§2.4 解线性方程组的迭代法

/* Iterative Techniques for Solving Linear Systems */

求解 $A\vec{x}=\vec{b}$

思路　将 $A\vec{x}=\vec{b}$ 等价改写为 $\vec{x}=B\vec{x}+\vec{f}$ 形式，建立迭代 $\vec{x}^{(k+1)}=B\vec{x}^{(k)}+\vec{f}$。从初值 $\vec{x}^{(0)}$ 出发，得到序列 $\{\vec{x}^{(k)}\}$。其中，称 B 为迭代矩阵。

计算精度可控，特别适用于求解系数为大型稀疏矩阵 /* sparse matrices */ 的方程组。

研究内容:

(1) 如何建立迭代格式?　　(3) 收敛速度?

(2) 向量序列的收敛条件?　　(4) 误差估计?

Jacobi 法和 Gauss - Seidel 法

/* Jacobi & Gauss-Seidel Iterative Methods */

1. Jacobi Iterative Method

例：求解方程组

$$\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases}$$

精确解为$(1.1, 1.2, 1.3)^T$

解：分别从上面的三个方程中分离出 x_1, x_2, x_3

$$\begin{aligned} x_1 &= 0.1x_2 + 0.2x_3 + 0.72 \\ x_2 &= 0.1x_1 + 0.2x_3 + 0.83 \\ x_3 &= 0.2x_1 + 0.2x_2 + 0.84 \end{aligned}$$

据此可建立迭代公式：

$$\begin{cases} x_1^{(k+1)} = 0.1x_2^{(k)} + 0.2x_3^{(k)} + 0.72 \\ x_2^{(k+1)} = 0.1x_1^{(k)} + 0.2x_3^{(k)} + 0.83 \\ x_3^{(k+1)} = 0.2x_1^{(k)} + 0.2x_2^{(k)} + 0.84 \end{cases}$$

$$\begin{cases} x_1^{(k+1)} = 0.1x_2^{(k)} + 0.2x_3^{(k)} + 0.72 \\ x_2^{(k+1)} = 0.1x_1^{(k)} + 0.2x_3^{(k)} + 0.83 \\ x_3^{(k+1)} = 0.2x_1^{(k)} + 0.2x_2^{(k)} + 0.84 \end{cases}$$

设取迭代初值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ 下表记录了迭代结果，当迭代次数 k 增大时，迭代值 $x_1^{(k)}, x_2^{(k)}, x_3^{(k)}$ 会越来越逼近方程组的精确解 $x_1^* = 1.1, x_2^* = 1.2, x_3^* = 1.3$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.00000	0.00000	0.00000
1	0.72000	0.83000	0.84000
2	0.97100	1.07000	1.15000
3	1.05700	1.15710	1.24820
4	1.08535	1.18534	1.28282
5	1.09510	1.19510	1.29414
6	1.09834	1.19834	1.29504
7	1.09944	1.19981	1.29934
8	1.09981	1.19941	1.29978
9	1.09994	1.19994	1.29992

Jacobi迭代分量形式:

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

$$\Big\Downarrow\ a_{ii} \neq 0$$

$$
\begin{cases}
x_1 = \dfrac{1}{a_{11}}\left(-a_{12}x_2 - \ldots - a_{1n}x_n + b_1\right) \\
x_2 = \dfrac{1}{a_{22}}\left(-a_{21}x_1 - \ldots - a_{2n}x_n + b_2\right) \\
\cdots \quad \cdots \quad \cdots \quad \cdots \\
x_n = \dfrac{1}{a_{nn}}\left(-a_{n1}x_1 - \ldots - a_{nn-1}x_{n-1} + b_n\right)
\end{cases}
$$

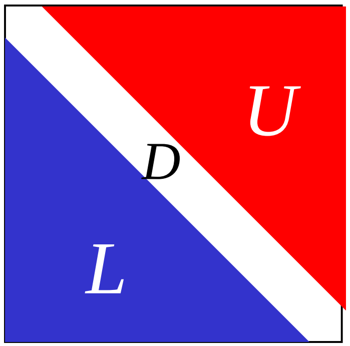
$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)}\right)$$

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

Jacobi迭代的矩阵形式

写成矩阵形式：

$A =$



$$A\vec{x} = \vec{b} \Leftrightarrow (D+L+U)\vec{x} = \vec{b}$$

$$\Leftrightarrow D\vec{x} = -(L+U)\vec{x} + \vec{b}$$

$$\Leftrightarrow \vec{x} = \underbrace{-D^{-1}(L+U)}_{B}\vec{x} + \underbrace{D^{-1}\vec{b}}_{\vec{f}}$$

Jacobi 迭代阵

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

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

Algorithm: Jacobi Iterative Method

Solve $A\vec{x} = \vec{b}$ given an initial approximation $\vec{x}^{(0)}$.

Input: **the number of equations and unknowns *n*; the matrix entries *a*[ ][ ]; the entries *b*[ ]; the initial approximation *X*0[ ]; tolerance *TOL*; maximum number of iterations N_{max}.**

Output: **approximate solution *X*[ ] or a message of failure.**

Step 1 **Set $k = 1$;**

Step 2 **While ($k \le N_{max}$) do steps 3-6**

Step 3 **For $i = 1, \ldots, n$**

Set $X_i = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n}(a_{ij}X0_j)\right)$; **/* compute x_k */**

Step 4 **If** $\|X - X0\|_\infty = \max_{1\le i\le n}|X_i - X0_i| < TOL$ **then Output (*X*[ ]); STOP; /* successful */**

Step 5 **For $i = 1, \ldots, n$ Set *X*0[ ] = *X* [ ]; /* update *X*0 */**

Step 6 **Set k ++;**

Step 7 **Output (Maximum number of iterations exceeded); STOP. /* unsuccessful */**

```
/*
  求解线性方程组的Jacobi迭代法
    AX=b
*/
void Jacobi_Method()
{
    double u[NN],sum1=0;
    for(int k=1;k<=MAXM;k++)
    {
        for(int i=0;i<NN;i++)
        {
            for(int j=0;j<NN;j++)
            {
                if(i!=j)
                    sum1+=A[i][j]*x0[j];
            }
            u[i]=(B[i]-sum1)/A[i][i];
            sum1=0;
        }
        cout<<"  "<<k<<"        ";

        for(i=0;i<NN;i++)
        {
            x0[i]=u[i];
            cout<<setprecision(7)<<setw(8)<<x0[i]<<"   ";

        }
        getchar();
        cout<<endl;
    }
}
```

$$X_i = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j\neq i\\ j=1}}^{n}(a_{ij}X0_j)\right)$$

$$||X - X0||_\infty = \max_{1\le i\le n}|X_i - X0_i| \ < TOL$$

➢ 2. Gauss - Seidel Iterative Method

Jacobi迭代的计算一般按 $x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_n^{(k+1)}$ 的次序进行，注意到计算 $x_i^{(k+1)}$ 时，$x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$ 已经算好，而Jacobi迭代法并不利用这些最新的近似值计算，

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

这启发我们对 Jacobi 迭代进行修改。

例　对例子，作修正得

$$\begin{cases} x_1^{(k+1)} = 0.1x_2^{(k)} + 0.2x_3^{(k)} + 0.72 \\ x_2^{(k+1)} = 0.1x_1^{(k+1)} + 0.2x_3^{(k)} + 0.83 \\ x_3^{(k+1)} = 0.2x_1^{(k+1)} + 0.2x_2^{(k+1)} + 0.84 \end{cases}$$

仍取初值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$，按上式的计算结果见下表，与前面的表的计算结果比较，本公式的效果明显比前面的要好。

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.00000	0.00000	0.00000
1	0.72000	0.90200	1.16400
2	1.04308	1.16719	1.28205
3	1.09313	1.19947	1.29972
4	1.09913	1.19947	1.29972
5	1.09989	1.19993	1.29996
6	1.09999	1.19999	1.30000

2. Gauss - Seidel Iterative Method

Gauss-Seidel迭代分量形式

$$x_1^{(k+1)} = \frac{1}{a_{11}}(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} - \cdots - a_{1n}x_n^{(k)} + b_1)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}(\boxed{-a_{21}x_1^{(k+1)}} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)} - \cdots - a_{2n}x_n^{(k)} + b_2)$$

$$x_3^{(k+1)} = \frac{1}{a_{33}}(\boxed{-a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}} - a_{34}x_4^{(k)} - \cdots - a_{3n}x_n^{(k)} + b_3)$$

$$\cdots \quad \cdots \quad \cdots \quad \cdots$$

$$x_n^{(k+1)} = \frac{1}{a_{nn}}(\boxed{-a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - a_{n3}x_3^{(k+1)} - \cdots - a_{nn-1}x_{n-1}^{(k+1)}} + b_n)$$

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

充分利用新值建立起来的公式称作高斯-塞德尔（Gauss-Seidel）公式

Gauss - Seidel Iterative Method

```
for(k=1;k<=MAXM;k++)
{
    for(i=0;i<NN;i++)
    {
        for(j=0;j<NN;j++)
        {
            if(j!=i)
                sum1+=A[i][j]*x0[j];
        }
        u[i]=(B[i]-sum1)/A[i][i];
        x0[i]=u[i];      //替换过程
        sum1=0;
    }
    cout<<"  "<<k<<"        ";
    for(i=0;i<NN;i++)
    {
        x0[i]=u[i];
        cout<<setprecision(7)<<setw(8)<<x0[i]<<"   ";
    }
}
```

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

Guass-Seidel迭代的矩阵形式

写成矩阵形式：
$$\vec{x}^{(k+1)} = -D^{-1}(L\vec{x}^{(k+1)} + U\vec{x}^{(k)}) + D^{-1}\vec{b}$$
$$\Leftrightarrow (D+L)\vec{x}^{(k+1)} = -U\vec{x}^{(k)} + \vec{b}$$
$$\Leftrightarrow \vec{x}^{(k+1)} = \underbrace{-(D+L)^{-1}U}_{B}\vec{x}^{(k)} + \underbrace{(D+L)^{-1}\vec{b}}_{\vec{f}}$$

Gauss-Seidel
迭代阵

注：二种方法都存在收敛性问题。

有例子表明：Gauss-Seidel法收敛时，Jacobi法可能不收敛；而Jacobi法收敛时， Gauss-Seidel法也可能不收敛。

例： 设方程组为

$$\begin{cases} 5x_1 + 2x_2 + x_3 = -12 \\ -x_1 + 4x_2 + 2x_3 = 20 \\ 2x_1 - 3x_2 + 10x_3 = 3 \end{cases}$$

试分别写出其 Jacobi 和 Gauss-Seidel 的迭代格式以及相应的迭代矩阵。

解： Jacobi 迭代格式为

$$\begin{cases} \boldsymbol{x}_1^{(k+1)} = \frac{1}{5}(-12 - 2\boldsymbol{x}_2^{(k)} - \boldsymbol{x}_3^{(k)}) = -\frac{2}{5}\boldsymbol{x}_2^{(k)} - \frac{1}{5}\boldsymbol{x}_3^{(k)} - \frac{12}{5} \\ \boldsymbol{x}_2^{(k+1)} = \frac{1}{4}(20 + \boldsymbol{x}_1^{(k)} - 2\boldsymbol{x}_3^{(k)} = \frac{1}{4}\boldsymbol{x}_1^{(k)} - \frac{1}{2}\boldsymbol{x}_3^{(k)} + 5 \\ \boldsymbol{x}_3^{(k+1)} = \frac{1}{10}(3 - 2\boldsymbol{x}_1^{(k)} + 3\boldsymbol{x}_2^{(k)}) = -\frac{1}{5}\boldsymbol{x}_1^{(k)} + \frac{3}{10}\boldsymbol{x}_2^{(k)} + \frac{3}{10} \end{cases}$$

故 Jacobi 迭代矩阵为

$$\mathbf{B_J}=\begin{bmatrix} 0 & -\frac{2}{5} & -\frac{1}{5} \\ \frac{1}{4} & 0 & -\frac{1}{2} \\ -\frac{1}{5} & \frac{3}{10} & 0 \end{bmatrix}$$

Seidel迭代格式为

$$\begin{cases} x_1^{(k+1)} = -\frac{2}{5}x_2^{(k)} - \frac{1}{5}x_3^{(k)} - \frac{12}{5} \\ x_2^{(k+1)} = \frac{1}{4}x_1^{(k+1)} - \frac{1}{2}x_3^{(k)} + 5 \\ x_3^{(k+1)} = -\frac{1}{5}x_1^{(k+1)} + \frac{3}{10}x_2^{(k+1)} + \frac{3}{10} \end{cases}$$

解出 $x_i^{(k+1)}, i=1,2,3$

$$\begin{cases} x_1^{(k+1)} = -\frac{2}{5}x_2^{(k)} - \frac{1}{5}x_3^{(k)} - \frac{12}{5} \\ x_2^{(k+1)} = -\frac{1}{10}x_2^{(k)} - \frac{11}{20}x_3^{(k)} + \frac{22}{5} \\ x_3^{(k+1)} = \frac{1}{20}x_2^{(k)} - \frac{1}{8}x_3^{(k)} + \frac{21}{10} \end{cases}$$

故可得Seidel迭代矩阵为

$$B_s = \begin{bmatrix} 0 & -\frac{2}{5} & -\frac{1}{5} \\ 0 & -\frac{1}{10} & -\frac{11}{20} \\ 0 & \frac{1}{20} & -\frac{1}{8} \end{bmatrix} \quad \blacksquare$$

§ 2.4 迭代法的收敛性 /* Convergence of Iterative methods */

$\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{f}$ 的收敛条件

$$\underline{\vec{e}^{(k+1)}} = \vec{x}^{(k+1)} - \vec{x}^* = (B\vec{x}^{(k)} + \vec{f}) - (B\vec{x}^* + \vec{f}) = B(\vec{x}^{(k)} - \vec{x}^*) = \underline{B\vec{e}^{(k)}}$$

$$\Rightarrow \boxed{\vec{e}^{(k)} = B^k \vec{e}^{(0)}}$$

定理 设 $\vec{x} = B\vec{x} + \vec{f}$ 存在唯一解，则从任意 $\vec{x}^{(0)}$ 出发，迭代 $\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{f}$ 收敛 $\Leftrightarrow$ $B^k \to 0$

定理 $B^k \to 0 \Leftrightarrow \rho(B) < 1$

定理 （迭代法基本定理）设有方程组 $\vec{x} = B\vec{x} + f$ 对于任意初始向量 $\vec{x}^{(0)}$及任意 f，解此方程组的迭代法(即 $\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{f}$ ）收敛的充要条件是 $\rho(B) < 1$。

证明: 注意到

$$\vec{e}^{(k+1)} = \vec{x}^{(k+1)} - \vec{x}^* = B(\vec{x}^{(k)} - \vec{x}^*) = B\vec{e}^{(k)} = B^{k+1}\vec{e}^{(0)}$$

则

$$\left\|\vec{e}^{k}\right\| = \left\|B^{k}\vec{e}^{(0)}\right\| \le \left\|B^{k}\right\|\left\|\vec{e}^{(0)}\right\| \le \|B\|^{k}\left\|\vec{e}^{(0)}\right\|,$$

$$\left\|\vec{e}^{k}\right\| \le (\rho(B))^{k}\left\|\vec{e}^{(0)}\right\| \to 0 \quad \blacksquare$$

例：已知方程组 $\begin{bmatrix} 1 & -2 \\ -0.32 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$，考察解此方程组的$Jacobi$迭代法是否收敛。

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

解：方程组的 $Jacobi$ 迭代阵 $B = \begin{bmatrix} 0 & 2 \\ 0.32 & 0 \end{bmatrix}$

$$|\lambda I - B| = \begin{vmatrix} \lambda & -2 \\ -0.32 & \lambda \end{vmatrix} = 0 \Longrightarrow \lambda_1 = -0.8 \quad \lambda_2 = 0.8$$

因 $\rho(B) = 0.8$，故方程组的 $Jacobi$ 迭代收敛。■

定理 （充分条件）若存在一个矩阵范数使得 $\| B \| = q < 1$，则迭代收敛，且有下列误差估计：

① $\| \vec{x}^* - \vec{x}^{(k)} \| \le \dfrac{q}{1-q} \| \vec{x}^{(k)} - \vec{x}^{(k-1)} \|$

② $\| \vec{x}^* - \vec{x}^{(k)} \| \le \dfrac{q^k}{1-q} \| \vec{x}^{(1)} - \vec{x}^{(0)} \|$

证明： 因$\rho(B) \le \|B\| < 1$，故收敛性得证。

① $$\begin{aligned} \vec{x}^* - \vec{x}^{(k)} &= B(\vec{x}^* - \vec{x}^{(k-1)}) \\ &= B(\vec{x}^* - \vec{x}^{(k)} + \vec{x}^{(k)} - \vec{x}^{(k-1)}) \end{aligned}$$

$$\Rightarrow \| \vec{x}^* - \vec{x}^{(k)} \| \le q(\| \vec{x}^* - \vec{x}^{(k)} \| + \| \vec{x}^{(k)} - \vec{x}^{(k-1)} \|) \quad \checkmark$$

② $$\vec{x}^{(k)} - \vec{x}^{(k-1)} = B(\vec{x}^{(k-1)} - \vec{x}^{(k-2)}) = \dots = B^{k-1}(\vec{x}^{(1)} - \vec{x}^{(0)})$$

$$\Rightarrow \| \vec{x}^{(k)} - \vec{x}^{(k-1)} \| \le q^{k-1} \| \vec{x}^{(1)} - \vec{x}^{(0)} \| \quad \blacksquare$$

例 考察用Jacobi方法解方程组

$$\begin{cases} 8x_1 - 3x_2 + 2x_3 = 20 \\ 4x_1 + 11x_2 - x_3 = 33 \\ 6x_1 + 3x_2 + 12x_3 = 36 \end{cases}$$

的收敛性。

解 Jacobi迭代矩阵为

$$B_J = -\begin{bmatrix} 0 & -\frac{3}{8} & \frac{2}{8} \\ \frac{4}{11} & 0 & -\frac{1}{11} \\ \frac{6}{12} & \frac{3}{12} & 0 \end{bmatrix}$$

因 $\|B_J\|_\infty = \max\{\frac{5}{8}, \frac{5}{11}, \frac{9}{12}\} < 1$，故其 Jacobi 迭代收敛。■

例 设有方程组 $X = BX + f$

$$\text{其中} B = \begin{pmatrix} 0.9 & 0 \\ 0.3 & 0.8 \end{pmatrix},\ f = \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

讨论用迭代法求解 $X^{(k+1)} = BX^{(k)} + f$ 的收敛性。

解

$$\|B\|_1 = 1.2,\ \|B\|_2 = 1.021,\ \|B\|_\infty = 1.1$$

但迭代阵的谱半径 $\rho(B) = 0.9$，故其迭代法收敛。∎

m文件:

```
a=[0.9,0;0.3,0.8]
n1=norm(a,1)
n2=norm(a,2)
n8=norm(a,inf)
eig(a)
```

关于解某些特殊方程组迭代法的收敛性

定义：设 $A=(a_{ij})_{n\times n}$

(1) 如果 A元素满足

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

称A为严格对角占优阵。

(2) 如果 A元素满足

$$|a_{ii}| \geq \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| \quad (i=1,2,\cdots,n)$$

且上式至少有一个不等式严格成立，

称A为弱对角占优阵。

定理：（充分条件）若A 为严格对角占优阵 /* strictly diagonally dominant matrix */ , 则解$A\vec{x}=\vec{b}$的Jacobi 和 Gauss - Seidel 迭代均收敛。

证明：首先证明一个引理 /* Lemma */

引理：若A 为对角严格占优，则$\det(A)\neq 0$，且所有的 $a_{ii}\neq 0$。

证明：若不然，即 $\det(A)=0$，则 A 是奇异阵。

存在非零向量 $\vec{x}_0=(x_1,x_2,\cdots x_n)^T$使得$A\vec{x}_0=\vec{0}$.

记 $|x_m|=\max\limits_{1\le i\le n}|x_i|$ $\longrightarrow$ $\sum\limits_{i=1}^{n}a_{mi}\,x_i=0$

$$\longrightarrow\ |a_{mm}x_m|=\left|\sum_{i\neq m}a_{mi}\,x_i\right|\le|x_m|\sum_{i\neq m}|a_{mi}|$$

于是与A 为对角严格占优矛盾。得证。

定理：（充分条件）若A为严格对角占优阵 /* strictly diagonally dominant matrix */ 则解$\boldsymbol{A}\vec{\boldsymbol{x}}=\vec{\boldsymbol{b}}$的Jacobi 和 Gauss - Seidel 迭代均收敛。

证明：我们需要对 Jacobi 迭代和 Gauss-Seidel迭代分别证明：任何一个$|\lambda|\geq 1$都不可能是对应迭代阵的特征根，即$|\lambda I-B|\neq 0$。

Jacobi: $B_J=-D^{-1}(L+U)$

$$|\boldsymbol{\lambda I}-\boldsymbol{B}|=|\boldsymbol{\lambda I}+\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})|$$

$$=|\boldsymbol{D}^{-1}(\boldsymbol{\lambda D}+\boldsymbol{L}+\boldsymbol{U})|=|\boldsymbol{D}^{-1}||\boldsymbol{\lambda D}+\boldsymbol{L}+\boldsymbol{U}|$$

如果$|\lambda|\geq 1$则$|\lambda a_{ii}|\geq|a_{ii}|>\sum_{j\neq i}|a_{ij}|$

$\Longrightarrow |\lambda I-B|\neq 0$ ■

关于 Gauss-Seidel 迭代的证明与此类似(略)。

例：对下列方程组建立收敛的迭代公式：

$$\begin{cases} 11x_1 - 3x_2 - 33x_3 = 1 \\ -22x_1 + 11x_2 + x_3 = 0 \\ x_1 - 4x_2 + 2x_3 = 1 \end{cases}$$

解： 上述方程组的系数矩阵不是严格对角占优的，但经行交换后

$$\begin{cases} -22x_1 + 11x_2 + x_3 = 0 \\ x_1 - 4x_2 + 2x_3 = 1 \\ 11x_1 - 3x_2 - 33x_3 = 1 \end{cases}$$

变为对角占优，对此线性方程组建立的 $Jacobi$ 迭代法或 $Gauss - seidel$ 迭代法是收敛的。∎