parts.

- First $\mathbf{A}$ is split into

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}$$

$$- \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{n-1,n} \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

- **To see this, let**
  $\mathbf{D}$ be the diagonal matrix whose diagonal is the same as $\mathbf{A}$,
  $-\mathbf{L}$ be the strictly lower-triangular part of $\mathbf{A}$,
  $-\mathbf{U}$ be the strictly upper-triangular part of $\mathbf{A}$.
- Then $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$
- The equation $\mathbf{Ax} = \mathbf{b}$, or $(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{b}$, is then transformed into

$$\mathbf{Dx} = (\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b}$$

and, finally,

$$\mathbf{x} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{D}^{-1}\mathbf{b}.$$

- This results in the matrix form of the **Jacobi iterative technique**:

$$\mathbf{x}^{(k)} = \mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\mathbf{x}^{(k-1)} + \mathbf{D}^{-1}\mathbf{b}, \quad k = 1, 2, \cdots \tag{2}$$

- Introducing the notation

$$\mathbf{T} = \mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$$

and $\mathbf{c} = \mathbf{D}^{-1}\mathbf{b}$, the Jacobi technique has the form

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}. \tag{3}$$

- In practice, Eq. (1) is used in computation and Eq. (3) for theoretical purposes.

I Jacobi Iterative Algorithm

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT:
- the number of equations and unknowns n;
- the entries $a_{ij}, 1 \leq i, j \leq n$ of the matrix A;
- the entries $b_i, 1 \leq i \leq n$ of $\mathbf{b}$;
- the entries $XO_i, 1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$;
- tolerance TOL;
- maximum number of iterations N.

OUTPUT: the approximate solution $x_1, x_2, \cdots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3-6.

Step 3 For $i = 1, \leq, n$, set

$$x_i = \frac{-\sum_{j=1, j\neq i}^{n} (a_{ij} XO_j) + b_i}{a_{ii}}$$

Step4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT $(x_1, x_2, \cdots, x_n)$; (Procedure completed successfully.) STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \cdots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded'); (Procedure completed unsuccessfully.) STOP.

Remarks:

1. Step 3 of the algorithm requires that $a_{ii} \neq 0$ for each $i = 1, 2, \cdots, n$. If one of the a_{ii} entries is zero and the system is nonsingular, a reordering of the equations (row interchange) can be performed so that no $a_{ii} = 0$.
2. To speed convergence, the equations should be arranged so that a_{ii} is as large as possible.
3. Another possible stopping criterion in Step 4 is to iterate until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|} < \varepsilon,$$

where $\varepsilon > 0$ is the tolerance. For this purpose, any convenient norm can be used, the usual being the l_∞ norm.

Improvement in Jacobi Iterative Technique —-Gauss-Seidel iterative technique

To compute $x_i^{(k)}$, the components of $\mathbf{x}^{(k-1)}$ are used. Since, for

$$i > 1, x_1^{(k)}, x_2^{(k)}, \cdots, x_{i-1}^{(k)}$$

have already been computed and are likely to be better approximations to the actual solutions

$$x_1, x_2, \cdots, x_{i-1}$$

than

$$x_1^{(k-1)}, x_2^{(k-1)}, \cdots, x_{i-1}^{(k-1)},$$

- It seems more reasonable to compute $x_i^{(k)}$ using these most recently calculated values; that is,

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i}{a_{ii}}, \qquad (4)$$

  for each $i = 1, 2, \cdots, n$, instead of Eq. (1).
- This modification is called the **Gauss-Seidel iterative technique** .
- In details

$$\begin{cases} x_1^{(k)} = \frac{1}{a_{11}}[\quad -a_{12}x_2^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1] \\ x_2^{(k)} = \frac{1}{a_{22}}[-a_{21}x_1^{(k)} \quad - a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2] \\ \vdots \\ x_i^{(k)} = \frac{1}{a_{ii}}[-a_{i1}x_1^{(k)} - \cdots - a_{i,i-1}x_{i-1}^{(k)} \quad - a_{i,i+1}x_{i+1}^{(k-1)} - \cdots - a_{i} \\ \vdots \\ x_n^{(k)} = \frac{1}{a_{nn}}[-a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)} \quad + b_n] \end{cases}$$

Gauss-Seidel iterative technique

- To write the Gauss-Seidel method in matrix form, multiply both sides of Eq. (4) by a_{ii} and collect all kth iterate terms to give

$$\begin{aligned} a_{i1}x_1^{(k)} &+ a_{i2}x_2^{(k)} + \cdots + a_{ii}x_i^{(k)} \\ &= -a_{i,i+1}x_{i+1}^{(k-1)} - a_{i,i+2}x_{i+2}^{(k-1)} - \cdots - a_{i,n}x_n^{(k-1)} + b_i \end{aligned}$$

  for each $i = 1, 2, \cdots, n$.

- Writing all n equations gives

$$\begin{array}{llllll} a_{11}x_1^{(k)} & & & = & -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} \\ a_{21}x_1^{(k)} & +a_{22}x_2^{(k)} & & = & -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} \\ \vdots & & & & \\ a_{n1}x_1^{(k)} & +a_{n2}x_2^{(k)} \cdots & +a_{nn}x_n^{(k)} & = & \end{array}$$

Matrix form

- It follows that the matrix form of the Gauss-Seidel method is

$$(\mathbf{D}-\mathbf{L})\mathbf{x}^{(k)} = \mathbf{U}\mathbf{x}^{(k-1)} + \mathbf{b}$$

or

$$\mathbf{x}^{(k)} = (\mathbf{D}-\mathbf{L})^{-1}\mathbf{U}\mathbf{x}^{(k-1)} + (\mathbf{D}-\mathbf{L})^{-1}\mathbf{b}, k = 1, 2, \cdots \quad (5)$$

- Letting $\mathbf{T}_g = (\mathbf{D}-\mathbf{L})^{-1}\mathbf{U}$ and $\mathbf{c}_g = (\mathbf{D}-\mathbf{L})^{-1}\mathbf{b}$, the Gauss-Seidel technique has the form

$$\mathbf{x}^{(k)} = \mathbf{T}_g\mathbf{x}^{(k-1)} + \mathbf{c}_g.$$

- For the lower-triangular matrix $\mathbf{D}-\mathbf{L}$ to be nonsingular, it is necessary and sufficient that $a_{ii} \neq 0$ for each $i = 1, 2, \cdots, n$.

II Gauss-Seidel Iterative Algorithm

To solve $\mathbf{Ax} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT: the number of equations and unknowns n; the entries $a_{ij}, 1 \leq i, j \leq n$ of the matrix A; the entries $b_i, 1 \leq i \leq n$ of $\mathbf{b}$; the entries $XO_i, 1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT: the approximate solution $x_1, x_2, \cdots, x_n$ or a message that the number of iterations was exceeded.

II Gauss-Seidel Iterative Algorithm

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3-6.

Step 3 For $i = 1, \leq, n$,set

$$x_i = \frac{-\sum_{j=1}^{i-1}(a_{ij}\, x_j) - \sum_{j=i+1}^{n}(a_{ij}\, XO_j) + b_i}{a_{ii}}$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT $(x_1, x_2, \cdots, x_n)$; (Procedure completed successfully.) STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \cdots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded'); (Procedure completed unsuccessfully.) STOP.

Remarks:

1. Step 3 of the algorithm requires that $a_{ii} \neq 0$ for each $i = 1, 2, \cdots, n$. If one of the a_{ii} entries is zero and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$
2. To speed convergence, the equations should be rearranged so that a_{ii} is as large as possible.
3. Another possible stopping criterion in Step 4 is to iterate until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|} < \varepsilon,$$

where $\varepsilon > 0$ is the tolerance. For this purpose, any convenient norm can be used, the usual being the l_∞ norm.

Convergence Analysis for Two Iterative Techniques

- To study the convergence of general iteration techniques, we consider the formula

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}.$$

  where $\mathbf{x}^{(0)}$ is arbitrary.

- **Lemma 3.44:** If the spectral radius $\rho(\mathbf{T})$ satisfies $\rho(\mathbf{T}) < 1$, then $(\mathbf{I} - \mathbf{T})^{-1}$ exists, and

$$(\mathbf{I} - \mathbf{T})^{-1} = \mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \cdots.$$

Convergence Analysis for Two Iterative Techniques

- To study the convergence of general iteration techniques, we consider the formula

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}.$$

where $\mathbf{x}^{(0)}$ is arbitrary.

- **Lemma 3.44:** If the spectral radius $\rho(\mathbf{T})$ satisfies $\rho(\mathbf{T}) < 1$, then $(\mathbf{I} - \mathbf{T})^{-1}$ exists, and

$$(\mathbf{I} - \mathbf{T})^{-1} = \mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \cdots .$$

Proof of Lemma 3.44:

- Since $T\mathbf{x} = \lambda\mathbf{x}$ is true precisely when

$$(\mathbf{I} - \mathbf{T})\mathbf{x} = (1 - \lambda)\mathbf{x},$$

  we have λ as an eigenvalue of $\mathbf{T}$ precisely when $1 - \lambda$ is an eigenvalue of $\mathbf{I} - \mathbf{T}$.
- But $|\lambda| \leq \rho(\mathbf{T}) < 1$, So $\lambda = 1$ is not an eigenvalue of $\mathbf{T}$, and 0 cannot be an eigenvalue of $\mathbf{I} - \mathbf{T}$.
- Hence $\mathbf{I} - \mathbf{T}$ is nonsingular. Let

$$\mathbf{S}_m = \mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \cdots + \mathbf{T}^m.$$

- Then

$$\begin{aligned}(\mathbf{I} - \mathbf{T})\mathbf{S}_m &= (\mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \cdots + \mathbf{T}^m) \\ &\quad -(\mathbf{T} + \mathbf{T}^2 + \cdots + \mathbf{T}^{m+1}) \\ &= \mathbf{I} - \mathbf{T}^{m+1}.\end{aligned}$$

- Since $\rho(\mathbf{T}) < 1$, then $\mathbf{T}$ is convergent and

$$\begin{aligned} \lim_{m\to\infty}(\mathbf{I}-\mathbf{T})\mathbf{S}_m &= \lim_{m\to\infty}(\mathbf{I}-\mathbf{T}^{m+1}) \\ &= \mathbf{I}. \end{aligned}$$

- Thus,

$$(\mathbf{I}-\mathbf{T})^{-1} = \lim_{m\to\infty}\mathbf{S}_m = \mathbf{I}+\mathbf{T}+\mathbf{T}^2+\cdots .\blacksquare\blacksquare\blacksquare.$$

Theorem 3.45

For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \texttt{for each} \quad k \geq 1. \tag{6}$$

converges to the unique solution of

$$\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$$

if and only if $\rho(\mathbf{T}) < 1$.

Proof of Theorem 3.45:

- First assume that $\rho(\mathbf{T}) < 1$.
- From Eq. (6),

$$\begin{aligned}
\mathbf{x}^{(k)} &= \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c} \\
&= \mathbf{T}(\mathbf{T}\mathbf{x}^{(k-2)} + \mathbf{c}) + \mathbf{c} \\
&= \mathbf{T}^2\mathbf{x}^{(k-2)} + (\mathbf{T} + \mathbf{I})\mathbf{c} \\
&\vdots \\
&= \mathbf{T}^k\mathbf{x}^{(0)} + (\mathbf{T}^{k-1} + \cdots + \mathbf{T} + \mathbf{I})\mathbf{c}
\end{aligned}$$

- Since $\rho(\mathbf{T}) < 1$, the matrix $\mathbf{T}$ is convergent and

$$\lim_{k\to\infty} \mathbf{T}^k\mathbf{x}^{(0)} = \mathbf{0}.$$

- Lemma 3.44 implies that

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

- Since

$$\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$$

implies that

$$(\mathbf{I}-\mathbf{T})\mathbf{x} = \mathbf{c},$$

the sequence $\{\mathbf{x}^{(k)}\}$ converges to the unique solution to the equation, the vector

$$\mathbf{x} = (\mathbf{I}-\mathbf{T})^{-1}\mathbf{c}.$$

- To prove the converse, we show that for any $\mathbf{z} \in \mathbb{R}^n$ we have

$$\lim_{k\to\infty} \mathbf{T}^k \mathbf{z} = 0$$

- Let $\mathbf{x}$ be the unique solution to the equation $\mathbf{x} = T\mathbf{x}$, that is, Eq. (6) with $\mathbf{c} = \mathbf{0}$.
- For $\mathbf{x}^{(0)} = \mathbf{x} - \mathbf{z}$, we have

$$\begin{aligned}
\lim_{k\to\infty} \mathbf{T}^k \mathbf{z} &= \lim_{k\to\infty} \mathbf{T}^k (\mathbf{x} - \mathbf{x}^{(0)}) \\
&= \lim_{k\to\infty} \mathbf{T}^{k-1} (\mathbf{T}\mathbf{x} - \mathbf{T}\mathbf{x}^{(0)}) \\
&= \lim_{k\to\infty} \mathbf{T}^{k-1} (\mathbf{x} - \mathbf{x}^{(1)}).
\end{aligned}$$

- Continuing in this manner, we have

$$\begin{aligned}\lim_{k\to\infty} \mathbf{T}^k\mathbf{z} &= \lim_{k\to\infty} \mathbf{T}^{k-1}(\mathbf{x}-\mathbf{x}^{(1)}) \\ &= \lim_{k\to\infty} \mathbf{T}^{k-2}(\mathbf{x}-\mathbf{x}^{(2)}) \\ &\vdots \\ &= \lim_{k\to\infty} (\mathbf{x}-\mathbf{x}^{(k)}) = \mathbf{0}.\end{aligned}$$

- Since $\mathbf{z} \in \mathbb{R}^n$ was arbitrary, Theorem 3.45 implies that $\mathbf{T}$ is a convergent matrix and that

$$\rho(\mathbf{T}) < 1.\blacksquare\blacksquare\blacksquare$$

Corollary 3.46

If $\|\mathbf{T}\| < 1$ for any natural matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}$$

converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, and the following error bounds hold:

1. $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|\mathbf{T}\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|$.
2. $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|\mathbf{T}\|^k}{1-\|\mathbf{T}\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$.

Convergence analysis on Jacobi and Gauss-seidel Iterative methods

- For Jacobi Iterative method, it can be written as

$$\mathbf{x}^{(k)} = \mathbf{T}_j \mathbf{x}^{(k-1)} + \mathbf{c}_j$$

  with $\mathbf{T}_j = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}), \mathbf{c}_j = \mathbf{D}^{-1}\mathbf{b}$,
- For Gauss-Seidel Iterative method, it can be written as

$$\mathbf{x}^{(k)} = \mathbf{T}_g \mathbf{x}^{(k-1)} + \mathbf{c}_g$$

  where $\mathbf{T}_g = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}, \mathbf{c}_g = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}$.
- If $\rho(\mathbf{T}_j)$ or $\rho(\mathbf{T}_g)$ is less than 1, then the corresponding sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ will converge to the solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$.

- For example, the Jacobi scheme has

$$\mathbf{x}^{(k)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k-1)} + \mathbf{D}^{-1}\mathbf{b},$$

- If $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ converges to $\mathbf{x}$, then

$$\mathbf{x} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{D}^{-1}\mathbf{b},$$

- This implies that

$$\mathbf{D}\mathbf{x} = (\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b},$$

and

$$(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{b}$$

- That is

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Sufficient conditions for convergence of the Jacobi and Gauss-Seidel methods.

Theorem 3.46

If $\mathbf{A}$ is **strictly diagonally dominant**, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss- Seidel methods give sequences $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge to the unique solution of $\mathbf{Ax} = \mathbf{b}$.

Theorem 3.47 (Stein-Rosenberg)

If $a_{ij} \leq 0$ for each $i \neq j$ and $a_{ii} > 0$ for each $i = 1, 2, \cdots, n$, then one and only one of the following statements holds:

a. $0 \leq \rho(\mathbf{T}_g) < \rho(\mathbf{T}_j) < 1$.

b. $1 < \rho(\mathbf{T}_j) < \rho(\mathbf{T}_g)$.

c. $\rho(\mathbf{T}_j) = \rho(\mathbf{T}_g) = 0$.

d. $\rho(\mathbf{T}_j) = \rho(\mathbf{T}_g) = 1$.

- For the special case described in Theorem 3.47, we see from part (a) that when one method gives convergence, then both give convergence, and the Gauss-Seidel method converges faster than the Jacobi method.
- Part (b) indicates that when one method diverges then both diverge, and the divergence is more pronounced for the Gauss-Seidel method.

Definition 3.48

- Suppose $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is an approximation to the solution of the linear system defined by $\mathbf{Ax} = \mathbf{b}$.
- The **residual vector** for $\tilde{\mathbf{x}}$ with respect to this system is $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$.

- The **object of the method** is to generate a sequence of approximations that will cause the associated **residual vectors to converge rapidly to zero.**

- Let

$$\mathbf{r}_i^{(k)} = \left(r_{1i}^{(k)}, r_{2i}^{(k)}, \cdots, r_{ni}^{(k)}\right)^T$$

denote the **residual vector** for the Gauss-Seidel method corresponding to the approximate solution vector $\mathbf{x}_i^{(k)}$, defined by

$$\mathbf{x}_i^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \cdots, x_{i-1}^{(k)}, x_i^{(k-1)}, \cdots, x_n^{(k-1)}\right)^T$$

- Thus the **residual vector** for $\mathbf{x}_i^{(k)}$ with respect to this system is

$$\mathbf{r}_i^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}_i^{(k)}.$$

- The mth component of $\mathbf{r}_i^{(k)}$ is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^{n} a_{mj} x_j^{(k-1)}, \qquad (7)$$

- Or, equivalently,

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

for each $m = 1, 2, \cdots, n$.

- In particular, the ith component of $\mathbf{r}_i^{(k)}$ is

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

so

$$a_{ii} x_i^{(k-1)} + r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}, \qquad (8)$$

- Recall, however, that in the Gauss-Seidel method, $x_i^{(k)}$ is chosen to be

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right]. \quad (9)$$

- so Eq. (8) can be rewritten as

$$a_{ii} x_i^{(k-1)} + r_{ii}^{(k)} = a_{ii} x_i^{(k)}. \quad (10)$$

- Consequently, the Gauss-Seidel method can be characterized as choosing $x_i^{(k)}$ to satisfy

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}. \quad (11)$$

- We can derive another connection between the residual vectors and the Gauss- Seidel technique.
- Consider the residual vector $\mathbf{r}_{i+1}^{(k)}$ associated with the vector

$$\mathbf{x}_{i+1}^{(k)} = \left(x_1^{(k)}, \cdots, x_i^{(k)}, x_{i+1}^{(k-1)}, \cdots, x_n^{(k-1)}\right)^T,$$

- the ith component of $\mathbf{r}_{i+1}^{(k)}$ is

$$\begin{aligned} r_{i,i+1}^{(k)} &= b_i - \sum_{j=1}^{i} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \\ &= b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} - a_{ii} x_i^{(k)}. \end{aligned}$$

- Above Equation (11) implies that $r_{i,i+1}^{(k)} = 0$.
- the Gauss-Seidel technique is also characterized by choosing $x_i^{(k)}$ in such a way that the ith component of $\mathbf{r}_{i+1}^{(k)}$ is zero.

IV. Relaxation Methods

- Reducing one coordinate of the residual vector to zero, however, is not generally the most efficient way to reduce the overall size of the vector $\mathbf{r}_{i+1}^{(k)}$.
- Instead, we need to choose $x_i^{(k)}$ so that $\|\mathbf{r}_{i+1}^{(k)}\|$ is small.
- Modifying the Gauss-Seidel procedure as given to

$$x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}}. \tag{12}$$

for certain choices of positive ω reduces the norm of the residual vector and leads to significantly faster convergence.

- Methods involving Eq. (12) are called **relaxation methods**.
- For choices of ω with $0 < \omega < 1$, the procedures are called **under-relaxation methods** and can be used to obtain convergence of some systems that are not convergent by the Gauss-Seidel method.
- For choices of ω with $1 < \omega$, the procedures are called **over-relaxation methods**, which are used to accelerate the convergence for systems that are convergent by the Gauss-Seidel technique.
- These methods are abbreviated **SOR**, for **Successive Over-Relaxation**, and are particularly useful for solving the linear systems that occur in the numerical solution of certain partial-differential equations.

- Before illustrating the advantages of the SOR method, with $m = i$, can be reformulated for calculation purposes to

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

- To determine the matrix of the SOR method, we rewrite this as

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

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

- so

$$(\mathbf{D} - \omega\mathbf{L})\mathbf{x}^{(k)} = [(1-\omega)\mathbf{D} + \omega\mathbf{U}]\mathbf{x}^{(k-1)} + \omega\mathbf{b}.$$

- or

$$\begin{aligned}\mathbf{x}^{(k)} &= (\mathbf{D} - \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega\mathbf{U}]\mathbf{x}^{(k-1)} \\ &\quad +\omega(\mathbf{D} - \omega\mathbf{L})^{-1}\mathbf{b}.\end{aligned}$$

- If we let

$$\mathbf{T}_\omega = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega\mathbf{U}]$$

and

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

we can express the SOR technique in the form

$$\mathbf{x}^{(k)} = \mathbf{T}_\omega\mathbf{x}^{(k-1)} + \mathbf{c}_\omega.$$

Theorem 3.49 (Kahan)

If $a_{ii} \neq 0$ for each $i = 1, 2, \cdots, n$, then $\rho(\mathbf{T}_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$.

Theorem 3.50 (Ostrowski-Reich)

If $\mathbf{A}$ is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$.

Theorem 7.26

If $\mathbf{A}$ is positive definite and tridiagonal, then $\rho(\mathbf{T}_g) = [\rho(\mathbf{T}_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.$$

with this choice of ω, we have $\rho(\mathbf{T}_\omega) = \omega - 1$.

III. SOR Algorithm:

To solve $\mathbf{Ax} = \mathbf{b}$ given the parameter ω and an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries $a_{ij}, 1 \leq i, j \leq n$, of the matrix A; the entries $b_i, 1 \leq i \leq n$, of $\mathbf{b}$; the entries $XO_i, 1 \leq i \leq n$, of $\mathbf{XO} = \mathbf{x}^{(0)}$; the parameter ω; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \cdots, x_n$ or a message that the number of iterations was exceeded.

III. SOR Algorithm:

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3-6.

- **Step 3** For $i = 1, \cdots, n$ set

$$x_i = (1-\omega)XO_i + \frac{\omega(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j) + b_i}{a_{ii}}$$

- **Step 4** If $\|\mathbf{x} - \mathbf{XO}\| < TOL$, then OUTPUT $(x_1, \cdots, x_n)$; (Procedure completed successfully.) STOP.
- **Step 5** Set $k = k + 1$.
- **Step 6** For $i = 1, \cdots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded'); (Procedure completed unsuccessfully.) STOP.