迭代法初步

Jacobi迭代法

Seidel迭代法

迭代法的矩阵表示

迭代法的基本概念

设方程组$Ax=f$有唯一解x^*，将$Ax=f$变形为等价的方程组

$$x = Bx + f$$

由此建立迭代公式

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

给定初始向量 $x^{(0)}$，按此公式计算的近似解向量序列$\{x^{(k)}\}$，称此方法为迭代法

若 $\lim_{k\to\infty} x^{(k)} = x^*$，显然有$x^* = Bx^* + f$

则称迭代法是收敛的，否则称为发散的。迭代格式中的矩阵B称为迭代矩阵。

例4.1 $\begin{cases} 9x_1 - x_2 - x_3 = 7 \\ -x_1 + 10x_2 - x_3 = 8 \\ -x_1 - x_2 + 15x_3 = 13 \end{cases}$

特点：系数矩阵主对角元均不为零

$$\Leftrightarrow \begin{cases} x_1 = (7 + x_2 + x_3)/9 \\ x_2 = (8 + x_1 + x_3)/10 \\ x_3 = (13 + x_1 + x_2)/15 \end{cases}$$

取 $X^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

计算格式 $X^{(1)} = B\,X^{(0)} + f$

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & 1/9 & 1/9 \\ 1/10 & 0 & 1/10 \\ 1/15 & 1/15 & 0 \end{bmatrix} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} + \begin{bmatrix} 7/9 \\ 8/10 \\ 13/15 \end{bmatrix}$$

计算格式: $X^{(k+1)}=BX^{(k)}+f$

$X^{(0)}$	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$	$X^{(4)}$	……
0	0.7778	0.9630	0.9929	0.9987	
0	0.8000	0.9644	0.9935	0.9988	
0	0.8667	0.9778	0.9952	0.9991	

准确解 ➔ X^*

1.0000
1.0000
1.0000

雅可比迭代法

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \Longleftrightarrow \sum_{j=1}^{n} a_{ij}x_j = b_i$$

$$(i = 1,2,\ldots,n)$$

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

$$(i = 1,2,\ldots n;\ k=1,2,\ldots\ldots)$$

取初始向量$X^{(0)}=[x_1^{(0)}\ x_2^{(0)}\ \cdots\ x_n^{(0)}]^T$, 迭代计算

迭代法适用于解大型稀疏方程组

(万阶以上的方程组, 系数矩阵中零元素占很大比例, 而非零元按某种模式分布)

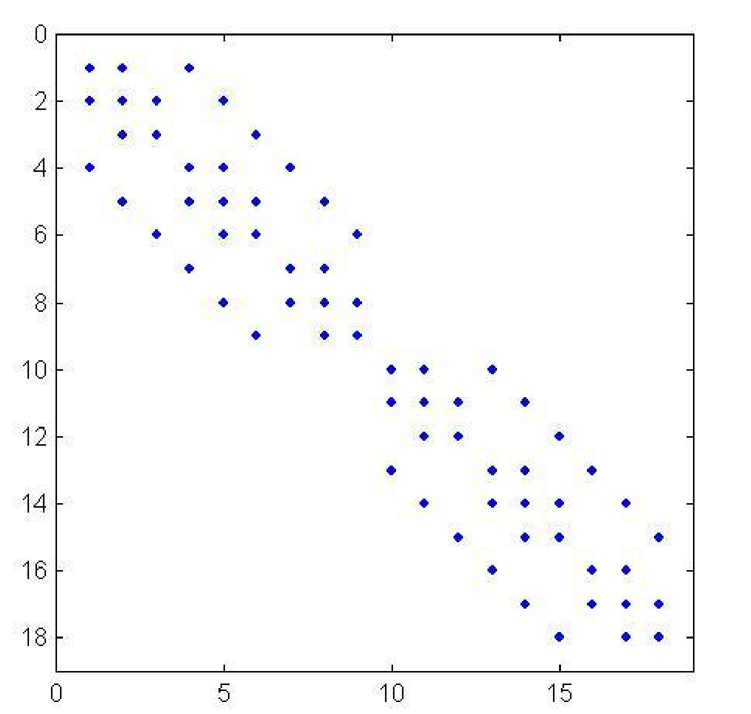
背景：电路分析、边值问题的数值解和数学物理方程

问题: (1)如何构造迭代格式?

(2)迭代格式是否收敛?

(3)收敛速度如何?

(4)如何进行误差估计?



高斯-赛德尔迭代法

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = 1,2,\ldots,n)$$

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

$$(i = 1,2,\ldots n;\ k = 1,2,\ldots\ldots)$$

取初始向量 $x^{(0)}=[x_1^{(0)}\ x_2^{(0)}\ \cdots\ x_n^{(0)}]^T$, 迭代计算

例 $\begin{cases} 9x_1 - x_2 - x_3 = 7 \\ -x_1 + 10x_2 - x_3 = 8 \\ -x_1 - x_2 + 15x_3 = 13 \end{cases} \Leftrightarrow \begin{cases} x_1 = (7 + x_2 + x_3)/9 \\ x_2 = (8 + x_1 + x_3)/10 \\ x_3 = (13 + x_1 + x_2)/15 \end{cases}$

$$x_1^{(k+1)} = (7 + x_2^{(k)} + x_3^{(k)})/9$$

$$x_2^{(k+1)} = (8 + x_1^{(k+1)} + x_3^{(k)})/10$$

$$x_3^{(k+1)} = (13 + x_1^{(k+1)} + x_2^{(k+1)})/15$$

$$\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1/10 & 1 & 0 \\ -1/15 & -1/15 & 1 \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} 0 & 1/9 & 1/9 \\ 0 & 0 & 1/10 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} + \begin{bmatrix} 7/9 \\ 8/10 \\ 13/15 \end{bmatrix}$$

雅可比迭代算法

$$\begin{cases} 9x_1 - x_2 - x_3 = 7 \\ -x_1 + 10x_2 - x_3 = 8 \\ -x_1 - x_2 + 15x_3 = 13 \end{cases}$$

0.7778	0.8000	0.8667
0.9630	0.9644	0.9719
0.9929	0.9935	0.9952
0.9987	0.9988	0.9991
0.9998	0.9998	0.9998
1.0000	1.0000	1.0000
1.0000	1.0000	1.0000

```
import numpy as np
A = np.array([[9, -1, -1], [-1, 10, -1],
[-1, -1, 15]])
B = np.array([7, 8, 13])
x0 = np.array([0.0, 0, 0])
x = np.array([0.0, 0, 0])
k = 0
while True:
    for i in range(3):
        temp = 0
        tempx = x0.copy()
        for j in range(3):
            if i != j:
                temp += x0[j] * A[i][j]
        x[i] = (B[i] - temp) / A[i][i]
    er = max(abs(x - x0))
    k += 1
    if er < 1e-4:
        break
    else:
        x0 = x.copy()
        print(k,x)
```

高斯-赛德尔迭代算法

$$\begin{cases} 9x_1 - x_2 - x_3 = 7 \\ -x_1 + 10x_2 - x_3 = 8 \\ -x_1 - x_2 + 15x_3 = 13 \end{cases}$$

0.7778	0.8778	0.9770
0.9839	0.9961	0.9987
0.9994	0.9998	0.9999
1.0000	1.0000	1.0000
1.0000	1.0000	1.0000

```
import numpy as np
A = np.array([[9, -1, -1], [-1, 10, -1],
[-1, -1, 15]])
B = np.array([7, 8, 13])
x0 = np.array([0.0, 0, 0])
x = np.array([0.0, 0, 0])
k = 0
while True:
    for i in range(3):
        temp = 0
        tempx = x0.copy()
        for j in range(3):
            if i != j:
                temp += x[j] * A[i][j]
        x[i] = (B[i] - temp) / A[i][i]
    er = max(abs(x - x0))
    k += 1
    if er < 1e-4:
        break
    else:
        x0 = x.copy()
        print(k,x)
```

雅可比 迭代法的矩阵表示

将方程组$AX=b$的系数矩阵 A 分裂

$$A = D - U - L$$

$$D=\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix} \quad L=-\begin{bmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix} \quad U=-\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & 0 & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

$$AX=b \quad => \quad DX^{(k+1)}=(U+L)X^{(k)}+b$$

$$X^{(k+1)}=D^{-1}(U+L)X^{(k)}+D^{-1}b$$

记$B_J=D^{-1}(U+L)$ $\qquad X^{(k+1)}=B_JX^{(k)}+f_J$

雅可比迭代矩阵

$$B_J = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}^{-1} \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & 0 & \cdots & -a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix}$$

$$B_J = \begin{bmatrix} 0 & -a_{12}/a_{11} & \cdots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & \cdots & -a_{2n}/a_{22} \\ \cdots & \cdots & \cdots & \cdots \\ -a_{n1}/a_{nn} & -a_{n2}/a_{nn} & \cdots & 0 \end{bmatrix} \qquad f_J = \begin{bmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \vdots \\ b_n/a_{nn} \end{bmatrix}$$

高斯-赛德尔迭代法的矩阵表示

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

$$\sum_{j=1}^{i} a_{ij}x_j^{(k+1)} = b_i - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \quad (i = 1,2,\ldots,n)$$

$$\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+2)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}$$

$$(D - L)X^{(\mathrm{k+1})} = b + UX^{(\mathrm{k})}$$

$$X^{(k+1)} = (D - L)^{-1}b + (D - L)^{-1}UX^{(k)}$$

记 $B_{G\text{-}S}=(D-L)^{-1}U$, $f_{G\text{-}S}=(D-L)^{-1}b$

高斯–赛德尔迭代格式: $X^{(k+1)}=B_{G\text{-}S}X^{(k)}+f_{G\text{-}S}$

$$B_{G-S}=\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}^{-1}\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

$$f_{G-S}=\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}^{-1}\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$