第三章 迭 代 法

3.4 解线性方程组的迭代法

求解线性方程组的迭代法

直接法的缺点：

- ✓运算量大，不适合大规模的线性方程组求解;
- ✓对于一些大型稀疏矩阵，浪费内存严重。

迭代法：从一个初始向量出发，按照一定的迭代格式，构造出一个趋向于真解的无穷序列。

- ✓只需存储系数矩阵中的非零元素;
- ✓运算量不超过 $O(kn^2)$，其中 k 为迭代步数.

迭代解法是目前求解大规模线性方程组的主要方法。

解线性方程组迭代法的基本思想

迭代格式的建立：

$$Ax = b \xleftrightarrow[A=M-N]{} Mx = Nx + b$$

$$\Updownarrow$$

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

给定一个初始向量 $x^{(0)}$，令$G = M^{-1}N$，$f = M^{-1}b$,可得 $x = Gx + f$,其迭代格式为$x^{(k+1)} = Gx^{(k)} + f$。其中 G 称为迭代矩阵。$x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \cdots, x_n^{(k)})^T, k = 0, 1, 2, \ldots$

若产生的迭代序列 $\{x^{(k)}\}$ 收敛到一个确定的向量 x^*，则 x^* 就是原方程组的解。

1. Jacobi 迭代

令 $A = D + L + U$，其中 $D = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$,

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

$(D + L + U)x = b, \Longrightarrow Dx = b - (L + U)x$

若D可逆，则$x = -D^{-1}(L + U)x + D^{-1}\mathrm{b}$

令$\mathrm{G} = -D^{-1}(L + U) = I - D^{-1}A, f = D^{-1}\mathrm{b}$,有$x = \mathrm{G}x + f$

则可得雅可比（Jacobi）迭代格式:$x^{(k+1)} = Gx^{(k)} + f$,

$k = 0, 1, 2, \ldots$，G称为雅可比 (Jacobi) 迭代矩阵。

Jacobi 迭代的分量形式：

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \ldots - a_{1n}x_n^{(k)}\right)\Big/ a_{11} \\ x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \ldots - a_{2n}x_n^{(k)}\right)\Big/ a_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_j^{(k+1)} = \left(b_j - a_{j1}x_1^{(k)} \ldots - a_{j,j-1}x_{j-1}^{(k)} - a_{j,j+1}x_{j+1}^{(k)} \ldots - a_{jn}x_n^{(k)}\right)\Big/ a_{jj} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \ldots - a_{n,n-1}x_{n-1}^{(k)}\right)\Big/ a_{nn} \end{cases}$$

或：

$$x_i^{(k)} = \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k-1)}\right] / a_{ii},\ i = 1, 2, \ldots n,$$

迭代结束条件一般用：

$\left\| x^{(k)} - x^{(k-1)} \right\| \leq \varepsilon$， ε为精度要求，$\|.\|$为某种向量范数（常用∞-范数）

例题3.7 用雅克比迭代解线性方程组：

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

初值取$x_1^{(0)}=x_2^{(0)}=x_3^{(0)}=1$, 精度要求$\varepsilon=10^{-3}$.

解：Jacobi 迭代的分量形式：

$$\begin{cases} x_1^{(k+1)} = \dfrac{\left(7.2 + x_2^{(k)} + 2x_3^{(k)}\right)}{10} \\ x_2^{(k+1)} = \dfrac{\left(8.3 + x_1^{(k)} + 2x_3^{(k)}\right)}{10} \\ x_3^{(k+1)} = \dfrac{\left(4.2 + x_1^{(k)} + x_2^{(k)}\right)}{5} \end{cases}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|x^{(k)}-x^{(k-1)}\|_\infty$
0	1	1	1	
1	1.0200	1.1300	1.2400	0.2400
2	1.0810	1.1800	1.2700	0.0610
3	1.0920	1.1921	1.2922	0.0222
4	1.0977	1.1976	1.2968	0.0057
5	1.0991	1.1991	1.2991	0.0023
6	1.0997	1.1997	1.2997	0.0006

在计算$x_i^{(k+1)}$时，如果用$x_1^{(k+1)},\cdots,x_{i-1}^{(k+1)}$代替$x_1^{(k)},\cdots,x_{i-1}^{(k)}$，则可能会得到更好的收敛效果。此时的迭代公式为

$$
\begin{cases}
x_1^{(k+1)}=\left(b_1-a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\ldots-a_{1n}x_n^{(k)}\right)/a_{11}\\
x_2^{(k+1)}=\left(b_2\boxed{-a_{21}x_1^{(k+1)}}-a_{23}x_3^{(k)}-\ldots-a_{2n}x_n^{(k)}\right)/a_{22}\\
\quad\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
x_j^{(k+1)}=\left(b_j\boxed{-a_{j1}x_1^{(k+1)}\ldots-a_{j,j-1}x_{j-1}^{(k+1)}}-a_{j,j+1}x_{j+1}^{(k)}\ldots-a_{jn}x_n^{(k)}\right)/a_{jj}\\
\quad\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
x_n^{(k+1)}=\left(b_n\boxed{-a_{n1}x_1^{(k+1)}-a_{n2}x_2^{(k+1)}-\ldots-a_{n,n-1}x_{n-1}^{(k+1)}}\right)/a_{nn}
\end{cases}
$$

例题3.7, 改进

$$\begin{cases} x_1^{(k+1)} = \dfrac{\left(7.2 + x_2^{(k)} + 2x_3^{(k)}\right)}{10} \\ x_2^{(k+1)} = \dfrac{\left(8.3 + x_1^{(k+1)} + 2x_3^{(k)}\right)}{10} \\ x_3^{(k+1)} = \dfrac{\left(4.2 + x_1^{(k+1)} + x_2^{(k+1)}\right)}{5} \end{cases}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right\|_\infty$
0	1	1	1	
1	1.0200	1.1320	1.2704	0.2704
2	1.08728	1.1928	1.2960	0.0673
3	1.09848	1.1990	1.2995	0.0112
4	1.0998	1.19988	1.2999	0.0013
5	1.1000	1.2000	1.3000	0.0002

2. Gauss-Seidel 迭代

写成矩阵形式:

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

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

解得 $\boxed{\boldsymbol{x}^{(k+1)} = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + (\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{b}}$ $k = 0, 1, 2, \ldots$

此迭代格式称为高斯-塞德尔 (Gauss-Seidel) 迭代

$\boldsymbol{G} = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}$ 称为 G-S 迭代矩阵

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

程序3.3 P62

3. 迭代的收敛性

定理3.4 设迭代矩阵G的某种范数$\|G\|<1$，则 $x=Gx+f$ 存在唯一解，且对任意初值$x^{(0)}$，迭代序列

$$x^{(k)}= Gx^{(k-1)} + f$$

收敛于x^*，进一步有误差估计式

$$\left\|x^{(k)}-x^*\right\| \le \frac{\|G\|}{1-\|G\|}\left\|x^{(k)}-x^{(k-1)}\right\| \le \frac{\|G\|^k}{1-\|G\|}\left\|x^{(1)}-x^{(0)}\right\|$$

后验估计　　　　先验估计

证明思路：(1)解的存在唯一性; (2)解的收敛性; (3)误差估计式。

证明：（1）若I-G不可逆

则 $(I-G)x=0$ 有非零解x，使得Gx= x

$\|x\|=\|Gx\|\le\|G\|\ \|x\|,$

由$(\|G\|<1)$，$\|x\|<\|x\|,$ 矛盾

所以 I-G可逆，有$(I-G)x=f$有唯一解

$$x^*=(I-G)^{-1}f$$

x^*也是x=Gx+f 的唯一解。

（2）$\left\|x^{(k)}-x^*\right\|=\left\|(Gx^{(k-1)}+f)-(Gx^*+f)\right\|$

$$=\left\|G(x^{(k-1)}-x^*)\right\|$$

$$\le\cdots\le\|G\|^k\left\|x^{(0)}-x^*\right\|\xrightarrow{k\to\infty}0$$

得$x^{(k)}\to x^*$

直接从Ax=b判断

推论3.1　若A按行严格对角占优($|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$), 则解Ax=b的Jacobi迭代和Gauss-Seidel迭代均收敛。

谱半径ρ(G)：G的特征值的模的最大值

引理3.2　设G是方阵，则$G^k \to 0 \Leftrightarrow \rho(G)<1$。

定理3.5　迭代$x^{(k)} = Gx^{(k-1)} + f$对任意初值收敛

$\Leftrightarrow \rho(G)<1.$

三种方法比较

方法一 (推论3.1): 从系数矩阵A判断, A严格对角占优，则Jacobi迭代和Gauss-Seidel迭代收敛, 充分条件, 最方便。

方法二(定理3.4): 从迭代矩阵G判断, 有一种范数$||G||<1$, 充分条件。

方法三(定理3.5): 从迭代矩阵G判断,谱半径$\rho(G) <1$, 充分必要条件, 最宽， 缺点：求特征值，计算困难

例3.8 判断Gauss-Seidel迭代求解 $A_i x=b$ 的收敛性。P60

解（1）系数 矩阵$\boldsymbol{A}=\begin{bmatrix}10 & -1 & -2\\ -1 & 10 & -2\\ -1 & -1 & 5\end{bmatrix}$**,A严格对角占优。**

4. 加速迭代--SOR 迭代

在G-S迭代中

$$x_i^{(k)}=\left(b_i-a_{i1}x_1^{(k)}-\cdots-a_{i,i-1}x_{i-1}^{(k)}-a_{i,i+1}x_{i+1}^{(k-1)}-\cdots-a_{i,n}x_n^{(k-1)}\right)\Big/a_{ii}$$

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

为了得到更好的收敛效果，可选参数ω作$x_i^{(k-1)}$与$x_i^{(k)}$两步的加权平均，于是就得到逐次超松弛迭代法，简称SOR迭代，其中ω称为松弛因子。

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

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

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

收敛的必要条件$0<\omega<2$。此时

$0<\omega<1$,称为低松弛法；让某些高斯赛德尔迭代不收敛变成收敛。

$\omega=1$， 称为**Gauss-Seidel**迭代；

$1<\omega<2$，称为超松弛法。

适当选取**ω，**可以加速收敛速度。

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

Jacobi、GS 和 SOR 算法

Jacobi 算法

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

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

GS 算法

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

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

SOR 算法

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

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

例：解线性方程组 $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$

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

取初始向量 $x^{(0)} = (\,0, 0, 0\,)$，迭代过程中保留小数点后4位。

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

令 $x = (x_1, x_2, x_3)^T$ 则迭代得：

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

$$\vdots$$

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

GS 迭代格式

$$\begin{cases} \boldsymbol{x}_1^{(k)} = \left(1 + \boldsymbol{x}_2^{(k-1)}\right)\big/2 \\ \boldsymbol{x}_2^{(k)} = \left(8 + \boldsymbol{x}_1^{(k)} + \boldsymbol{x}_3^{(k-1)}\right)\big/3 \\ \boldsymbol{x}_3^{(k)} = \left(-5 + \boldsymbol{x}_2^{(k)}\right)\big/2 \end{cases}$$

得 $\boldsymbol{x}^{(1)} = (\ 0.5000,\ 2.8333,\ -1.0833\ )^T$

$$\vdots$$

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

SOR 迭代格式

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

取 $\omega = 1.1$，得

$$x^{(1)} = (\,0.5500,\ 3.1350,\ -1.0257\,)^T$$

$$\vdots$$

$$x^{(7)} = (\,2.0000,\ 3.0000,\ -1.0000\,)^T$$

如何确定SOR迭代中的最优松弛因子是一件很困难的事。

线性方程组的迭代特点
---与非线性方程相比

- 复杂之处

  多维问题：计算量大、理论分析需要矩阵和向量理论

- 简单之处

  线性问题，理论性质好：

  ✓ 存在收敛性的充分必要条件

  ✓ 迭代收敛性仅与方程组系数矩阵有关，与右端无关

  ✓ 迭代收敛性不依赖于初始量的选取

非线性方程组的求解----Matlab

线性方程组的迭代法与直接法（第二章）

- 优势：对大型稀疏方程组，可利用分量形式/稀疏矩阵的方式存储，节省空间。
- 存在问题：可能不收敛

P61-63，算法及程序

分别写出解线性方程组

$$\begin{cases} x_1 - 5x_2 + x_3 = 16 \\ x_1 + x_2 - 4x_3 = 7 \\ -8x_1 + x_2 + x_3 = 1 \end{cases}$$

收敛的雅可比迭代格式与高斯-赛德尔迭代格式

$$\begin{cases} -8x_1 + x_2 + x_3 = 1 \\ x_1 - 5x_2 + x_3 = 16 \\ x_1 + x_2 - 4x_3 = 7 \end{cases}$$

系数矩阵 $\begin{bmatrix} -8 & 1 & 1 \\ 1 & -5 & 1 \\ 1 & 1 & -4 \end{bmatrix}$ 按行严格主对角占优

对应雅可比迭代格式：

$$\begin{cases} x_1^{(k)} = -[1 - x_2^{(k-1)} - x_3^{(k-1)}]/8 \\ x_2^{(k)} = -[16 - x_1^{(k-1)} - x_3^{(k-1)}]/5 \\ x_3^{(k)} = -[7 - x_1^{(k-1)} - x_2^{(k-1)}]/4 \end{cases}$$

对应高斯-赛德尔格式：

$$\begin{cases} x_1^{(k)} = -[1 - x_2^{(k-1)} - x_3^{(k-1)}]/8 \\ x_2^{(k)} = -[16 - x_1^{(k)} - x_3^{(k-1)}]/5 \\ x_3^{(k)} = -[7 - x_1^{(k)} - x_2^{(k)}]/4 \end{cases}$$