Chapter 5 Numerical methods for system of linear equations

School of science

Contents

Introduction

- 1. Numerical method for linear equations
- 2. Direct method v,s. iterative method

Introduction

- 1. Numerical method for linear equations
  - Almost half of the problems in engineering involve the solution of linear equations
  - Suppose the *n-th* order linear equations:

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

or $\boldsymbol{Ax=b}$

where $A = (a_{ij})_{n\times n}$ $\quad b = (b_j)_{n\times 1}$ $\quad x = (x_i)_{n\times 1}$

Introduction

- A is called the coefficient matrix of the equations, and when it is a nonsingular matrix of order n, i.e., $|A| \neq 0$, the equations have a unique solution.
- X is the solution vector, and B is a constant vector
- In linear algebra, we've learned to solve the problem by Cramer's rule, which is a direct method (belonging to analytical method). While, with $n\uparrow$, the amount of computation$\uparrow$

Introduction

Cremer’s rule

when $\det|A| \neq 0$, the equations has only one solution:

$$x_i = \frac{\det(A_i)}{\det(A)} \quad (i = 1,2,\cdots,n)$$

computation cost $\approx$ (n-1)*(n+1)!

Very large, not feasible when n is large enough.

Introduction

- Direct method v,s. iterative method
  - Direct method : the exact solution is obtained by finite calculation, suitable for small coefficient matrix with little computation cost.
  - Iterative method : the iterative method transforms the problem into an infinite sequence and approximates the exact solution. It is suitable for very large coefficient matrix, but has the problem of convergence and requires large amount of computation

Introduction

- In application, the choice of method shall be determined according to the characteristics and requirements of the problem
- In this chapter, direct method, such as Gaussian elimination and LU factorization will be introduced. Meanwhile, the iterative method, e.g., *Jacobi* iteration, *Gauss-Seidel* iteration will also be introduced.

§ 5.1 Gaussian elimination

contents

- 1. introduction
- 2. Gaussian elimination
- 3. computation cost

§ 5.1 Gaussian elimination

- Introduction
  - It is an ancient method for solving linear equations
  - A direct method for n-array linear equations

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

§ 5.1 Gaussian elimination

Solving $Ax=b$, $x=?$

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

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$$

§ 5.1 Gaussian elimination

- Gaussian elimination

denote: $\mathbf{A}^{(1)} = \mathbf{A}, \mathbf{b}^{(1)} = \mathbf{b}, a_{ij}^{(1)} = a_{ij}, b_i^{(1)} = b_i$

thus the augmented matrix of matrix A is

$$(\mathbf{A}^{(1)}, \mathbf{b}^{(1)}) = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & b_3^{(1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}$$

§ 5.1 Gaussian elimination

First step:

suppose $a_{11}^{(1)} \neq 0$, take $-m_{i1} = -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$, $(i = 2,3,\ldots,n)$

multiplication the 1-th row then adding to the i-th row, thus

$$(\mathrm{A}^{(2)},\mathrm{b}^{(2)}) = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \ldots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \ldots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} & b_3^{(2)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \ldots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

where

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)}, \quad i,j = 2,3,\ldots,n$$

$$b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)}, \quad i = 2,3,\ldots,n$$

§ 5.1 Gaussian elimination

Second step:

suppose $a_{22}^{(2)} \neq 0$, take $-m_{i2} = -\dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}}$, $(i = 3,4,\ldots,n)$

multiplication the 2-th row then adding to the i-th row, thus

$$(\mathbf{A}^{(3)}, \mathbf{b}^{(3)}) = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \ldots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \ldots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \ldots & a_{3n}^{(3)} & b_3^{(3)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & a_{n3}^{(3)} & \ldots & a_{nn}^{(3)} & b_n^{(3)} \end{pmatrix}$$

where

$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2} a_{2j}^{(2)}, \quad i, j = 3,4,\ldots,n$$

$$b_i^{(3)} = b_i^{(2)} - m_{i2} b_2^{(2)}, \quad i = 3,4,\ldots,n$$

§ 5.1 Gaussian elimination

continue the elimination process, and after n-1 times we have:

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

The elimination process is done

§ 5.1 Gaussian elimination

At this time, the original equations change to

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \ldots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ \qquad\qquad a_{22}^{(2)}x_2 + \ldots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad\qquad\qquad\qquad\qquad a_{nn}^{(n)}x_n = b_n^{(n)} \end{cases}$$

Back substitution the equations above, we can get

$$\begin{cases} x_n = \dfrac{b_n^{(n)}}{a_{nn}^{(n)}} \\ x_i = \dfrac{\left(b_i^{(i)} - \sum\limits_{j=i+1}^{n} a_{ij}^{(i)} x_j\right)}{a_{ii}^{(i)}}, i = n-1, n-2, \cdots, 1 \end{cases}$$

$a_{kk}^{(k)}$ is called the reduced main element

§ 5.1 Gaussian elimination

Conclusion

- Gaussian elimination = elimination (upper triangle matrix) + back substitution (lower triangle matrix)

- It must satisfies $a_{kk}^{(k)} \neq 0, (k=1,2,\ldots,n)$, during the elimination. If some $a_{kk}^{(k)}=0$, the row and column should be exchanged before elimination

§ 5.1 Gaussian elimination

Computation cost

elimination calculation

k step	division	multiplication(A)	multiplication	sum and subtraction
1	$n-1$	$(n-1)^2$	$n-1$	$(n-1)^2$
2	$n-2$	$(n-2)^2$	$n-2$	$(n-2)^2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	1	1	1	1
total	$n(n-1)/2$	$n(n-1)(2n-1)/6$	$n(n-1)/2$	$n(n-1)(2n-1)/6$

$(n-k)^2$

$(n-k)$

computation cost in total：

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

$$=\frac{n(n-1)}{2}+\frac{n(n-1)(2n-1)}{6}+\frac{n(n-1)}{2}=\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}$$

§ 5.1 Gaussian elimination

Computation cost

back substitution calculation needs:

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

Total computation cost of Gaussian elimination to solve $Ax=b$ is :

$$\frac{n(n+1)}{2}+\left(\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}\right)=\frac{n^3}{3}+n^2-\frac{n}{3}=O(n^3)$$

§ 5.1 Gaussian elimination

Conclusion

- Gaussian elimination = elimination (upper triangle matrix) + back substitution (lower triangle matrix)

- The total computation cost is $O(n^3)$

- It must satisfies $a_{kk}^{(k)} \neq 0, (k=1,2,\ldots,n)$, during the elimination. If some $a_{kk}^{(k)}=0$, the row and column should be exchanged before elimination

§ 5.1 Gaussian elimination

Example : solve the equation by Gaussian elimination

$$\begin{cases} x_1 + \quad x_2 + x_3 = 6 \\ \qquad 4x_2 - x_3 = 5 \\ 2x_1 - 2x_2 + x_3 = 1 \end{cases}$$

Solution : solving with the use of augmented matrix of A

$$(A \mid b) = \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 2 & -2 & 1 & 1 \end{bmatrix} \xrightarrow{r_3 - 2*r_1 \to r_3} \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & -4 & -1 & -11 \end{bmatrix}$$

$$\xrightarrow{r_3 - (-1)*r_2 \to r_3} \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & -2 & -6 \end{bmatrix}$$

Contents

- §5.2 LU factorization

§ 5.2 LU factorization

Solving $Ax=b$, $x=?$

- 1. calculate : $L_{n-1} L_{n-2} \dots L_1 A = U$
- 2. denote: $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$
- 3. thus: $A = LU$
- 4. compute : $LU = b$
- 5. solve : $Ly = b$, $Ux = y$

§ 5.2 LU factorization

For the matrix *A* with order n

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

§ 5.2 LU factorization

suppose $a_{11}^{(1)} \neq 0$, set $m_{i1} = a_{i1}^{(1)} \div a_{11}^{(1)}, i = 2,3,\cdots,n,$ denote

$$L_1 = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & & 1 & & \\ \vdots & & & \ddots & \\ -m_{n1} & & & & 1 \end{pmatrix}$$

thus $$A^{(2)} = L_1 A^{(1)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

§ 5.2 LU factorization

suppose $a_{22}^{(2)} \neq 0$, set $m_{i2} = a_{i2}^{(2)} \div a_{22}^{(2)}, i = 3,4,\cdots,n$, denote

$$L_2 = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & -m_{32} & 1 & & \\ & \vdots & & \ddots & \\ & -m_{n2} & & & 1 \end{pmatrix}$$

thus $$A^{(3)} = L_2 A^{(2)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{pmatrix}$$

§ 5.2 LU factorization

Following the same process, and at (n-1)-th step:

$$\boldsymbol{A}^{(n)} = \boldsymbol{L}_{n-1}\boldsymbol{A}^{(n-1)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{pmatrix}$$

where $$\boldsymbol{L}_{n-1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -m_{n,n-1} & 1 \end{pmatrix}$$

§ 5.2 LU factorization

Thus:

$$A^{(n)}=L_{n-1}A^{(n-1)}=L_{n-1}L_{n-2}A^{(n-2)}=\ldots=L_{n-1}L_{n-2}\ldots L_2L_1A^{(1)}$$

where

$$\mathrm{L}_k=\begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -m_{k+1k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -m_{nk} & & & 1 \end{pmatrix} \begin{array}{l} \\ \\ \\ \leftarrow \text{ the k-th row} \\ \\ \\ \end{array} ,k=1,2,\cdots,n-1$$

so that: $$A=A^{(1)}=L_1^{-1}L_2^{-1}\ldots L_{n-1}^{-1}A^{(n)}=LU$$

where $$L=L_1^{-1}L_2^{-1}\ldots L_{n-1}^{-1},\quad U=A^{(n)}$$

§ 5.2 LU factorization

Moreover:

$$\mathbf{L}_k^{-1}=\begin{pmatrix}1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{k+1k} & 1 & & \\ & & \vdots & & \ddots & \\ & & m_{nk} & & & 1\end{pmatrix},\quad \mathbf{L}=\begin{pmatrix}1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ m_{n1} & m_{n2} & m_{n3} & \cdots & 1\end{pmatrix}$$

L is called the unit lower triangular matrix;

U is the upper triangle matrix.

$$\mathbf{U}=\begin{pmatrix}a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)}\end{pmatrix}$$

$A=LU$ is the LU factorization of matrix A (Doolittle decomposition)

§ 5.2 LU factorization

- **Theorem:** suppose the sequential principal minors of the matrix A of order n is nonzero, then there exists the unique unit lower triangle matrix L and upper triangle matrix U , thus satisfies $A=LU$.

- **Proof:** suppose there are two decomposition

$$\mathbf{A} = \mathbf{L}\mathbf{U} = \overline{\mathbf{L}}\,\overline{\mathbf{U}}$$

as $\overline{\mathbf{L}}^{-1}\mathbf{L} = \overline{\mathbf{U}}\mathbf{U}^{-1} = \mathbf{E}$ thus $\mathbf{L} = \overline{\mathbf{L}}$， $\mathbf{U} = \overline{\mathbf{U}}$。

so Ax=b $\Rightarrow$ LUx=b , and set Ux=y, thus $\begin{cases} \mathbf{L}\mathbf{y} = \mathbf{b} \\ \mathbf{U}\mathbf{x} = \mathbf{y} \end{cases}$

§ 5.2 LU factorization

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & & a_{3n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} =$$

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

K=1

We have: $u_{1j} = a_{1j} \quad j = 1,2,\cdots n$

$m_{i1} = a_{i1} \div u_{11} \quad i = 2,3,\cdots,n$

§ 5.2 LU factorization

For k=2,3,…,n, calculation U :

If already known the (i-1)-th row of U and (i-1)-th column of L, by the matrix multiplication we have

$$a_{ij} = m_{i1}u_{1j} + m_{i2}u_{2j} + m_{i3}u_{3j} + \cdots + m_{in}u_{nj}$$

$$= \sum_{k=1}^{n} m_{ik}u_{kj} = \sum_{k=1}^{i-1} m_{ik}u_{kj} + u_{ij}, \quad j \geq i$$

where: $$u_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik}u_{kj} \quad j = i, i+1, \cdots, n$$

Find the element in i-th row of U

§ 5.2 LU factorization

For k=2,3,…,n, calculation L :

and: $$a_{ij} = m_{i1}u_{1j} + m_{i2}u_{2j} + m_{i3}u_{3j} + \cdots + m_{in}u_{nj}$$

$$= \sum_{k=1}^{j-1} m_{ik}u_{kj} + m_{ij}u_{jj}, \quad i = j+1, \cdots, n$$

if: $u_{ij} \neq 0,$

we have: $$m_{ij} = \frac{1}{u_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} m_{ik}u_{kj}\right) \quad , i = j+1, \cdots, n$$

Find the element in i-th column of L

§ 5.2 LU factorization

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1k-1} & u_{1k} & \cdots & u_{1n} \\ m_{21} & u_{22} & \cdots & u_{2k-1} & u_{2k} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ m_{k-11} & m_{k-12} & \cdots & u_{k-1k-1} & u_{k-1k} & \cdots & u_{k-1n} \\ m_{k1} & m_{k2} & \cdots & m_{kk-1} & u_{kk} & \cdots & u_{kn} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nk-1} & m_{nk} & \cdots & u_{nn} \end{pmatrix}$$

Doolittle decomposition

$$\begin{cases} u_{1j} = a_{1j} \qquad j = 1,2,\cdots,n \\ m_{i1} = a_{i1} \div u_{11} \qquad i = 2,3,\cdots,n \\ \text{for k=2,3,…n, calculate :} \\ u_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \quad j = i, i+1, \cdots, n \\ m_{ij} = \dfrac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \right) \quad , i = j+1, \cdots, n \end{cases}$$

§ 5.2 LU factorization

According to :

$$\begin{pmatrix} 1 & & & \\ m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ m_{n1} & m_{n2} & \dots & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ & u_{22} & \dots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

we have :

$$y_1 = b_1 \qquad y_i = b_i - \sum_{j=1}^{i-1} m_{ij} y_j \quad, i = 2,3,\cdots,n$$

$$x_n = y_n \div u_{nn} \qquad x_i = \frac{1}{u_{ii}}(y_i - \sum_{j=i+1}^{n} u_{ij} x_j), i = n-1,\cdots,2,1$$

This is the LU factorization for solving equations Ax=b

§ 5.2 LU factorization

Example: solve linear equations with LU factorization

$$A=\begin{bmatrix}2 & -1 & 3\\ 1 & 2 & 1\\ 2 & 4 & 3\end{bmatrix} \quad A^{(1)}=\begin{bmatrix}2 & -1 & 3\\ 0 & \frac{5}{2} & -\frac{1}{2}\\ 0 & 5 & 0\end{bmatrix} \quad A^{(2)}=\begin{bmatrix}2 & -1 & 3\\ 0 & \frac{5}{2} & -\frac{1}{2}\\ 0 & 0 & 1\end{bmatrix}$$

$$L_1=\begin{bmatrix}1 & & \\ -\frac{1}{2} & 1 & \\ -1 & & 1\end{bmatrix} \quad L_2=\begin{bmatrix}1 & & \\ & 1 & \\ & -2 & 1\end{bmatrix}$$

$$U=A^{(2)}=L_2L_1A$$

§ 5.2 LU factorization

Example: solve linear equations with LU factorization

$$A=\begin{bmatrix}2 & -1 & 3\\ 1 & 2 & 1\\ 2 & 4 & 3\end{bmatrix}\qquad L_1=\begin{bmatrix}1 & & \\ -\frac{1}{2} & 1 & \\ -1 & & 1\end{bmatrix}\qquad L_2=\begin{bmatrix}1 & & \\ & 1 & \\ & -2 & 1\end{bmatrix}$$

$$U=\begin{bmatrix}2 & -1 & 3\\ 0 & \frac{5}{2} & -\frac{1}{2}\\ 0 & 0 & 1\end{bmatrix}\qquad L_1^{-1}=\begin{bmatrix}1 & & \\ \frac{1}{2} & 1 & \\ 1 & & 1\end{bmatrix}\qquad L_2^{-1}=\begin{bmatrix}1 & & \\ & 1 & \\ & 2 & 1\end{bmatrix}$$

$$L=L_1^{-1}L_2^{-1}\qquad A=LU$$

§ 5.2 LU factorization

- The computation cost of **LU factorization** of linear equations $Ax=b$ is about $\boldsymbol{O(1/3n^3)}$ **,** which is almost equal to the computation cost of **Gaussian elimination**
- The advantage of LU factorization is that when solving a sequential linear equations $Ax=b_k$, (k=1,2,…,m) with the same coefficient matrix A, it can greatly save the amount of computation.

§ 5.2 supplement (with partial pivot)

Problem:

- $a_{kk}^{(k)} = 0$ or $a_{kk}^{(k)}$ is small enough may appear in Gaussian elimination and LU factorization. In that case, traditional calculation process could not continue to go on otherwise may cause very large rounding error.

- To avoid this, elements with large absolute values can be selected as principal elements by exchanging the order of equations, which leads to the idea of partial pivoting.

§ 5.2 supplement (row pivot)

Example : LU factorization with row pivot

$$A = \begin{bmatrix} 0 & -1 & 2 \\ 3 & -\frac{1}{2} & 1 \\ 1 & 4 & 2 \end{bmatrix} \qquad \widetilde{A}^{(1)} = \begin{bmatrix} 2 & -1 & 0 \\ 1 & -\frac{1}{2} & 3 \\ 2 & 4 & 1 \end{bmatrix} \qquad A^{(1)} = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 0 & 3 \\ 0 & 5 & 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0 & & 1 \\ & 1 & \\ 1 & & 0 \end{bmatrix} \qquad L_1 = \begin{bmatrix} 1 & & \\ -\frac{1}{2} & 1 & \\ -1 & & 1 \end{bmatrix}$$

$$\widetilde{A}^{(1)} = AP_1 \qquad A^{(1)} = L_1\widetilde{A}^{(1)} \qquad A^{(1)} = L_1 A P_1$$

§ 5.2 supplement (row pivot)

Example : LU factorization with row pivot

$$A^{(1)} = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 0 & 3 \\ 0 & 5 & 1 \end{bmatrix} \qquad \widetilde{A}^{(2)} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & 1 & 5 \end{bmatrix} \qquad A^{(2)} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 1 & & \\ & 0 & 1 \\ & 1 & 0 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{1}{3} & 1 \end{bmatrix}$$

$$\widetilde{A}^{(2)} = A^{(1)} P_2 \qquad A^{(2)} = L_2 \widetilde{A}^{(2)} \qquad A^{(2)} = L_2 A^{(1)} P_2$$

§ 5.2 supplement (row pivot)

Solving $Ax=b$, $x=?$

- 1. calculate : $L_{n-1} L_{n-2} L_1 A P_1 \dots P_{n-2} P_{n-1} = U$
- 2. denote: $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$, $P = P_1 \dots P_{n-2} P_{n-1}$
- 3. thus: $A = LUP^{-1}$
- 4. compute : $LUP^{-1}x = b$
- 5. solve : $Ly=b$, $Uz=y$, $x=Pz$

§ 5.2 supplement (column pivot)

Example : Gaussian elimination/LU factorization with column pivoting

$$A = \begin{bmatrix} 0 & -1 & 2 \\ 3 & -\frac{1}{2} & 1 \\ 1 & 4 & 2 \end{bmatrix} \qquad \widetilde{A}^{(1)} = \begin{bmatrix} 3 & -\frac{1}{2} & 1 \\ 0 & -1 & 2 \\ 1 & 4 & 2 \end{bmatrix} \qquad A^{(1)} = \begin{bmatrix} 3 & -\frac{1}{2} & 1 \\ 0 & -1 & 2 \\ 0 & \frac{25}{6} & \frac{5}{3} \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0 & 1 & \\ 1 & 0 & \\ & & 1 \end{bmatrix} \qquad L_1 = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ -\frac{1}{3} & & 1 \end{bmatrix}$$

$$\widetilde{A}^{(1)} = P_1 A \qquad A^{(1)} = L_1 \widetilde{A}^{(1)} \qquad A^{(1)} = L_1 P_1 A$$

§ 5.2 supplement (column pivot)

Example : Gaussian elimination/LU factorization with column pivoting

$$A^{(1)} = \begin{bmatrix} 3 & -\frac{1}{2} & 1 \\ 0 & -1 & 2 \\ 0 & \frac{25}{6} & \frac{5}{3} \end{bmatrix} \quad \widetilde{A}^{(2)} = \begin{bmatrix} 3 & -\frac{1}{2} & 1 \\ 0 & \frac{25}{6} & \frac{5}{3} \\ 0 & -1 & 2 \end{bmatrix} \quad A^{(2)} = \begin{bmatrix} 3 & -\frac{1}{2} & 1 \\ 0 & \frac{25}{6} & \frac{5}{3} \\ 0 & 0 & \frac{12}{5} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 1 & & \\ & 0 & 1 \\ & 1 & 0 \end{bmatrix} \quad L_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & \frac{6}{25} & 1 \end{bmatrix}$$

$$\widetilde{A}^{(2)} = P_2 A^{(1)} \qquad A^{(2)} = L_2 \widetilde{A}^{(2)} \qquad A^{(2)} = L_2 P_2 A^{(1)}$$

§ 5.2 supplement (column pivot)

Solving $Ax=b$, $x=?$

- 1. calculate : $L_3P_3L_2P_2L_1P_1A=U$
- 2. calculate : $L_3P_3L_2P_2L_1P_2P_3*(P_3P_2P_1)=U$
- 3. denote : $L_3P_3L_2P_2L_1P_2P_3*PA=U$

$$L= P_3P_2L_1^{-1}P_2\ L_2^{-1}P_3L_3^{-1}$$

- 3. thus: $PA=LU$ compute $PAx=LUx=Pb$
- 4. solve : $Ly=Pb$, $Ux=y$

§ 5.2 supplement (column pivot)

In MATLAB function lu():

$$[L, U, P] = lu(A)$$

Is the column pivot that satisfies: $PA=LU$

- §5.3 Solving tridiagonal matrix

§ 5.3 Solving tridiagonal matrix

- tridiagonal matrix is very important in the interpolation and boundary value problems.

$$Ax = b$$

where

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

sub diagonal

sub diagonal

diagonal element

A is the tridiagonal matrix

§ 5.3 Solving tridiagonal matrix

Obviously, A is nonsingular, thus the LU factorization gives:

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}$$

Crout decomposition

bidiagonal matrix

$$= \begin{bmatrix} \gamma_1 & & & & \\ \beta_2 & \gamma_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_{n-1} & \gamma_{n-1} & \\ & & & \beta_n & \gamma_n \end{bmatrix} \begin{bmatrix} 1 & \delta_1 & & & \\ & 1 & \delta_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \delta_{n-1} \\ & & & & 1 \end{bmatrix}$$

§ 5.3 Solving tridiagonal matrix

So that we have :

$$\begin{cases} \gamma_1 = b_1, \quad \delta_1 = \dfrac{c_1}{\gamma_1} \\ \beta_i = a_1, \quad \gamma_i = b_i - \beta_i \delta_{i-1} \quad (i = 2,3,\cdots,n) \\ \delta_i = \dfrac{c_i}{\gamma_i} \quad (i = 2,3,\cdots,n-1) \end{cases}$$

$$Ax = b \Leftrightarrow LUx = b \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

Quasi-diagonal linear equations

§ 5.3 Solving tridiagonal matrix

Solving $Ly=b$ that we have :

$$\begin{bmatrix} \gamma_1 & & & & \\ \beta_2 & \gamma_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_{n-1} & \gamma_{n-1} & \\ & & & \beta_n & \gamma_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

$$y_1 = \frac{b_1}{\gamma_1}, \quad y_i = \frac{b_i - \beta_i y_{i-1}}{\gamma_i} \quad (i = 2,3,\cdots,n)$$

equivalent to elimination process

§ 5.3 Solving tridiagonal matrix

Solving $Ux=y$ that we have :

$$\begin{bmatrix} 1 & \delta_1 & & & \\ & 1 & \delta_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \delta_{n-1} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_{n-1} \\ y_n \end{bmatrix}$$

$$x_n = y_n, \quad x_i = y_i - \delta_i x_{i+1} \quad (i = n-1, \cdots, 1)$$

equivalent to back substitution process

Also named Thomas method

- §5.4 Norm and the state of equations

Review:

Full rank of matrix *A*

rank(A)=n:

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

§ 5.4 vector norm

A *norm* is a function $\|\cdot\| : \mathbb{C}^m \to \mathbb{R}$ that assigns a real-valued length to each vector. In order to conform to a reasonable notion of length, a norm must satisfy the following three conditions. For all vectors x and y and for all scalars $\alpha \in \mathbb{C}$,

$$\begin{aligned} &(1) \quad \|x\| \geq 0, \text{ and } \|x\| = 0 \text{ only if } x = 0, \\ &(2) \quad \|x + y\| \leq \|x\| + \|y\|, \\ &(3) \quad \|\alpha x\| = |\alpha| \, \|x\|. \end{aligned} \tag{3.1}$$

In words， the three conditions means

(1) positive;

(2) triangle inequality;

(3) scaling a vector scales by the same amount

§ 5.4 vector norm

$$\|x\|_2 = \sqrt{x_1{}^2 + \cdots + x_n{}^2} = \left(\sum_{i=1}^{n} x_i{}^2\right)^{\frac{1}{2}}$$

$$\|x\|_\infty = \max\{\,|x_1|, \cdots, |x_n|\,\} = \max_{1\le i\le n}\{\,|x_i|\,\}$$

$$\|x\|_1 = |x_1| + \cdots + |x_n| = \sum_{i=1}^{n} |x_i|$$

$$* \qquad \|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \qquad 1 \le p < \infty$$

§ 5.4 vector norm

Example :

Calculate the 1-norm, 2-norm and max-norm of the vector $x = (1,2,3)^T$

§ 5.4 vector norm

Example :

Calculate the 1-norm, 2-norm and max-norm of the vector $x = (1,2,3)^T$

Solution: $\|x\|_1 = 6,\ \|x\|_\infty = 3,\ \|x\|_2 = \sqrt{14}$。

§ 5.4 vector norm

For the two norms $\|\cdot\|$ and $\|\cdot\|'$ in R^n, if there exist two real number m, M>0 that for any vector $x \in R^n$ satisfies

$$m\|x\| \le \|x\|' \le M\|x\|,$$

thus, the two norms are equivalent.

It is very easy to know that 1-norm, 2-norm and max-norm are equivalent.

§ 5.4 vector norm

Example：

$$\|x\|_\infty = \max_{1\le i\le n}\{\ |x_i|\ \} \le \sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2} = \|x\|_2。$$

$$such\ as\ \|x\|_\infty = |x_j| = \max_{1\le i\le n}\{\ |x_i|\ \}$$

$$thus\ \|x\|_\infty = |x_j| \ge \sqrt{\frac{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}{n}} = \frac{\|x\|_2}{\sqrt{n}}$$

$$\Rightarrow \frac{\|x\|_2}{\sqrt{n}} \le \|x\|_\infty \le \|x\|_2$$

$2 - norm$ is equivalent $to\ \infty - norm$。

§ 5.4 matrix norm

- Matrix norm induced by vector norm

suppose A is a $m \times n$ matrix, the matrix norm can be defined equivalently in terms of the unit vector under A :

$$\|A\|_{(m,n)} = \sup_{\substack{x \in \mathbb{C}^n \\ x \neq 0}} \frac{\|Ax\|_{(m)}}{\|x\|_{(n)}} = \sup_{\substack{x \in \mathbb{C}^n \\ \|x\|_{(n)} = 1}} \|Ax\|_{(m)}.$$

§ 5.4 matrix norm

General matrix norm of a n-dimensional matrix A must satisfies the following conditions

$$(1)\quad \|A\| \geq 0, \text{ and } \|A\| = 0 \text{ only if } A = 0,$$

$$(2)\quad \|A + B\| \leq \|A\| + \|B\|,$$

$$(3)\quad \|\alpha A\| = |\alpha|\, \|A\|.$$

(1) positive;
(2) triangle inequality;
(3) scaling a vector scales by the same amount

§ 5.4 matrix norm

Example :

1-norm of matrix A is equal to the "maximum column sum" of A. write A in terms of its columns

$$A = \left[\begin{array}{c|c|c} a_1 & \cdots & a_n \end{array} \right],$$

$$\|Ax\|_1 = \left\|\sum_{j=1}^{n} x_j a_j\right\|_1 \le \sum_{j=1}^{n} |x_j| \, \|a_j\|_1 \le \max_{1\le j\le n} \|a_j\|_1.$$

choose $x_j = e_j$

$$\|A\|_1 = \max_j \left\{ \sum_{i=1}^{m} |a_{ij}| \right\}$$

§ 5.4 matrix norm

Similarly, the ∞-norm of matrix A is equal to the "maximum row sum" of A. write A in terms of its rows

$$\|A\|_\infty = \max_i \left\{ \sum_{j=1}^{n} |a_{ij}| \right\}$$

The 2-norm of matrix A is called the spectral norm

$$\|A\|_2 = \sqrt{\lambda_1}, \lambda_1 = \max\left\{ \lambda\left(A^H A \right) \right\}$$

§ 5.4 matrix norm

The Hilbert-Schmidt or Frobenius norm, defined by

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$

Bounding $\|AB\|$ in an Induced Matrix Norm

The induced matrix norm of a matrix product can also be bounded. Let $\|\cdot\|_{(\ell)}$, $\|\cdot\|_{(m)}$, and $\|\cdot\|_{(n)}$ be norms on $\mathbb{C}^l$, $\mathbb{C}^m$, and $\mathbb{C}^n$, respectively, and let A be an $l \times m$ matrix and B an $m \times n$ matrix. For any $x \in \mathbb{C}^n$ we have

$$\|ABx\|_{(\ell)} \le \|A\|_{(\ell,m)} \|Bx\|_{(m)} \le \|A\|_{(\ell,m)} \|B\|_{(m,n)} \|x\|_{(n)}.$$

Therefore the induced norm of AB must satisfy

$$\|AB\|_{(\ell,n)} \le \|A\|_{(\ell,m)} \|B\|_{(m,n)}. \tag{3.14}$$

§ 5.4 matrix norm

Example :

Find the $\|A\|_1$, $\|A\|_2$ and $\|A\|_\infty$ norm of the matrix A=(-1,2,1)

Solution :

$$\|A\|_1 = \max\{|-1|, 2, , 1\} = 2$$
$$\|A\|_\infty = |-1| + 2 + 1 = 4$$
$$A^T A = AA^T = 6, thus\ \|A\|_2 = \sqrt{6}$$

§ 5.4 matrix norm

The spectral radius of the matrix A , defined by

$$\rho(A) = \max_{1 \le i \le n} |\lambda_i|$$

where λ_i (i=1,2,…,n) is the eigenvalue of A

Theorem：

$$\rho(A) \le \|A\| \text{ for any norm of A}$$

$$(\|Ax\| = \|\lambda x\| = |\lambda|\,\|x\|), \|Ax\| \le \|A\| \cdot \|x\|$$

$$\Rightarrow |\lambda|\,\|x\| \le \|A\| \cdot \|x\| \Rightarrow |\lambda| \le \|A\| \Rightarrow \rho(A) \le \|A\|$$

Condition number: $cond(A) = \|A\| \cdot \|A^{-1}\|$

- §5.5 Iterative method

§ 5.5 iterative method for linear equations

- contents
  - 1. Introduction
  - 2. Jacobi iteration
  - 3. Gauss-Seidel iteration
  - 4. SOR iteration

Solving large sparse linear equations

§ 5.5 iterative method for linear equations

1. Introduction

Considering the linear equations

$$Ax = b$$

$$Ax = b \Leftrightarrow x = Bx + g \text{ (B is not unique)}$$

$$x^{(k+1)} = Bx^{k} + g$$

§ 5.5 iterative method for linear equations

1. Introduction

Considering the linear equations

$$Ax = b$$

$$Ax = b \Leftrightarrow x = Bx + g \text{ (B is not unique)}$$

$$(1)\begin{cases} \quad find\ the\ initial\ point\ x^0 \\ x^{(k+1)} = Bx^k + g, \qquad k = 0,1,\dots \end{cases}$$

If the sequence converges to x^*, $\lim_{k\to\infty} x^k = x^*$

then we have $x^* = Bx^* + g$

§ 5.5 iterative method for linear equations

1. Introduction

From the above, we can know that x^* is the solution of

$$Ax = b$$

when k is large enough that $x^k \approx x^*$

(1) is called the iteration method, and B is the iteration matrix.

Notice: B is not unique that affects the convergence

§ 5.5 iterative method for linear equations

- Convergence

The spectral radius of the matrix B , defined by

Spectral radius

$$\rho(B) = \max_{1 \le i \le n} |\lambda_i|$$

eigenvalue

Theorem： If the iteration method $x^{(k+1)} = Bx^k + f$ is convergence for any initial vector x^0 when $\rho(B) < 1$.

Notice: $\rho(B) \ge 1$ doesn't means non-convergence for any initial vector x^0

For iteration method $x^{(k+1)} = Bx^{(k)} + f \quad (k = 0,1,2,\cdots)$,

if $\|B\|_1 = \max_{1\le j\le n}\sum_{i=1}^{n}|b_{ij}| < 1$ or $\|B\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}|b_{ij}| < 1$

converges for any $x^{(0)}$

homework

Prove： for $\|B\|_\infty < 1$, substrate (1) with $x = Bx + f$, then

$$x_i^{(k+1)} - x_i = \sum_{j=1}^{n} b_{ij}(x_j^{(k)} - x_j)$$

$$|x_i^{(k+1)} - x_i| \le \sum_{j=1}^{n}|b_{ij}|\cdot|x_j^{(k)} - x_j| \le \max_{1\le i\le n}\sum_{j=1}^{n}|b_{ij}|\cdot|x_j^{(k)} - x_j|$$

set $\delta_k = \max_{1\le j\le n}|x_j^{(k)} - x_j|$, thus

$$|x_i^{(k+1)} - x_i| \le \|B\|_\infty \cdot \delta_k \qquad (i = 1,2,\cdots,n)$$

$$\delta_{k+1} \le \|B\|_\infty \delta^k \le \cdots \le \|B\|_\infty{}^{k+1}\delta_0 \to 0 \qquad (k \to \infty)$$

$\lim\limits_{k\to\infty}\max\limits_{1\le i\le n}\left|x_i^{(k+1)} - x_i\right| = 0$, $x_i^{(k+1)}$ converges $x_i (i = 1,2,\cdots,n)$

§ 5.5 iterative method for linear equations

- contents
  - 2. Jacobi iteration
  - 3. Gauss-Seidel iteration
  - 4. SOR iteration

§ 5.5 iterative method for linear equations

- 2. Jacobi iteration

suppose the equations Ax=b, set A=D-L-U

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

$$- \begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \ddots & \vdots \\ & & & \ddots & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

§ 5.5 iterative method for linear equations

- 2. Jacobi iteration

suppose the equations Ax=b, set A=D-L-U

$$(D-L-U)x=b,$$

$$Dx=(L+U)x+b,$$

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

set $B=D^{-1}(L+U), g=D^{-1}b,$

Jacobi iteration: $x^{(k+1)}=Bx^{(k)}+g$

§ 5.5 iterative method for linear equations

- 2. Jacobi iteration

suppose the equations Ax=b, set A=D-L-U

$$\text{set } B = D^{-1}(L+U), g = D^{-1}b,$$

$$\text{Jacobi iteration: } x^{(k+1)} = Bx^{(k)} + g$$

$$\rho(B) < 1 \Leftrightarrow \text{iteration convergence}$$

- 2. Component form of Jacobi iteration

for the equations $Ax = b$ $(a_{ii} \neq 0)$, thus

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

§ 5.5 iterative method for linear equations

- 2. Component form of Jacobi iteration

making the equivalent change

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}\left[b_1 - 0x_1 - a_{12}x_2 - \cdots - a_{1n}x_n\right] \\ x_2 = \dfrac{1}{a_{22}}\left[b_2 - a_{21}x_1 - 0x_2 - \cdots - a_{2n}x_n\right] \\ \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n = \dfrac{1}{a_{nn}}\left[b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - 0x_n\right] \end{cases}$$

§ 5.5 iterative method for linear equations

- 2. Component form of Jacobi iteration

Jacobi iteration:

$$\begin{cases} x_1^{(k+1)} = \dfrac{1}{a_{11}}[b_1 - 0x_1^{(k)} - a_{12}x_2^{(k)} - \cdots\cdots - a_{1n}x_n^{(k)}] \\ x_2^{(k+1)} = \dfrac{1}{a_{22}}[b_2 - a_{21}x_1^{(k)} - 0x_2^{(k)} - \cdots\cdots - a_{2n}x_n^{(k)}] \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(k+1)} = \dfrac{1}{a_{nn}}[b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots\cdots - 0x_n^{(k)}] \end{cases}$$

$$(k = 0,1,2,\cdots)$$

§ 5.5 iterative method for linear equations

- 2. Component form of Jacobi iteration

Jacobi iteration:

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

§ 5.5 iterative method for linear equations

- 3. Gauss-Seidel iteration

§ 5.5 iterative method for linear equations

- Component form of Jacobi iteration

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

§ 5.5 iterative method for linear equations

- Component form of Jacobi iteration

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

- Component form of Gauss-Seidel iteration

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

§ 5.5 iterative method for linear equations

- 3. Gauss-Seidel iteration

for the equations Ax=b, set A=D-L-U

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

$$
- \begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \ddots & \vdots \\ & & & \ddots & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}
$$

§ 5.5 iterative method for linear equations

- 3. Gauss-Seidel iteration

for the equations Ax=b, set A=D-L-U

$$(D-L-U)x=b,$$

$$(D-L)x=Ux+b,$$

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

set $B=(D-L)^{-1}U,\ g=(D-L)^{-1}b,$

Gauss-Seidel iteration: $x^{(k+1)}=Bx^{(k)}+g$

§ 5.5 iterative method for linear equations

- 3. Gauss-Seidel iteration

for $Ax = b$, set $A = D - L - U$,

$$B = (D-L)^{-1}U, g = (D-L)^{-1}b,$$

Gauss-Seidel iteration: $x^{(k+1)} = Bx^{(k)} + g$

$\rho(B) < 1 \Leftrightarrow$ iteration convergence

§ 5.5 iterative method for linear equations

- Component form of Gauss-Seidel iteration

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

§ 5.5 iterative method for linear equations

- 4. SOR (Successive Over-Relaxation) iteration

§ 5.5 iterative method for linear equations

- Component form of Gauss-Seidel iteration

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

$$i = 1,2,\cdots n,$$

$$k = 0,1,2,\cdots$$

§ 5.5 iterative method for linear equations

- Component form of Gauss-Seidel iteration

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

- SOR iteration

$$\mathrm{x}_i^{(k+1)} = (1-\omega)\mathrm{x}_i^{(k)} + \frac{\omega}{\mathrm{a}_{ii}}(\mathrm{b}_i - \sum_{j=1}^{i-1} \mathrm{a}_{ij} \mathrm{x}_j^{(k+1)} - \sum_{j=i+1}^{n} \mathrm{a}_{ij} \mathrm{x}_j^{(k)}),$$

§ 5.5 iterative method for linear equations

- 4. SOR iteration
  - Component form

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

ω : relaxation parameter

$\omega = 1$: Gauss-Seidel iteration

$0 < \omega < 1$ under-relaxation

$1 < \omega < 2$: over-relaxation

§ 5.5 iterative method for linear equations

- 4. SOR iteration

for the equations Ax=b, set A=D-L-U

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

$$- \begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \ddots & \vdots \\ & & & \ddots & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

§ 5.5 iterative method for linear equations

- 4. SOR iteration

$$(D-L-U)x=b,\quad Dx=(b+Lx+Ux),$$

$$x=D^{-1}(b+Lx+Ux),\quad \omega x=\omega D^{-1}(b+Lx+Ux),$$

$$x=(1-\omega)x+\omega D^{-1}(b+Lx+Ux),$$

$$Dx=D(1-\omega)x+\omega Lx+\omega Ux+\omega b,$$

$$(D-\omega L)x=\left((1-\omega)D+\omega U\right)x+\omega b,$$

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

§ 5.5 iterative method for linear equations

- 4. SOR iteration

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

set $B_\omega = (D - \omega L)^{-1}\big((1-\omega)D + \omega U\big), g_\omega = (D - \omega L)^{-1}\omega b,$

SOR iteration: $x^{(k+1)} = B_\omega x^{(k)} + g_\omega$

$\rho(B) < 1 \Leftrightarrow$ iteration convergence

§ 5.5 iterative method for linear equations

- convergence:

(1) necessary condition for the convergence of SOR iteration starting from any x^0 is :

$$0 < \omega < 2$$

(2) If the matrix A is symmetric positive definite, then SOR is convergence for any x^0 when $0 < \omega < 2$

(3) If the matrix A is strictly diagonally dominant, then SOR is convergence for any x^0 when $0 < \omega < 1$

§ 5.5 iterative method for linear equations

- 5. Example :
  - 1. solve the linear equation by Jacobi and Gauss-Seidel iteration

$$\begin{cases} 10x_1 + 3x_2 + x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases}$$

Exact solution $x^* = (1,1,1)^T$

§ 5.5 iterative method for linear equations

Solution :

using the Jacobi iteration

$$\begin{cases} x_1^{(k+1)} = -\frac{3}{10}x_2^{(k)} - \frac{1}{10}x_3^{(k)} + \frac{7}{5} \\ x_2^{(k+1)} = \frac{1}{5}x_1^{(k)} + \frac{3}{10}x_3^{(k)} + \frac{1}{2} \\ x_3^{(k+1)} = -\frac{1}{10}x_1^{(k)} - \frac{3}{10}x_2^{(k)} + \frac{7}{5} \end{cases}$$

start from the initial point $x^0 = (0,0,0)^T$, we have

$$x_1^{(1)} = 1.4, x_2^{(1)} = 0.5, x_3^{(1)} = 1.4$$

$$x_1^{(2)} = 1.11, x_2^{(2)} = 1.2, x_3^{(2)} = 1.11$$

Result:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|x^{(k)}-x^*\|_\infty$
0	0	0	0	1
1	1.4	0.5	1.4	0.5
2	1.11	1.20	1.11	0.2
3	0.929	1.055	0.929	0.071
4	0.9906	0.9645	0.9906	0.0355
5	1.01159	0.9953	1.01159	0.01159
6	1.000251	1.005795	1.000251	0.005795
7	0.9982364	1.0001255	0.9982364	0.0017636

It can be seen that the iterative sequence converges to the solution of the equations.

§ 5.5 iterative method for linear equations

Gauss-Seidel iteration:

$$\begin{cases} x_1^{(k+1)} = -\frac{3}{10} x_2^{(k)} - \frac{1}{10} x_3^{(k)} + \frac{7}{5} \\ x_2^{(k+1)} = \frac{1}{5} x_1^{(k+1)} + \frac{3}{10} x_3^{(k)} + \frac{1}{2} \\ x_3^{(k+1)} = -\frac{1}{10} x_1^{(k+1)} - \frac{3}{10} x_2^{(k+1)} + \frac{7}{5} \end{cases}$$

§ 5.5 iterative method for linear equations

start from $x^{(0)}=(0,0,0)^T$， and the result：

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|x^{(k)}-x^*\|_\infty$
0	0	0	0	1
1	1.4	0.78	1.026	0.4
2	1.0634	1.02048	0.987516	0.0634
3	0.9951044	0.99527568	1.00190686	0.0048956

It can be seen that the G-S iteration converges faster than Jacobi iteration. (3 times for G-S and 7 times for Jacobi to reach the same precision)

Example 2 :

solving the linear equations via SOR iteration

$$\begin{cases} 4x_1 - 2x_2 - 4x_3 = 10 \\ -2x_1 + 17x_2 + 10x_3 = 3 \\ -4x_1 + 10x_2 + 9x_3 = -7 \end{cases}$$

solving the linear equations via SOR iteration

$$\begin{cases} 4x_1 - 2x_2 - 4x_3 = 10 \\ -2x_1 + 17x_2 + 10x_3 = 3 \\ -4x_1 + 10x_2 + 9x_3 = -7 \end{cases}$$

exact solution is $x^*=(2,1,-1)^T$。

solution: SOR iteration

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + \dfrac{\omega}{4}(10 - 4x_1^{(k)} + 2x_2^{(k)} + 4x_3^{(k)}) \\ x_2^{(k+1)} = x_2^{(k)} + \dfrac{\omega}{17}(3 + 2x_1^{(k+1)} - 17x_2^{(k)} - 10x_3^{(k)}) \\ x_3^{(k+1)} = x_3^{(k)} + \dfrac{\omega}{9}(-7 + 4x_1^{(k+1)} - 10x_2^{(k+1)} - 9x_3^{(k)}) \end{cases}$$

start from $x^{(0)}=(0,0,0)^T$, $\omega=1.46$, and the result:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	3.65	0.8845882	-0.2021098
2	2.32166910	0.4230939	-0.22243214
3	2.5661399	0.6948261	-0.4952594
…	……	……	……
20	1.9999987	1.0000013	-1.0000034

It can be seen that the SOR method with $\omega=1.46$ iterates 20 times to reach the 5-th order precision. If taking $\omega=1$, it will cause 110 times iteration to reach the same precision.