第2章 线性方程组-2

苏 芮
srhello@zju.edu.cn
开物苑4-202

数值方法的设计基本原则

可靠性高

- 易收敛：方法的可行性
- 高稳定：初始数据等产生的误差对结果的影响
- 低误差：运算结果不能产生太大的偏差且能够控制误差

复杂度低

- 便于编程实现：逻辑复杂度要小
- 计算量要小：运算复杂度要低，运行时间要短
- 存贮量要小：运算过程变量尽量少

常用数值方法的性能分析

方法一 克莱姆法则 （大学一年级线性代数）

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

$$\boxed{Ax = b}$$

方程组的解为 $x_i = \dfrac{\det(A_i)}{\det(A)}$

A_i 是 A 第 i 列的列向量被 $\boldsymbol{b}$ 取代后得到的矩阵

行列式

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}$$

2个未知量

$$x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{ed - bf}{ad - bc} \qquad y = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{af - ec}{ad - bc}$$

方程组的解

常用数值方法的性能分析

方法一 克莱姆法则 （大学一年级线性代数）

例： 求解一个 n 阶线性方程组，若使用**克莱姆法则**，需要计算 $n+1$ 个 n 阶行列式，在不计加减运算情况下，需要 $n!(n^2-1)$ 次乘除运算。

- 当 $n=20$ 时， $20!\times(20^2-1)\approx 9.7\times 10^{20}$

用每秒运算 30 亿次（主频3.0G）的计算机求解时，大约需要10000年的时间！！！

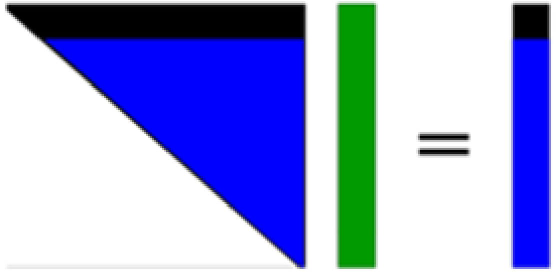
计算量要小，运算复杂度要低，运行时间要短！

常用数值方法的性能分析

方法二 高斯消元法 （上节课授课内容）

高斯消去法的主要思路：

将系数矩阵 A 化为上三角矩阵，然后回代求解



$$\rightarrow \begin{bmatrix} 1 & -2 & 2 & -2 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 61 & -61 \end{bmatrix} \rightarrow \begin{cases} x_3 = -1 \\ x_2 = 8 + 7x_3 = 1 \\ x_1 = -2 + 2x_2 - 2x_3 = 2 \end{cases}$$

高斯消元法的计算量为n^3量级，存储量为n^2量级，当n很大时，用高斯消元法直接求解，就会耗费大量的时间和存储单元。

计算量要小，运算复杂度要低，运行时间要短！

存贮量要小，运算过程变量尽量少！

迭代法解线性方程组

直接法（高斯消元法，LU分解）得到的解是理论上准确的，但它们的计算量都是n^3量级，存储量为n^2量级

当n很大时（n>400），用直接法时就会耗费大量的时间和存储单元。因此引入一类新的方法：**迭代法**

迭代法速度快。需要我们构造一个**等价的方程**，从而构造一个**收敛序列**，序列的极限值就是方程组的解（近似解）。

迭代法解线性方程组

介绍三种稳定的迭代方法

Jacobi （雅各比迭代）

它基本思想是单独求解某一个独立的变量；一次迭代仅仅对一个变量有作用，它便于理解但是收敛速度慢。

Gauss-Seidel （高斯-赛德尔）

每次都利用当前的最新信息，一般情况下， 如果Jacobi方法收敛，那么Gauss-Seidel方法会收敛更快。

SOR （逐次超松弛）

在Gauss-Seidel迭代法的基础上插入一个参数得到的，SOR可以比Guass-Seidel的收敛速度快一个量级。

迭代法解线性方程组

对方程组 $Ax=b$ 做等价变换 $x = Bx + f$

如：令 $A=M-N$，则

$$Ax=b \Rightarrow (M-N)x=b \Rightarrow Mx=b+Nx \Rightarrow x=M^{-1}Nx+M^{-1}b$$

则，我们可以构造序列 $x^{(k+1)} = Bx^{(k)} + f$ 　与初值的选取无关

定理 8 迭代过程 $X^{(k+1)}=BX^{(k)}+f$ 对任给初始向量 $X^{(0)}$ 收敛的充分必要条件是迭代矩阵的谱半径 $\rho(B)<1$；且当 $\rho(B)<1$ 时，迭代矩阵谱半径越小，收敛速度越快.

定义 3 矩阵 $A\in\mathbb{R}^{n\times n}$ 的所有特征值 $\lambda_i(i=1,2,\cdots,n)$ 的模的最大值称为矩阵 A 的谱半径，记作 $\rho(A)$，即

$$\rho(A)=\max_{1\leq i\leq n}|\lambda_i| \tag{3.53}$$

定理 4 矩阵 A 的谱半径不超过矩阵 A 的任何一种算子范数 $\|A\|_r$.

谱半径定义

定义 3 矩阵 $A \in \mathbb{R}^{n\times n}$ 的所有特征值 $\lambda_i(i=1,2,\cdots,n)$ 的模的最大值称为矩阵 A 的谱半径，记作 $\rho(A)$，即

$$\rho(A)=\max_{1\leqslant i\leqslant n}|\lambda_i| \tag{3.53}$$

定理 4 矩阵 A 的谱半径不超过矩阵 A 的任何一种算子范数 $\|A\|_r$.

证明 设 λ 为 A 的任一特征值，X 为对应于 λ 的 A 的特征向量，即

$$AX=\lambda X \qquad (X\neq 0)$$

则由范数的性质立即可得

$$|\lambda|\,\|X\|_r=\|\lambda X\|_r=\|AX\|_r\leqslant\|A\|_r\|X\|_r$$

因 X 是非零向量，故由上式可得

$$|\lambda|\leqslant\|A\|_r$$

这表明 A 的任一特征值的模都不超过 $\|A\|_r$，于是 $\rho(A)\leqslant\|A\|_r$. ∎

雅可比迭代法

Jacobi Iteration　雅可比迭代（形式1）

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \cdots \\ a_{n1}x_1 + \Lambda + a_{nn}x_n = b_n \end{cases}$$

$$\Rightarrow \begin{cases} x_1 = \dfrac{-1}{a_{11}}(a_{12}x_2 + \cdots + a_{1n}x_n - b_1) \\ x_2 = \dfrac{-1}{a_{22}}(a_{21}x_1 + a_{23}x_3 + \cdots + a_{1n}x_n - b_2) \\ \cdots \\ x_n = \dfrac{-1}{a_{nn}}(a_{n1}x_1 + \cdots + a_{n\,n-1}x_{n-1} - b_n) \end{cases}$$

雅可比迭代法

Jacobi Iteration 雅可比迭代（形式1）

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

```
x1={0,0,…..,0} , x2={1,1,…..,1} //赋初值
while( ||A*x2-b||>eps)  {
        x1=x2;
        for(i=0;i<n;i++)  {
                x2[i]=0;
                for(j=0;j<i;j++)  {
                        x2[i] += A[i][j]*x1[j]
                }
                for(j=i+1;j<n;j++)  {
                        x2[i] += A[i][j]*x1[j]
                }
                x2[i]=-(x2[i]-b[i])/A[i][i]
        }
}
```

雅可比迭代法

Jacobi Iteration　雅可比迭代（形式1 ）

$$\begin{cases} 3x_1 + x_2 = 5 \\ x_1 + 2x_2 = 5 \end{cases}$$

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

k	0	1	2	…	9	10	…
$x_1^{(k)}$	0	1.6667	0.8333	…	1.0005	0.9998	…
$x_2^{(k)}$	0	2.5	1.6667	…	2.0004	1.9997	…

Gauss-Seidel Iteration　　高斯—赛德尔迭代法

在Jacobi迭代中并没有使用最新的信息，新的分量值只有在整个扫描过程全部完成之后才能被利用（如何改进？）

$$
\begin{cases}
x_1^{(k+1)} = \dfrac{-1}{a_{11}}\left(a_{12}x_2^{(k)} + \Lambda + a_{1n}x_n^{(k)} - b_1\right) \\
x_2^{(k+1)} = \dfrac{-1}{a_{22}}\left(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + \Lambda + a_{1n}x_n^{(k)} - b_2\right) \\
\cdots \\
x_n^{(k+1)} = \dfrac{-1}{a_{nn}}\left(a_{n1}x_1^{(k+1)} + \Lambda + a_{n\,n-1}x_{n-1}^{(k+1)} - b_n\right)
\end{cases}
$$

高斯—赛德尔迭代法

Gauss-Seidel Iteration　　高斯—赛德尔迭代法

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

```
x2={1,1,…..,1} // 只有一个变量，直接在上面进行更新
while( ||A*x2-b||>eps)  {
        for(i=0;i<n;i++)  {
            for(j=0;j<i;j++)  {
                x2[i] += A[i][j]*x2[j]
            }
            for(j=i+1;j<n;j++)  {
                x2[i] += A[i][j]*x2[j]
            }
            x2[i]=-(x2[i]-b[i])/A[i][i]
        }
    }
```

Gauss-Seidel Iteration　　高斯—赛德尔迭代法

$$\begin{cases} 10x_1 - 2x_2 - x_3 = 3 \\ -2x_1 + 10x_2 - x_3 = 15 \\ -x_1 - 2x_2 + 5x_3 = 10 \end{cases}$$

$$\begin{cases} x_1^{(k+1)} = 0.2x_2^{(k)} + 0.1x_3^{(k)} + 0.3 \\ x_2^{(k+1)} = 0.2x_1^{(k+1)} + 0.1x_3^{(k)} + 1.5 \\ x_3^{(k+1)} = 0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} + 2 \end{cases}$$

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.3	0.8804	0.9843	0.9978	0.9997
$x_2^{(k)}$	0	1.56	1.9445	1.9923	1.9989	1.9999
$x_3^{(k)}$	0	2.684	2.9539	2.9938	2.9991	2.9999

矩阵分裂迭代法

矩阵分裂迭代法

基本思想

$A = M - N$

A 的一个矩阵分裂

$$Ax = b \iff Mx = Nx + b$$

M 可逆

$$\Updownarrow$$

$$x = M^{-1}Nx + M^{-1}b$$

不同的矩阵分裂产生不同的迭代公式和结果！

给定一个初始向量 $x^{(0)}$，可得 **迭代公式**

$$\boxed{x^{(k+1)} = Bx^{(k)} + f} \qquad k = 0, 1, 2, \ldots$$

其中 $B = M^{-1}N$ 称为 **迭代矩阵**

矩阵分裂迭代法

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

$$Ax = b$$

线性方程组

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}\right)/a_{11} \\ x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}\right)/a_{22} \\ \qquad\vdots \\ x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)}\right)/a_{nn} \end{cases}$$

迭代公式 $\boxed{x^{(k+1)} = Bx^{(k)} + f}$

将 A 分裂成 $A = D - (L+U)$ $\boxed{Dx = (L+U)x + b}$ $B = D^{-1}(L+U)$ $f = D^{-1}b$

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1,n} \\ 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

矩阵分裂迭代法

$$
\begin{cases}
x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}\right) \Big/ a_{11} \\
x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}\right) \Big/ a_{22} \\
\quad \vdots \\
x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)}\right) \Big/ a_{nn}
\end{cases}
$$

高斯

赛德尔

观察发现：迭代后的解更加趋向于真实解

雅各比迭代公式

如何改进雅各比迭代法?

$$
\begin{cases}
x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}\right) \Big/ a_{11}
\end{cases}
$$

迭代 **矩阵分裂：$A = (D - L) - U$**

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

高斯-塞德尔迭代公式

将 A 分裂成 $A = D - L - U$， 其中

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

Jacobi Iteration　雅可比迭代（形式2）

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

```
function y=jacobi(a, b, x0)
      D=diag(diag(a));
      U=-triu(a, 1);
      L= -tril(a, -1);
      B=D\(L+U);
      f=D\b;
      y=B*x0+f;n=1;
      while norm(y-x0) >=1.0e-6
          x0=y;
          y=B*x0+f;n=n+1;
      End

>> a=[10 -1 0;-1 10 -2;0 -2 10];
>> b=[9;7;6];
>> jacobi(a,b,[0;0;0])
```

矩阵分裂迭代法

$$X^{(k+1)} = B_G X^{(k)} + f_G \qquad (k=0,1,2,\cdots)$$

其中 $B_G = (D-L)^{-1}U$（称为高斯—赛德尔迭代矩阵），$f_G = (D-L)^{-1}b$.

```
function y=seidel(a, b, x0)
D=diag(diag(a));
U=-triu(a, 1);
L= -tril(a, -1);
G=(D-L)\U;
f=(D-L)\b;
y=G*x0+f; n=1;
while norm(y-x0) >=1.0e-6
    x0=y;
    y=G*x0+f;
    n=n+1;
End
```

```
>> a=[10 -1 0;-1 10 -2;0 -2 10];
>> b=[9;7;6];
>> seidel(a,b,[0;0;0])
```

矩阵分裂迭代法

分别用雅各比和高斯-塞德尔迭代法求解

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

取初始向量 $x^{(0)} = [0, 0, 0]^T$，计算过程中小数点后保留 4 位。

$$x^* = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$$

精确解

雅各比迭代

$$\begin{cases} x_1^{(k+1)} = \left(1 + x_2^{(k)}\right)/2 \\ x_2^{(k+1)} = \left(8 + x_1^{(k)} + x_3^{(k)}\right)/3 \\ x_3^{(k+1)} = \left(-5 + x_2^{(k)}\right)/2 \end{cases}$$

计算可得：

$$x^{(1)} = [\ 0.5000, 2.6667, -2.5000\ ]^T$$

$$\vdots$$

$$x^{(21)} = [\ 2.0000, 3.0000, -1.0000\ ]^T$$

共迭代21步

矩阵分裂迭代法

分别用雅各比和高斯-塞德尔迭代法求解

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

取初始向量 $x^{(0)}=[0,\ 0,\ 0]^T$，计算过程中小数点后保留 4 位。

$$x^* = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$$

精确解

高斯-塞德尔迭代

$$\begin{cases} x_1^{(k+1)} = \left(1 + x_2^{(k)}\right)/2 \\ x_2^{(k+1)} = \left(8 + x_1^{(k+1)} + x_3^{(k)}\right)/3 \\ x_3^{(k+1)} = \left(-5 + x_2^{(k+1)}\right)/2 \end{cases}$$

计算可得：$x^{(1)} = [\ 0.5000,\ 2.8333,\ -1.0833\ ]^T$

$$\vdots$$

$x^{(9)} = [\ 2.0000,\ 3.0000,\ -1.0000\ ]^T$

不同的矩阵分裂产生不同的迭代公式和结果！

共迭代9步

迭代收敛性判断定理

定理 5　若迭代过程 $X^{(k+1)}=BX^{(k)}+f$ 中迭代矩阵 B 的某种算子范数 $\|B\|_r=q<1$，则

(1)对任意初始向量 $X^{(0)}$，该迭代过程均收敛于方程 $X=BX+f$ 的唯一解 X^*；

(2) $$\|X^*-X^{(k)}\|_r\leqslant\frac{1}{1-q}\|X^{(k+1)}-X^{(k)}\|_r \tag{3.54}$$

(3) $$\|X^*-X^{(k)}\|_r\leqslant\frac{q^k}{1-q}\|X^{(1)}-X^{(0)}\|_r \tag{3.55}$$

定理 6　若方程组 $AX=b$ 的系数矩阵 $A=[a_{ij}]_{n\times n}$ 按行严格对角占优或按列严格对角占优，即满足条件

$$|a_{ii}|>\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|\qquad(i=1,2,\cdots,n) \tag{3.57}$$

或

$$|a_{jj}|>\sum_{\substack{i=1\\i\neq j}}^{n}|a_{ij}|\qquad(j=1,2,\cdots,n) \tag{3.58}$$

则方程组 $AX=b$ 有唯一解，且对任意初始向量 $X^{(0)}$，雅可比迭代法与高斯—赛德尔迭代法都收敛.

定理 7　若方程组 $AX=b$ 的系数矩阵为对称正定矩阵，则对任意初始向量 $X^{(0)}$，高斯—赛德尔迭代法收敛.

A阵为正定矩阵的充分必要条件是A阵的顺序主子式大于零。

松弛迭代法

SOR

记 $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$

则 $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$

可以看作在前一步上加一个修正量。若在修正量前乘以一个因子

$$x^{(k+1)} = x^{(k)} + \omega \Delta x^{(k)}$$

$$x^{(k+1)} = x^{(k)} + \omega(x^{(k+1)} - x^{(k)})$$

对Gauss - Seidel迭代格式 $x^{(k+1)} = D^{-1}(Lx^{(k+1)} + Ux^{(k)} + b)$

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

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

那么可以期望适当地选择因子 ω 的值，就可能使迭代过程收敛更快．这种迭代法称为**超松弛迭代法**，简称 SOR 方法，其中 ω 称为**松弛因子**．

松弛迭代法

例 11 用 SOR 法求解线性方程组

$$\begin{bmatrix} 4 & -2 & -4 \\ -2 & 17 & 10 \\ -4 & 10 & 9 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \\ -7 \end{bmatrix}$$

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + \dfrac{\omega}{4}(10 - 4x_1^{(k)} + 2x_2^{(k)} + 4x_3^{(k)}) \\ x_2^{(k+1)} = x_2^{(k)} + \dfrac{\omega}{17}(8 + 2x_1^{(k+1)} - 17x_2^{(k)} - 10x_3^{(k)}) \qquad (k = 0, 1, 2, \cdots) \\ x_3^{(k+1)} = x_3^{(k)} + \dfrac{\omega}{9}(-7 + 4x_1^{(k+1)} - 10x_2^{(k+1)} - 9x_3^{(k)}) \end{cases}$$

取 $\omega = 1.46$, $X^{(0)} = (0,0,0)^T$，计算结果见表 3－5.

表 3－5

k	0	1	2	3	…	20
$x_1^{(k)}$	0	3.65	2.321669	2.566140	…	1.999998
$x_2^{(k)}$	0.	0.8845883	0.4230939	0.6948260	…	1.000001
$x_3^{(k)}$	0	－0.2021098	－0.2224321	－0.4952594	…	－1.000003

超松弛迭代法收敛性判断

定理 9 若方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}=[a_{ij}]_{n\times n}$ 主对角线上元素 $a_{ii}\neq 0(i=1,2,\cdots,n)$，则用 SOR 法求解 $\boldsymbol{AX}=\boldsymbol{b}$，收敛的必要条件是 $0<\omega<2$.

定理 10 若方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的系数矩阵是对称正定矩阵，且 $0<\omega<2$，则对任意初始向量 $\boldsymbol{X}^{(0)}$，SOR 迭代法收敛.

迭代法收敛判断小结

收敛条件		雅可比迭代法	高斯—赛德尔迭代法	超松弛迭代法
$\rho(B)<1$	$\Leftrightarrow$	收敛	收敛	
$\lVert B\rVert_r<1$	$\Rightarrow$	收敛	收敛	
A阵为对角占优矩阵	$\Rightarrow$	收敛	收敛	
A阵为对称正定矩阵	$\Rightarrow$		收敛	当 $0<\omega<2$ 时收敛

A阵为正定矩阵的充分必要条件是A阵的顺序主子式大于零。

思考题

$$\begin{bmatrix} 4 & -2 & -4 \\ -2 & 17 & 10 \\ -4 & 10 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \\ -7 \end{bmatrix}$$

分别用高斯—赛德尔迭代法（函数已提供）和超松弛迭代法（函数自己编写）求解，并调节超松弛法的松弛因子，比较两种方法的迭代效率。

感谢聆听,欢迎讨论！