数值计算方法

第六章

线性方程组的数值解法

6.1 引言

- 线性方程组的解法
  - 直接法
    - 将方程组的形式，根据某些计算规律，解析成另外一种形式（例如将系数矩阵变为单位矩阵），直接求得线性方程组的解
    - 直接法要求存储的数据很多，因此必须应用存储容量大的计算机
    - 在计算过程中，会产生积累误差，因此计算精度是有限的
    - 直接法的计算时间是固定的、有限的
    - 当线性方程组的维数不大，计算机的容量又较大时，适宜采用直接解法

6.1 引言

- 间接解法
  - 假定一组线性方程组的解，将这组解带入方程组（迭代形式的线性方程组），得到一组新解，再把这组新解代入方程组，然后又得到一组更新的解，反复这样代入求解下去，直到这组解满足一定精度的时候，才结束计算
  - 间接法在计算过程中，只需将系数矩阵中的非零元素存入，因此可使计算机的存储容量小一些
  - 采用反复迭代求解，逐步逼近最佳解，每次迭代都可看成是重新开始计算，因此在计算机中无积累误差
  - 迭代法计算时间较长，迭代格式必须是收敛的
  - 所以对于维数较大的线性方程组，在电子计算机容量有限的情况下，选用间接解法是适宜的

顺序高斯消元法的使用条件

- 定理6.1：如果n阶矩阵A的顺序主子式均不为零，即

$$a_{11} \neq 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{21} \end{vmatrix} \neq 0, \ldots, det\,(A) \neq 0$$

则用高斯消元法求解方程组$Ax=b$时的主元素$a_{kk}^{(k-1)} \neq 0$ $(k=1, 2, \ldots, n)$，也就是说能够用高斯消元法求解该方程组的解

- 定义6.1：设矩阵$A=(a_{ij})_n$每一行对角元素的绝对值都大于同行其它元素绝对值之和

顺序高斯消元法的使用条件

$$|a_{ii}|>\sum_{\substack{j=1\\ j\neq i}}^{n}|a_{ij}| \qquad i=1,2,\ldots,n$$

则称A为严格对角占优矩阵。

- 定理6.2：设方程组$Ax=b$，如果A为严格对角占优矩阵，则用高斯消元法求解时，$a_{kk}^{(k-1)}$全不为零

矩阵的直接三角分解

- 定义6.2：将矩阵A分解成一个下三角矩阵L和一个上三角矩阵U的乘积

$$A=LU$$

称为对矩阵A的三角分解，又称LU分解

矩阵的直接三角分解

- **定理6.4：矩阵*A*的各阶顺序主子式**

$$D_1 = a_{11} \neq 0, \qquad D_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0, \cdots$$

$$D_k = \begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix} \neq 0$$

矩阵的直接三角分解

不为零，可对矩阵A进行LU分解

- 例6.3.1：设 $A=\begin{bmatrix} a+1 & 2 \\ 2 & 1 \end{bmatrix}$，讨论$a$取何值时，

矩阵A可作LU分解

- 解：

$$a+1\neq 0, \qquad (a+1)-2\times 2\neq 0$$

$$a\neq -1, \qquad a\neq 3$$

解三对角方程组的追赶法

- 在实际问题中，经常遇到以下形式的方程组

$$
\begin{cases}
b_1 x_1 + c_1 x_2 = f_1 \\
a_2 x_1 + b_2 x_2 + c_2 x_3 = f_2 \\
\cdots \\
a_k x_{k-1} + b_k x_k + c_k x_{k+1} = f_k \\
\cdots \\
a_{n-1} x_{n-2} + b_{n-1} x_{n-1} + c_{n-1} x_n = f_{n-1} \\
a_n x_{n-1} + b_n x_n = f_n
\end{cases}
$$

解三对角方程组的追赶法

- 这种方程组的系数矩阵称为三对角矩阵，非零元素分布在主对角线及其相邻两条次对角线上

$$A=\begin{pmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ \ddots & \ddots & \ddots & & & \\ & a_k & b_k & c_k & & \\ & & \ddots & a_{n-1} & \ddots & b_{n-1} & c_{n-1} \\ & & & & a_n & & b_n \end{pmatrix}$$

- 针对这种方程组的特点提供一种简便有效的算法—追赶法

解三对角方程组的追赶法

- 定义6.3：称三对角矩阵

$$A=\begin{pmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ \ddots & \ddots & \ddots & & & \\ & a_k & b_k & c_k & & \\ & & \ddots & a_{n-1} & \ddots & b_{n-1} & c_{n-1} \\ & & & & a_n & & b_n \end{pmatrix}$$

为对角占优矩阵，如果其主对角元素的绝对值大于同行次对角元素的绝对值之和，即成立

解三对角方程组的追赶法

$$\begin{aligned} |b_1| &> |c_1| \\ |b_i| &\geq |a_i| + |c_i| \qquad i = 2,3,\dots,n-1 \\ |b_n| &> |a_n| \end{aligned}$$

- 定理6.5：如果所给方程组的系数矩阵是对角占优的，则$l_i(i=1,2,\dots,n)$的值全不为0，可以使用追赶法进行求解

6.4 迭代法

- 迭代法是一种间接求解的方法
- 将给定的方程组转化为迭代形式的方程组，由任意给定的一组初始值进行试探运算，将所得结果作为新的初始值再进行试探运算，这样反复运算多次，不断迭代，最后获得满足精度要求的解

迭代法基本原理

- 迭代法的实现原理
  - 已知线性方程组

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

  - 写成矩阵形式为：

$$Ax=b$$

迭代法基本原理

- 其中

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

- 对给定的方程组 $\boldsymbol{Ax=b}$，写成等价的形式

$$x = Bx + f$$

- 构造一个迭代公式

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

迭代法基本原理

- 当给定初始向量 $x^{(0)}$，进行迭代，生成向量序列

$$x^{(1)}, x^{(2)}, \cdots, x^{(k)}, \cdots$$

- 当向量序列收敛到某个极限向量x^*，即

$$\lim_{k\to\infty} x^{(k)} = x^*$$

- 则x^*是方程组$Ax=b$的精确解，即满足

$$Ax^* = b$$

- 定义**6.4**：对线性方程组$Ax=b$用等价方程$x= Bx+f$建立迭代格式$x^{(k+1)}=Bx^{(k)}+f$，$k=0,1,\ldots$，逐步求解的方法叫迭代法。如果 $\lim_{k\to\infty} x^{(k)} = x^*$ ，则称迭代法收敛于x^*，x^*即是方程组的解，否则称此迭代法发散

迭代法基本原理

- 迭代法对计算机的要求
  - 可节省内存容量
    - 对于大型的系数矩阵，只要存储其中的非零元素就可以运算
    - 消元法，则必须将全部a_{ij}元素都放入内存单元
  - 计算时间较长
    - 迭代法收敛速度较慢（越到最后越慢），如果求解的精度又很高，那么计算的时间是很长的
    - 将给定的方程组转化为迭代形式的方程组时，一定要按照收敛条件去转化

雅可比迭代（Jacobi）

- 雅可比迭代过程
  - 例6.4.1：求解方程组

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

  - 解：分别从上式三个方程中分离出x_1，x_2和x_3

$$\begin{cases} x_1 = \quad 0.2x_2 + 0.1x_3 + 0.3 \\ x_2 = 0.2x_1 \quad + 0.1x_3 + 1.5 \\ x_3 = 0.2x_1 + 0.4x_2 + 2 \end{cases}$$

雅可比迭代（Jacobi）

- 据此可建立迭代公式为

$$\begin{cases} x_1^{(k+1)} = \quad 0.2x_2^{(k)} + 0.1x_3^{(k)} + 0.3 \\ x_2^{(k+1)} = 0.2x_1^{(k)} \quad + 0.1x_3^{(k)} + 1.5 \end{cases}, k = 0,1,2,\cdots$$

$$x_3^{(k+1)} = 0.2x_1^{(k)} + 0.4x_2^{(k)} + 2$$

- 设取迭代初值为 $\boldsymbol{x}^{(0)} = 0 = [0,0,0]^{\mathrm{T}}$
- 下表记录了迭代结果

雅可比迭代（Jacobi）

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.0000	0.0000	0.0000
1	0.3000	1.5000	2.0000
2	0.8000	1.7600	2.6600
3	0.9180	1.9260	2.8640
4	0.9716	1.9700	2.9540
5	0.9894	1.9897	2.9823
6	0.9963	1.9961	2.9938
7	0.9986	1.9986	2.9977
8	0.9995	1.9995	2.9992
9	0.9998	1.9998	2.9998

雅可比迭代（Jacobi）

- 雅可比迭代的一般格式
  - 设线性方程组

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

其中系数矩阵非奇异，且主对角元$a_{ii} \neq 0$，$(i = 1,2,\ldots,n)$

雅可比迭代（Jacobi）

- 由第i个方程解出x_i，有

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n) \\ x_2 = \dfrac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n) \\ \cdots\cdots\cdots\cdots \\ x_n = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1}) \end{cases}$$

雅可比迭代（Jacobi）

- 建立迭代格式

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

- 可以写成

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

雅可比迭代（Jacobi）

- 迭代求解的终止条件
  - 取相邻两次解的每一个分量之差的绝对值中最大的一个 $\max|\Delta x_i| \le \varepsilon$ ， 即

$$\max_{1\le i\le n} | x_i^{(k+1)} - x_i^{(k)} | \le \varepsilon$$

- 雅可比迭代法的流程图

高斯－塞德尔迭代

- 高斯－塞德尔迭代法的思想
  - 迭代收敛时，新值$x_i^{(k+1)}$比老值$x_i^{(k)}$更准确
  - 算出新值$x_i^{(k+1)}$后，用新值$x_i^{(k+1)}$代替用于后面计算的老值$x_i^{(k)}$，使每次迭代计算，都是利用“最新求解信息”
  - 这样，必然会使迭代求解的速度加快

高斯－塞德尔迭代

- 例6.4.2：用高斯-塞德尔迭代法求解方程组

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

- 解：方程组化为等价的方程组

$$\begin{cases} x_1 = \quad 0.2x_2 + 0.1x_3 + 0.3 \\ x_2 = 0.2x_1 \quad + 0.1x_3 + 1.5 \\ x_3 = 0.2x_1 + 0.4x_2 + 2 \end{cases}$$

高斯—塞德尔迭代

- 构造高斯-塞德尔迭代公式

$$\begin{cases} x_1^{(k+1)} = 0.2x_2^{(k)} + 0.1x_3^{(k)} + 0.3 \\ x_2^{(k+1)} = 0.2x_1^{(k+1)} + 0.1x_3^{(k)} + 1.5, k = 0,1,2,\cdots \\ x_3^{(k+1)} = 0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} + 2 \end{cases}$$

- 仍取初值$x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$，进行迭代计算，计算结果如下表所示

高斯—塞德尔迭代

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.00000	0.00000	0.00000
1	0.30000	1.56000	2.68400
2	0.88040	1.94448	2.95387
3	0.98428	1.99224	2.99375
4	0.99782	1.99894	2.99914
5	0.99970	1.99985	2.99988
6	0.99996	1.99998	2.99998

- 由计算结果可以明显看出，高斯—塞德尔迭代法比雅可比迭代法效果好

高斯－塞德尔迭代

- 高斯-塞德尔(Seidel)迭代公式

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

- 一般说来，高斯－塞德尔迭代法比雅可比迭代法好，但是情况并不总是这样，也有高斯－塞德尔迭代法比雅可比迭代法收敛得慢，甚至有雅可比迭代法收敛但高斯－塞德尔迭代法反而发散的例子

迭代收敛的条件

- **定理6.6：如果线性方程组**

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

是对角占优的，则其雅可比迭代公式对于任意给定的初始值都收敛

迭代收敛的条件

- 定理6.7：如果所给方程组是对角占优的，则其高斯-塞德尔迭代公式对于任意给定的初值均收敛

迭代收敛的条件

- 例6.4.3：已知三阶线性方程组

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

检验雅可比迭代公式的收敛性

- 解：首先将原方程组写为迭代形式的方程组，即

$$\begin{cases} x_1 = \frac{20}{8} + \frac{3}{8}x_2 - \frac{2}{8}x_3 \\ x_2 = \frac{33}{11} - \frac{4}{11}x_1 + \frac{1}{11}x_3 \\ x_3 = \frac{36}{12} - \frac{6}{12}x_1 - \frac{3}{12}x_2 \end{cases}$$

迭代收敛的条件

- 任一行之和的最大值<1，即

$$\max\{\frac{5}{8},\frac{5}{11},\frac{9}{12}\}=\frac{9}{12}<1$$

- 因此该方程组采用雅可比迭代法计算是收敛的

松弛法

- 松弛法的主要思想
  - 为了使迭代计算速度更快，提出一种线性加速的方法
  - 将前一步的计算结果$x_i^{(k)}$与高斯－塞德尔方法的迭代值 $\tilde{x}_i^{(k+1)}$ 适当加权平均，期望获得更好的近似值 $x_i^{(k+1)}$
  - 设高斯－塞德尔迭代 公式

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

松弛法

- 加速

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

- 上两式即松弛法，可合并表示为

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

或表示为

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

- 参数ω称为松弛因子，ω=1时迭代格式就是高斯-塞德尔迭代格式，0<ω<1叫低松弛因子，当1<ω<2时叫超松弛因子

松弛法

- 为了保证迭代过程收敛，必须要求$0<\omega<2$
- 由于迭代值 $\tilde{x}_i^{(k+1)}$ 通常比$x_i^{(k)}$精确，在加速公式中加大 $\tilde{x}_i^{(k+1)}$ 的比重，以尽可能扩大它的效果，为此取松弛因子$1<\omega<2$，即采用所谓超松弛法
- 超松弛法简称为SOR方法

松弛法

- 例6.4.4：用SOR方法求解线性方程组

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

- 解：该方程组的精确解为 $x^*=[2, 1, -1]^T$，用SOR法求解，其迭代公式为

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

松弛法

- 取$\omega=1.46$，$x^{(0)}=[0, 0, 0]^T$，计算结果如下表所示

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	3.65	0.8845883	-0.2021098
2	2.321669	0.4230939	-0.2224321
3	2.566140	0.6948260	-0.4852594
…	…	…	…
20	1.999998	1.000001	-1.000003

- 如果$\omega=1$（即高斯－塞德尔迭代法）和同一初始向量$x^{(0)}=[0, 0, 0]^T$，要达到同样精确的近似解，需要迭代110次以上

松弛法

- 例6.4.5：给定一个三元线性方程组

$$\begin{cases} 4x_1 + 3x_2 = 24 \\ 3x_1 + 4x_2 - x_3 = 30 \\ -x_2 + 4x_3 = -24 \end{cases}$$

  - 解：精确解为

$$x^* = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$$

松弛法

- 将具体给定的上述方程组的系数代入超松弛迭代计算式，即可得

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + \dfrac{\omega}{4}(24 - 4x_1^{(k)} - 3x_2^{(k)}) \\ x_2^{(k+1)} = x_2^{(k)} + \dfrac{\omega}{4}(30 - 3x_1^{(k+1)} - 4x_2^{(k)} + x_3^{(k)}) \\ x_3^{(k+1)} = x_3^{(k)} + \dfrac{\omega}{4}(-24 + x_2^{(k+1)} - 4x_3^{(k)}) \end{cases}$$

- 给定初始迭代点$x^{(0)}=[0, 0, 0]^T$，计算精度取$1/2\times 10^{-7}$，则计算结果为：当取$\omega=1$时，则迭代计算34次；当取$\omega=1.25$时，则迭代计算14次