第三章 线性方程组解法

消去法

- Gauss 消去法
- 列/全主元素消去法
- 解三对角方程组的追赶法

矩阵分解法

- 矩阵的 LU 分解
- 正定矩阵的三角分解

方程组的性态和条件数

迭代法

- Jacobi、Gauss-Seidel、SOR 迭代
- 收敛条件及误差估计

引言

设有 n 元线性方程组：

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

若用 A 表示矩阵，x 表示未知数，b 表示右端项，该系统可以写为 $Ax = b$.

引言

最简单的直接解法：Cramer 法则，根据该方法，方程组的解为

$$x_i = \frac{\Delta_i}{\Delta}(i = 1, 2, \ldots, n)$$

其中 $\Delta = \det(A)$，而 Δ_i 是 A 中第 i 列换成向量 b 所得矩阵的行列式。假定用按某行展开的方法计算行列式，那么，Cramer 法则求解一个 n 元线性方程组所需要的乘法运算次数超过 $(n+1)!$。

求解线性方程组的方法

求解线性方程组的方法可分为两大类：直接法和迭代法。

- **直接解法**：假设计算过程没有误差的情况下，经过有限步算术运算可以得到线性方程组的精确解。但是由于实际计算中舍入误差的影响，这类算法一般也只能得到精确解的近似，比如：高斯消去法、三角分解法。
- **迭代解法**：构造适当的向量序列，用某种极限过程来逼近精确解。例如：Jacobi 方法、Gauss-Seidel 方法、SOR 方法等。

§1 *Gauss*消元法

Gauss消元法是最基本的一种方法，下例说明其***基本思想***:

例1. 解线性方程组：

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 12x_1 - 3x_2 + 3x_3 = 15 \\ -18x_1 + 3x_2 - x_3 = -15 \end{cases}$$

解：消去x_1，进行第一次消元：首先找乘数，以-12乘第一个方程加到第二个方程，以18乘第一个方程加到第三个方程上可得同解方程组：

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ -15x_2 - 9x_3 = -57 \\ 21x_2 + 17x_3 = 93 \end{cases}$$

例1（续）

再消一次元得：

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ -15x_2 - 9x_3 = -57 \\ \dfrac{22}{5}x_3 = \dfrac{66}{5} \end{cases}$$

二次消元后将方程化为上三角形式，然后进行回代容易解出：$x_3 = 3, x_2 = 2, x_1 = 1$。

1.1 Gauss消去法

求解思想：

（1）逐次消去变量，将原方程组转化为同解的上三角方程组，这个过程称为消元过程；

（2）按照与方程相反的顺序求解该上三角方程组，这个过程称为回代过程。

——称为高斯消元法。(Gauss Elimination)

消元过程目标：对方程组对应的增广矩阵

$$[A\quad b]=\begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_2 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{pmatrix}$$

做有限次的初等行变换，使得最后的系数矩阵部分变换为上三角矩阵。

7

*Gauss*消元法的消元过程

将初始输入改写为：

$$[A\quad b]=\begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}$$

8

第一步：假设$a_{11}^{(1)}\neq 0$。保持第 1 行不变，依次消去第 1 列中除$a_{11}^{(1)}$以外的所有元素。方法：从第 2 行起，每行(第i行)减去第 1 行乘以$l_{i1}=\dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}$(i=2,3,…n)，得

$$\boxed{\begin{cases} a_{ij}^{(2)}=a_{ij}^{(1)}-l_{i1}a_{1j}^{(1)} \\ b_i^{(2)}=b_i^{(1)}-l_{i1}b_1^{(1)} \end{cases} i,j=2,3,\ldots,n}$$

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & & & & & \vdots \\ 0 & a_{n-1,2}^{(2)} & a_{n-1,3}^{(2)} & & a_{n-1,n}^{(2)} & b_{n-1}^{(2)} \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

9

第二步：假设$a_{22}^{(2)}\neq 0$。保持前 2 行不变，依次消去第 2 列中除$a_{12}^{(1)},a_{22}^{(2)}$以外的所有元素。方法：从第 3 行起，每行(第$i$行)减去第 2 行乘以$l_{i2}=\dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}}$(i=3,…n)，得

$$\boxed{\begin{cases} a_{ij}^{(3)}=a_{ij}^{(2)}-l_{i2}a_{2j}^{(2)} \\ b_i^{(3)}=b_i^{(2)}-l_{i2}b_2^{(2)} \end{cases} i,j=3,\ldots,n}$$

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & a_{n-1,3}^{(3)} & \cdots & a_{n-1,n}^{(3)} & b_{n-1}^{(3)} \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} & b_n^{(3)} \end{pmatrix}$$

10

第k 步：设第k–1步消元后得原方程组的同解方程组为：

$$\begin{cases} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+\ldots+a_{1k}^{(1)}x_k+\ldots+a_{1n}^{(1)}x_n=b_1^{(1)} \\ \qquad a_{22}^{(2)}x_2+\ldots+a_{2k}^{(2)}x_k+\ldots+a_{2n}^{(2)}x_n=b_2^{(2)} \\ \qquad\cdots\cdots\cdots \\ \qquad\qquad a_{kk}^{(k)}x_k+\ldots+a_{kn}^{(k)}x_n=b_k^{(k)} \\ \qquad\qquad\cdots\cdots \\ \qquad\qquad a_{nk}^{(k)}x_k+\ldots+a_{nn}^{(k)}x_n=b_n^{(k)} \end{cases}$$

设$a_{kk}^{(k)}\neq 0$,
记$l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}(i=k+1,\ldots,n)$
将上方程组中第i个方程减去
第k个方程乘以$l_{ik}(i=k+1,\ldots,n)$,
完成第k步消元,得同解方程组：

$$\begin{cases} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+\ldots+a_{1k}^{(1)}x_k+a_{1k+1}^{(1)}x_{k+1}+\ldots+a_{1n}^{(1)}x_n=b_1^{(1)} \\ \qquad a_{22}^{(2)}x_2+\ldots+a_{2k}^{(2)}x_k+a_{2k+1}^{(2)}x_{k+1}+\ldots+a_{2n}^{(2)}x_n=b_2^{(2)} \\ \qquad\cdots\cdots\cdots \\ \qquad\qquad a_{kk}^{(k)}x_k+a_{kk+1}^{(k)}x_{k+1}+\ldots+a_{kn}^{(k)}x_n=b_k^{(k)} \\ \qquad\qquad\qquad a_{k+1,k+1}^{(k+1)}x_{k+1}+\ldots+a_{k+1n}^{(k+1)}x_n=b_{k+1}^{(k+1)} \\ \qquad\qquad\qquad\cdots\cdots\cdots \\ \qquad\qquad\qquad a_{nk+1}^{(k+1)}x_{k+1}+\ldots+a_{nn}^{(k+1)}x_n=b_n^{(k+1)} \end{cases}$$

$$\boxed{\begin{cases} a_{ij}^{(k+1)}=a_{ij}^{(k)}-l_{ik}a_{kj}^{(k)} \\ b_i^{(k+1)}=b_i^{(k)}-l_{ik}b_k^{(k)} \end{cases} \quad (i,j=k+1,\ldots,n)}$$

11

将上述过程进行下去，对n阶方程组，共需进行$n-1$步消元过程，就能将原方程化为同解的上三角形的方程组

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n}^{(n-1)} & b_{n-1}^{(n-1)} \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

12

其等价于线性方程组

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \dots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ \quad a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \dots + a_{2n}^{(2)} = b_2^{(2)} \\ \quad\quad a_{33}^{(3)}x_3 + \dots + a_{3n}^{(3)}x_n = b_3^{(3)} \\ \quad\quad \cdots \\ \quad\quad\quad a_{nn}^{(n)}x_n = b_n^{(n)} \end{cases}$$

13

Gauss消元法的回代过程

回代过程：逐步回代求得原方程组的解

$$\begin{cases} x_n = b_n^{(n)} / a_{nn}^{(n)} \\ x_k = (b_k^{(k)} - \sum_{l=k+1}^{n} a_{kl}^{(k)} x_l) / a_{kk}^{(k)} \quad k = n-1, n-2, \dots, 2, 1 \end{cases}$$

14

Gauss消元法可以进行的充分条件

由以上讨论可知,消元过程能进行到底的条件是：

$a_{kk}^{(k)} \neq 0 \quad (k = 1, 2, \dots, n-1) \quad$ (称$a_{kk}^{(k)}$为主元素),

而只有所有主元素$a_{11}^{(1)}, a_{22}^{(2)}, \dots, a_{nn}^{(n)}$均不为零,才能使用Gauss消去法求解线性方程组：$Ax = b$.

定理：$a_{ii}^{(i)} \neq 0\,(i = 1, 2, \cdots, k)$的充要条件是 A 的前 k 阶顺序主子式均不为 0。

定理：对方程组 $Ax = b$，若 A 的所有顺序主子式均不为 0，则可用高斯消去法将方程组化为三角形方程组，且求出唯一解。

15

Gauss消元法的算法描述

消元过程

```
for k = 1 : n-1
    for i = k+1 : n
        a_ik ← l_ik = a_ik / a_kk
        for j = k+1 : n+1
            a_ij ← a_ij − l_ik a_kj
        end
    end
end
```

回代过程

```
x_n = b_n / a_nn
for i = n-1 : -1 : 1
    a_{i,n+1} ← x_i = (b_i − Σ_{j=i+1}^{n} a_ij x_j) / a_ii
end
```

16

Gauss消元法的计算量

由于在计算机中作乘除运算量所需时间远大于作加减运算所需时间，故只考虑作乘除运算量。

由消元法步骤知，第k次消元需作$n-k$次除法，作$(n-k)(n-k+1)$次乘法，故消元过程中乘除法运算量为：

乘法次数 $\sum_{k=1}^{n-1}(n-k)(n-k+1) = \frac{n}{3}(n^2-1)$	除法次数 $\sum_{k=1}^{n-1}(n-k) = \frac{n}{2}(n-1)$

回代过程中：求x_k需1次除法和$n-k$次乘法，因此回代过程运算量为：

除法：n次；　乘法：$\sum_{k=1}^{n}(n-k) = \frac{n}{2}(n-1)$

所以***Gauss*** 消去法的乘除法总运算量为：

$$N = \frac{n}{3}(n^2-1) + \frac{n}{2}(n-1) + n + \frac{n}{2}(n-1) = \frac{n^3}{3} + n^2 - \frac{n}{3}$$

17

1.2 主元消去法

对一般的很好求解的系统，高斯消去法计算效果不一定好。

例 2. $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

该系统的解很容易得到，但是却无法用高斯消去法。

18

例3. $\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 1 \\ 2 \end{bmatrix}$，其中$\varepsilon$是一个小量。

解. 若利用高斯消去法 $\begin{bmatrix} \varepsilon & 1 \\ 0 & 1-\varepsilon^{-1} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 1 \\ 2-\varepsilon^{-1} \end{bmatrix}$

对应的解为：$\begin{cases} x_2=\left(2-\varepsilon^{-1}\right)/\left(1-\varepsilon^{-1}\right)\approx 1 \\ x_1=\left(1-x_2\right)\varepsilon^{-1}\approx 0 \end{cases}$

而该问题的真解为 $\begin{cases} x_1=1/\left(1-\varepsilon\right)\approx 1 \\ x_2=\left(1-2\varepsilon\right)/\left(1-\varepsilon\right)\approx 1 \end{cases}$

若在上述例子求解过程中交换两行，则问题将有所改善：

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

高斯消去法可以推出 $\begin{bmatrix} 1 & 1 \\ 0 & 1-\varepsilon \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 2 \\ 1-2\varepsilon \end{bmatrix}$

其对应的解为 $\begin{cases} x_2=(1-2\varepsilon)/(1-\varepsilon)\approx 1 \\ x_1=2-x_2=1 \end{cases}$

注：除了避免$a_{kk}^{(k)}\neq 0$外，还应避免主元素绝对值过小。

列主元消去法

列主元素法的具体步骤如下：

$$[A^{(1)} \quad b^{(1)}]=\begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}$$

在该过程中，主元是按列选取的，故称为列主元法。

第一步：首先在第1列中选取绝对值最大的元素作为主元素，即：$\left|a_{i_1,1}^{(1)}\right|=\max\limits_{1\le i\le n}\left|a_{i1}^{(1)}\right|$

将第1行与第i_1行交换，然后进行第一次消元，得矩阵$[A^{(2)},b^{(2)}]$

第二步：在矩阵$[A^{(2)},b^{(2)}]$的第2列中选主元，使$\left|a_{i_2 2}^{(2)}\right|=\max\limits_{2\le i\le n}\left|a_{i2}^{(2)}\right|$

将矩阵$[A^{(2)},b^{(2)}]$的第2行与第i_2行列换，再进行第二次消元，得矩阵$[A^{(3)},b^{(3)}]$。

第k步：在矩阵$[A^{(k)},b^{(k)}]$的第k列中选主元，使$\left|a_{i_k k}^{(k)}\right|=\max\limits_{k\le i\le n}\left|a_{ik}^{(k)}\right|$

将矩阵$[A^{(k)},b^{(k)}]$的第k行与第i_k行列换，进行第k次消元.

全主元消去法

如果不是按列选主元，而是在全体待选系数：$a_{ij}^{(k)}(i,j=k,k+1,\ldots,n)$中选取主元，则得到全主元法，其计算过程如下：

第一步：在全体系数$a_{ij}(i,j=1,2,\ldots,n)$中选取绝对值最大的元素作为主元，并通过行与列的互换到$a_{11}^{(1)}$的位置，然后进行第一次消元，得到矩阵$(A^{(2)},b^{(2)})$。

第k步：在矩阵$A^{(k)}$的右下方$(n-k+1)$阶子矩阵的所有元素$a_{ij}^{(k)}(i,j=k,k+1,\ldots,n)$中，选取绝对值最大的元作为主元，并通过行与列的互换将它换到$a_{kk}^{(k)}$的位置，然后进行第k次消元。

全主元消去法：精度高，但是要换列和记录次序，比较麻烦。

例4. 用主元素法求解线性方程组：$\begin{cases} x_1+x_2+x_3=6 \\ 12x_1-3x_2+3x_3=15 \\ -18x_1+3x_2-x_3=-15 \end{cases}$

计算过程保留三位小数。

解法1（列主元法）：求解过程如下：

因其增广矩阵为：$\begin{pmatrix} 1 & 1 & 1 & 6 \\ 12 & -3 & 3 & 15 \\ \underline{-18} & 3 & -1 & -15 \end{pmatrix}$

第一与第三行互换 $\begin{pmatrix} \underline{-18} & 3 & -1 & -15 \\ 12 & -3 & 3 & 15 \\ 1 & 1 & 1 & 6 \end{pmatrix}$

第一次消元 $\begin{pmatrix} -18 & 3 & -1 & -15 \\ 0 & -1 & 2.333 & 5 \\ 0 & \underline{1.167} & 0.944 & 5.167 \end{pmatrix}$

第二,三行互换 $\begin{pmatrix} -18 & 3 & -1 & -15 \\ 0 & \underline{1.167} & 0.944 & 5.167 \\ 0 & -1 & 2.333 & 5 \end{pmatrix}$

第二次消元 $\begin{pmatrix} -18 & 3 & -1 & -15 \\ 0 & 1.167 & 0.944 & 5.167 \\ 0 & 0 & 3.142 & 9.428 \end{pmatrix}$

由回代过程求解得：$x_3=3.001$，$x_2=2.00$，$x_1=1.00$

解法2（全主元法）：其求解过程如下：

它的增广矩阵为：$\begin{pmatrix} 1 & 1 & 1 & 6 \\ 12 & -3 & 3 & 15 \\ \underline{-18} & 3 & -1 & -15 \end{pmatrix}$

第一,三行互换 $\begin{pmatrix} \underline{-18} & 3 & -1 & -16 \\ 12 & -3 & 3 & 15 \\ 1 & 1 & 1 & 6 \end{pmatrix}$

第一次消元 $\begin{pmatrix} -18 & 3 & -1 & -15 \\ 0 & -1 & \underline{2.333} & 5 \\ 0 & 1.167 & 0.944 & 5.167 \end{pmatrix}$

第二,三列互换 $\begin{pmatrix} -18 & -1 & 3 & -15 \\ 0 & \underline{2.333} & -1 & 5 \\ 0 & 0.944 & 1.167 & 5.167 \end{pmatrix}$

第二次消元 $\begin{pmatrix} -18 & -1 & 3 & -15 \\ 0 & 2.333 & -1 & 5 \\ 0 & 0 & 1.572 & 3.144 \end{pmatrix}$

由回代解得：$x_2=2.00$，$x_3=3.00$，$x_1=1.00$

1.3 解三对角方程组的追赶法

在很多问题中，需要解如下形式的三对角方程组：

$$\begin{cases} b_1x_1+c_1x_2 & =d_1 \\ a_2x_1+b_2x_2+c_2x_3 & =d_2 \\ \quad a_3x_2+b_3x_3+c_3x_4 & =d_3 \\ \quad \cdots \quad \cdots \quad \cdots & \\ \qquad a_{n-1}x_{n-2}+b_{n-1}x_{n-1}+c_{n-1}x_n & =d_{n-1} \\ \qquad\qquad a_nx_{n-1}+b_nx_n & =d_n \end{cases}$$

三对角方程组的系数矩阵为三对角阵，对于这种特殊而又简单的方程组，用前面介绍的方法求解由于有大量的零元素既占内存又浪费计算时间，显然很不经济。充分注意到三对角方程组的特点，根据顺序消元的思想导出一个简便的算法——**追赶法**。

首先进行顺序消元，且每步将主元系数化为1，将方程组化为：

$$\begin{cases} x_1 + q_1 x_2 = p_1 \\ \quad x_2 + q_2 x_3 = p_2 \\ \quad\quad \cdots\cdots \\ \quad\quad x_{n-1} + q_{n-1} x_n = p_{n-1} \\ \quad\quad\quad x_n = p_n \end{cases}$$

其中系数按下式计算：

$$\begin{cases} p_1 = d_1 / b_1, \quad q_1 = c_1 / b_1 \\ t_k = b_k - a_k q_{k-1} \quad (k = 2, 3, \ldots, n) \\ p_k = (d_k - a_k p_{k-1}) / t_k, q_k = c_k / t_k \end{cases}$$

回代求解得：

$$\begin{cases} x_n = p_n \\ x_k = p_k - q_k x_{k+1} \quad (k = n-1, n-2, \ldots, 2, 1) \end{cases}$$

25

定理：设三对角方程组的系数矩阵满足

（1）所有 a_k, b_k, c_k 均不为 0；

（2）$|b_k| \geq |a_k| + |c_k|$，且 $(k = 1, 2, \cdots, n, \; a_1 = c_n = 0)$，且其中至少有一个取不等号。

则追赶法计算过程中每步的分母 $t_k = b_k - a_k q_{k-1}$ 满足

$$|t_k| = |b_k - a_k q_{k-1}| \geq |b_k| - |a_k| > 0 \quad (k = 2, 3, \cdots, n)$$

因此追赶法能进行到底。

26

例5 用追赶法解下列三对角方程组：

$$\begin{cases} x_1 = 0 \\ x_1 + 4x_2 + x_3 = -6 \\ \quad x_2 + 4x_3 + x_4 = 6 \\ \quad\quad x_3 + 4x_4 + x_5 = 18 \\ \quad\quad\quad x_5 = 0 \end{cases}$$

解：首先将方程组化为（先追）：

$$\begin{cases} x_1 = 0 \\ x_2 + \frac{1}{4} x_3 = -\frac{3}{2} \\ x_3 + \frac{4}{15} x_4 = 2 \\ x_4 + \frac{15}{56} x_5 = 30/7 \\ x_5 = 0 \end{cases}$$

然后回代（赶）求解：

$x_5 = 0$，$x_4 = 30/7$，

$x_3 = 6/7$，$x_2 = -12/7$，

$x_1 = 0$

可以看出，追赶法本质上还是顺序消元法，但由于计算过程中只涉及系数矩阵的非零元，因此大大节约了计算机内存与计算量，按乘除法次数进行比较，*Gauss*消元法约为$n^3/3$，而追赶法仅为$5n-3$次，可见追赶法是求解三对角方程组的非常好的方法。

27

§2 矩阵分解法

几类特殊的线性方程组求解：

1、若矩阵具有对角结构，即求解

$$\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

2、若矩阵具有下三角结构，即求解

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

3、若矩阵具有上三角结构，即求解

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

28

想法：

如果一个矩阵A可以化为一个下三角矩阵 L和一个上三角矩阵 U的乘积，

即$A = LU$，那么 $Ax = b \Leftrightarrow LUx = b$

令$\begin{cases} Ly = b & (\text{求出}y) \\ Ux = y & (\text{求出}x) \end{cases}$，其对应的解即为$Ax = b$的解。

于是问题转化为如何对给定的矩阵 做分解。

29

2.1 *Gauss*消元法的矩阵形式

如果用矩阵形式表示，*Gauss*消元法的消元过程是对方程组的增广矩阵（A、b）进行一系列的初等行变换，将系数矩阵A化成上三角矩阵的过程，也等价于用一串初等变换阵去左乘增广矩阵，因此，消元过程可以通过矩阵运算来实现。

事实上，*Gauss*消元法的第一次消元相当于用初等矩阵：

$$L_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \cdots & \cdots & \cdots & \\ -l_{n1} & & & 1 \end{pmatrix}$$

左乘增广矩阵 $(A^{(1)}, b^{(1)}) = (A, b)$,

其中 $l_{i1} = a_{i1}^{(1)} / a_{11}^{(1)} \quad (i = 2, 3, \ldots, n)$，于是：$L_1(A^{(1)}, b^{(1)}) = (A^{(2)}, b^{(2)})$

30

第二次消元相当于用初等矩阵：

$$L_2=\begin{pmatrix}1&&&&\\&1&&&\\&-l_{32}&1&&\\&\cdots&\cdots&\cdots&\\&-l_{n2}&&&1\end{pmatrix}$$

左乘增广矩阵$\left(A^{(2)},b^{(2)}\right)$,

其中$l_{i2}=a_{i2}^{(2)}/a_{22}^{(2)}(i=3,4,\ldots,n)$

于是：$L_2\left(A^{(2)},b^{(2)}\right)=\left(A^{(3)},b^{(3)}\right)$

即有 $L_2L_1\left(A^{(1)},b^{(1)}\right)=\left(A^{(3)},b^{(3)}\right)$

第k次消元相当于用初等矩阵：

$$L_k=\begin{pmatrix}1&&&&&\\&\ddots&&&&\\&&1&&&\\&&&1&&\\&&&-l_{k+1,k}&1&\\&&&\vdots&&\ddots&\\&&&-l_{n,k}&&&1\end{pmatrix}$$

左乘矩阵$\left(A^{(k)},b^{(k)}\right)$,其中$l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}(i=k+1,k+2,\ldots,n)$

于是：$L_k\left(A^{(k)},b^{(k)}\right)=\left(A^{(k+1)},b^{(k+1)}\right)$

即有：$L_kL_{k-1}\ldots L_2L_1\left(A^{(1)},b^{(1)}\right)=\left(A^{(k+1)},b^{(k+1)}\right)$

31

经过$n-1$步消元后得到：

$$\left(A^{(n)},b^{(n)}\right)=\begin{pmatrix}a_{11}^{(1)}&a_{12}^{(1)}&\ldots&a_{1n}^{(1)}&b_1^{(1)}\\&a_{22}^{(2)}&\ldots&a_{2n}^{(2)}&b_2^{(2)}\\&&\ldots&\ldots&\ldots\\&&&a_{nn}^{(n)}&b_n^{(n)}\end{pmatrix}$$

$$=L_{n-1}\left(A^{(n-1)},b^{(n-1)}\right)$$

$$=L_{n-1}L_{n-2}\ldots L_1\left(A^{(1)},b^{(1)}\right)$$

因为$L_k\,(k=1,2,\ldots,n-1)$均为非奇异阵，故它们的逆矩阵存在。容易求出：

$$L_k^{-1}=\begin{pmatrix}1&&&&&\\&\ddots&&&&\\&&1&&&\\&&l_{k+1,k}&1&&\\&&\vdots&&\ddots&\\&&l_{n,k}&&&1\end{pmatrix}$$

令：$L=L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}=\begin{pmatrix}1&&&&&\\l_{21}&1&&&&\\l_{31}&l_{32}&1&&&\\\ldots&\ldots&\ldots&\ldots&1&\\l_{n1}&l_{n2}&l_{n3}&\ldots&l_{nn-1}&1\end{pmatrix}$

于是有：$(A,b)=(A^{(1)},b^{(1)})=L_1^{-1}L_2^{-1}\ldots L_{n-1}^{-1}\left(A^{(n)},b^{(n)}\right)$

即：$(A,b)=(LA^{(n)},Lb^{(n)})=(LU,Ly)$

这说明：在 $a_{kk}^{(k)}\neq0(k=1,2,\ldots,n-1)$的条件下，消元过程实际上是把系数矩阵$A$分解成单位下三角阵与上三角矩阵的乘积的过程。

32

2.2 Doolittle分解——LU分解

由上述过程，可以将矩阵A分解成两个三角形矩阵的乘积，即：$A=LU$，其中L为单位下三角矩阵，U为上三角矩阵：

$$L=\begin{pmatrix}1&&&&&\\l_{21}&1&&&&\\l_{31}&l_{32}&1&&&\\\ldots&\ldots&\ldots&\ldots&1&\\l_{n1}&l_{n2}&l_{n3}&\ldots&l_{nn-1}&1\end{pmatrix}\qquad U=\begin{pmatrix}u_{11}&u_{12}&u_{13}&\ldots&u_{1n}\\&u_{22}&u_{23}&\ldots&u_{2n}\\&&\ldots&&\ldots\\&&&\ldots&\ldots\\&&&&u_{nn}\end{pmatrix}$$

上述分解称为杜利特尔（Doolittle）分解，也称为LU分解。

定理. 设A为n阶方阵，矩阵A存在唯一的Doolittle分解的充要条件为：A的顺序主子式$A_i(i=1,2,\ldots,n-1)$均不为零。

33

下面讨论如何对A进行LU分解：

由于两个矩阵相等就是它们的对应元素都相等，因此通过比较A与LU的对应元素，即可得到直接计算L、U的元素的公式：$A=LU$即：

$$\begin{pmatrix}a_{11}&a_{12}&a_{13}&\ldots&a_{1n}\\a_{21}&a_{22}&a_{23}&\ldots&a_{2n}\\a_{31}&a_{32}&a_{33}&\ldots&a_{3n}\\\ldots&\ldots&&&\\a_{n1}&a_{n2}&a_{n3}&\ldots&a_{nn}\end{pmatrix}=\begin{pmatrix}1&&&&\\l_{21}&1&&&\\l_{31}&l_{32}&1&&\\\ldots&\ldots&\ldots&\ldots&\\l_{n1}&l_{n2}&l_{n3}&\ldots&1\end{pmatrix}\begin{pmatrix}u_{11}&u_{12}&\ldots&u_{1n}\\&u_{22}&\ldots&u_{2n}\\&&\ldots&\ldots\\&&&u_{nn}\end{pmatrix}$$

由矩阵乘法规则及比较式子两端的元素，得：

$$\begin{cases}a_{1j}=u_{1j}\\a_{i1}=l_{i1}u_{11}\end{cases}\Rightarrow\begin{cases}u_{1j}=a_{1j}&(j=1,2,\ldots,n)\\l_{i1}=a_{i1}/u_{11}&(i=2,3,\ldots,n)\end{cases}$$

34

对A进行LU分解 （k行k 列）

假设已经求得U的前k-1行，L的前k-1列，下面讨论U的第k行，L的第k列：

$$a_{kj}=\sum_{s=1}^{n}l_{ks}u_{sj}$$
$$=\sum_{s=1}^{k-1}l_{ks}u_{sj}+l_{kk}u_{kj}+\sum_{s=k+1}^{n}l_{ks}u_{sj}=\sum_{s=1}^{k-1}l_{ks}u_{sj}+u_{kj}\qquad j\geq k$$

$$a_{ik}=\sum_{s=1}^{n}l_{is}u_{sk}$$
$$=\sum_{s=1}^{k-1}l_{is}u_{sk}+l_{ik}u_{kk}+\sum_{s=k+1}^{n}l_{is}u_{sk}=\sum_{s=1}^{k-1}l_{is}u_{sk}+l_{ik}u_{kk}\qquad i>k$$

得到计算u_{kj}和l_{ik}的公式：

35

对A进行LU分解的具体步骤

1. 计算U的第1行，L的第1列，亦称为计算第1框；
2. 计算U的第k行，L的第k列（k=2,…,n），即第k框；

$$u_{kj}=a_{kj}-\sum_{s=1}^{k-1}l_{ks}u_{sj}\,(j=k,k+1,\ldots,n)$$

$$l_{ik}=(a_{ik}-\sum_{s=1}^{k-1}l_{is}u_{sk})/u_{kk}\,(i=k+1,\ldots,n)$$

36

三角分解的紧凑格式

矩阵的三角分解可按以下格式及顺序进行。这种格式既便于记忆，又便于计算，称为**紧凑格式**。

$(a_{11})u_{11}$	$(a_{12})u_{12}$	$(a_{13})u_{13}$	……	$(a_{1n})u_{1n}$
$(a_{21})l_{21}$	$(a_{22})u_{22}$	$(a_{23})u_{23}$	……	$(a_{2n})u_{2n}$
$(a_{31})l_{31}$	$(a_{32})l_{32}$	$(a_{33})u_{33}$	……	$(a_{3n})u_{3n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$(a_{n1})l_{n1}$	$(a_{n2})l_{n2}$	$(a_{n3})l_{n3}$	……	$(a_{nn})u_{nn}$

例6. 求矩阵的：$A=\begin{pmatrix} 2 & 2 & 3 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{pmatrix}$三角分解。

（2）2	（2）2	（3）3
$(4)\ \frac{4}{2}=2$	（7）7–2×2=3	（7）7–2×3=1
$(-2)\frac{-2}{2}=-1$	（4）（4–（–1）×2）/3=2	（5）5–2×1–（–1）×3=6

由表可得：

$$L=\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix} \quad U=\begin{pmatrix} 2 & 2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 6 \end{pmatrix}$$

所以 $A=LU$

三角分解法的几点说明

1、用三角分解法求线性方程组的乘除法运算量也是$n^3/3$数量级。由于在求出u_{ij}, l_{ij}和y_i后，a_{ij}和b_i无需保留，故上机计算时，可把L，U和y存在A，b所占的单元，回代时x取代y，整个计算过程中不需要增加新的存贮单元。

2、从三角分解法的推导中可以看出，系数矩阵的三 角分解与右端项无关。因 而在计算多个系数矩阵为A而右端不同的线性方程组系时，用三角分解法更为 简便（如可用于求逆矩阵）。

3、也可以把矩阵A分解成一个下三角矩阵与一个单位上三角矩阵的乘积，矩阵的 这种分解称为**克劳特（Crout）分解**。

2.2 正定矩阵的三角分解

定理：若A是对称正定矩阵，则存在唯一的单位下三角矩阵L和对角阵：

$$D=diag(u_{11},u_{22},\dots,u_{nn})\ (u_{ii}>0, i=1,2,\dots,n),$$

使得：$A=LDL^T$。

推论：若A是对称正定矩阵，则存在唯一的主对角线元素都为正的下三角阵L，使得：$A=LL^T$（矩阵的这种分解称为Cholesky分解，也称平方根分解）。

平方根法分解算法

$A=LL^T$，用比较法可以导出L的计算公式。设 $L=\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \dots & & \ddots & \\ l_{n1} & l_{n2} & & l_{nn} \end{pmatrix}$

$$a_{ij}=\sum_{s=1}^{j} l_{is}l_{js}=\sum_{s=1}^{j-1} l_{is}l_{js}+l_{ij}l_{jj} \qquad (i\ge j)$$

可以得到

$$\begin{cases} l_{jj}=(a_{jj}-\sum_{k=1}^{j-1} l_{jk}^2)^{1/2} & (j=1,2,\dots,n) \\ l_{ij}=(a_{ij}-\sum_{k=1}^{j-1} l_{ik}l_{jk})/l_{jj} & (i=j+1,j+2,\dots,n) \end{cases}$$

改进平方根法分解算法

$A=LDL^T=LU,\quad U=DL^T$

若记$d_i=u_{ii}$,则$\frac{u_{sk}}{d_s}=l_{ks}$，由矩阵的LU分解算法

$$u_{kj}=a_{kj}-\sum_{s=1}^{k-1} l_{ks}u_{sj}(j=k,k+1,\dots,n)$$

$$l_{ik}=(a_{ik}-\sum_{s=1}^{k-1} l_{is}u_{sk})/u_{kk}(i=k+1,\dots,n)$$

可得改进的平方根算法

$$\begin{cases} d_k=a_{kk}-\sum_{s=1}^{k-1} d_s l_{ks}^2 \\ l_{ik}=(a_{ik}-\sum_{s=1}^{k-1} d_s l_{is}l_{ks})/d_k \quad i=k+1,\cdots,n \end{cases}$$

（改进）平方根法的几点说明

1. 利用平方根法只需求L，不需求U，计算量比LU分解少了近一半；同时由于A的对称性，可以节省约一半的存储量。

2. 基于上述LDL^T分解的方法称为改进的平方根法，其乘除运算量与*Cholesky*分解相当，且避免了开方运算。计算顺序按先行后列逐层分解计算；

§3 方程组的性态和条件数

向量范数

定义：在R^n上定义了$\|\cdot\|$，$\forall x,y\in R^n$，如果满足

（1）$\|x\|\ge 0$，且$\|x\|=0\Leftrightarrow x=0$；（正定性）

（2）$\forall a\in R$，有$\|ax\|=|a|\|x\|$；（齐次性）

（3）$\|x+y\|\le\|x\|+\|y\|$；（三角不等式）

则称$\|\cdot\|$为R^n中的一个范数（或模）。

常用的向量范数

$\forall x\in R^n$，$x=(x_1,\cdots,x_n)^T$，

（1）1 范数：$\|x\|_1=\sum_{i=1}^n|x_i|$；

（2）2 范数：$\|x\|_2=(\sum_{i=1}^n x_i^2)^{1/2}$；

（3）∞ 范数：$\|x\|_\infty=\max_{1\le i\le n}|x_i|$；

（4）p 范数：$\|x\|_p=(\sum_{i=1}^n|x_i|^p)^{1/p}$，$p\ge 1$；

事实上，$\lim_{p\to\infty}\|x\|_p=\|x\|_\infty$。

有了向量范数，就可以用它来表示方程组解向量的误差，设x是方程组$Ax=b$的准确解向量，$\tilde{x}$是近似解向量，则

$\|x-\tilde{x}\|_p$，$\dfrac{\|x-\tilde{x}\|_p}{\|x\|_p}$

分别称为$\tilde{x}$关于$p-$范数的绝对误差与相 对误差。

矩阵范数

定义：在$R^{n\times n}$上定义了$\|\cdot\|$，$\forall A,B\in R^{n\times n}$，如果满足

（1）$\|A\|\ge 0$，且$\|A\|=0\Leftrightarrow A=0$；（正定性）

（2）$\forall a\in R$，有$\|aA\|=|a|\|A\|$；（齐次性）

（3）$\|A+B\|\le\|A\|+\|B\|$；（三角不等式）

（4）$\|AB\|\le\|A\|\|B\|$；（相容性条件）

则称$\|\cdot\|$为$R^{n\times n}$中的一个矩阵范数。

例：Frobenius（佛罗贝尼乌斯）范数，又称为 Euclid 范数：$\|A\|_F=(\sum_{i,j=1}^n a_{ij}^2)^{1/2}$。

矩阵范数与向量范数的关系

定义：对给定的矩阵范数$\|\cdot\|$和向量范数$\|\cdot\|$，若对任意的$x\in R^n$，$A\in R^{n\times n}$，有

$\|Ax\|\le\|A\|\|x\|$，则称矩阵范数$\|\cdot\|$与向量范数$\|\cdot\|$是相容的。

注：对任意给定的一种矩阵范数$\|\cdot\|$，总存在与该范数相容的向量范数。

例：定义向量范数$\|x\|:=\|xe_1^T\|$，$\|Ax\|=\|Axe_1^T\|\le\|A\|\|xe_1^T\|=\|A\|\|x\|$。

定理：设在R^n中给定了一种范数$\|\cdot\|$，对任意矩阵$A\in R^{n\times n}$，定义$\|A\|=\max_{x\ne 0}\dfrac{\|Ax\|}{\|x\|}$，

则 $\|A\|$是一种矩阵范数，并且它与所给定的向量范数是相容的。

常用的矩阵范数

有了矩阵范数，就可以用它 来描述矩阵的误差。设$\tilde{A}$是A的近似矩阵，$A-\tilde{A}$称为$\tilde{A}$的误差矩阵，则

$\|A-\tilde{A}\|_p$，$\dfrac{\|A-\tilde{A}\|_p}{\|A\|_p}$

分别称为$\tilde{A}$关于$p-$范数的绝对误差与 相对误差。

（1）$\|A\|_\infty=\max_{x\ne 0}\dfrac{\|Ax\|_\infty}{\|x\|_\infty}=\max_{1\le i\le n}\sum_{j=1}^n|a_{ij}|$；（行范数）

（2）$\|A\|_1=\max_{x\ne 0}\dfrac{\|Ax\|_1}{\|x\|_1}=\max_{1\le j\le n}\sum_{i=1}^n|a_{ij}|$；（列范数）

（3）$\|A\|_2=\max_{x\ne 0}\dfrac{\|Ax\|_2}{\|x\|_2}=\sqrt{\lambda_{max}(A^TA)}$；（谱范数）

注意：当A为对称矩阵时，$\|A\|_2=\rho(A)$（$\rho(A)=\max_i|\lambda_i|$，称为谱半径）。

两个常用结论

定理：矩阵 A 的谱半径不超过 A 的任意一种矩阵范数，即有 $\rho(A)\le\|A\|$。

定理：若 $\|A\|<1$，则 $I\pm A$ 为非奇异矩阵，且当范数为算子范数时，有

$$\|(I\pm A)^{-1}\|\le\frac{1}{1-\|A\|}。$$

舍入误差的影响及算法的稳定性

前面曾介绍了多种解线性方程组的方法，由于计算机字长的限制，无论哪种方法，求解的每一步几乎都可能引入新的舍入误差，这些误差随着计算的推进而向前传播。

算法不同误差的累积情况不一样，舍入误差对解的影响不大的算法称为数值稳定的算法。

可以证明：主元素法、约当消去法、Cholesky分解法和追赶法都是数值稳定的算法。

数值稳定的算法是否一定能求得精度比较高的解呢？回答是不一定，解的精度还与方程组本身的性态有关，下面来考察几个例：

$\begin{cases}x_1+x_2=2\\x_1+1.0001x_2=2\end{cases}$	$\begin{cases}x_1+x_2=2\\x_1+1.0001x_2=2.0001\end{cases}$	$\begin{cases}x_1+x_2=2\\x_1+1.0005x_2=2.0001\end{cases}$
$\Downarrow$	$\Downarrow$	$\Downarrow$
$x_1=2,\quad x_2=0$	$x_1=1,\quad x_2=1$	$x_1=1.8,\quad x_2=0.2$

可以看出，后两个方程组与第一个方程组相比，系数矩阵或右端向量仅有0.0005以下的误差，但准确解却相差很大。

方程组的性态

右端项或是系数矩阵的误差会引起解的误差：

（1）右端项误差：

若 $A(x+\delta x)=b+\delta b$ 有 $\dfrac{\|\delta x\|}{\|x\|}\le\|A\|\|A^{-1}\|\dfrac{\|\delta b\|}{\|b\|}$；

（2）系数矩阵误差

$(A+\delta A)(x+\delta x)=b$ 有 $\dfrac{\|\delta x\|}{\|x\|}\le\dfrac{\|A\|\|A^{-1}\|\dfrac{\|\delta A\|}{\|A\|}}{1-\|A\|\|A^{-1}\|\dfrac{\|\delta A\|}{\|A\|}}$

条件数

定义：称 $cond(A)=\|A\|\|A^{-1}\|$ 为矩阵 A 的条件数。

注：$\|A\|\|A^{-1}\|$ 表征系数矩阵误差、右端项误差对解的误差的影响大小，其越小越好。

常用的条件数

（1）$cond(A)_1=\|A\|_1\|A^{-1}\|_1$

（2）$cond(A)_\infty=\|A\|_\infty\|A^{-1}\|_\infty$

（3）$cond(A)_2=\|A\|_2\|A^{-1}\|_2$

特别，当 A 为对称正定矩阵时，$cond(A)_2=\left|\dfrac{\lambda_{max}(A)}{\lambda_{min}(A)}\right|$.

条件数与线性方程组

定义：A 非奇异，

若 $cond(A)$ 很大，$Ax=b$ 病态线性方程组（A 为病态矩阵）；

若 $cond(A)$ 较小，$Ax=b$ 良态线性方程组（A 为良态矩阵）。

定理：设 $Ax=b$，A 非奇异，x 为精确解，$\bar{x}$ 为方程组的近似解，其残（剩余）向量为 $r=b-A\bar{x}$，则有误差估计

$$\frac{\|x-\bar{x}\|}{\|x\|}\le cond(A)\frac{\|r\|}{\|b\|}.$$

§4 迭代法

迭代法基本思想:

将 $Ax=b$ 化为等价的方程 $x=Mx+g$，取初始向量 $x^{(0)}$，按照以下格式迭代：$x^{(k+1)}=Mx^{(k)}+g$，则生成向量序列 $x^{(1)},x^{(2)},x^{(3)},\cdots,x^{(k)},\cdots$，若序列收敛于 x^*，则有 $x^*=Mx^*+g$，即 x^* 为原方程 $Ax=b$ 的解。

向量序列的收敛性:

向量序列 $x^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T$，若存在向量 $x^*=(x_1^*,x_2^*,\cdots,x_n^*)^T$ 满足 $\lim\limits_{k\to\infty}x_i^{(k)}=x_i^*$，则称向量序列 $\{x^{(k)}\}$ 收敛于常向量 x^*，记为 $\lim\limits_{k\to\infty}x^{(k)}=x^*$。

$x^{(k+1)}=Mx^{(k)}+g$：迭代格式

M ：迭代矩阵

55

Jacobi方法

设有n阶线性方程组：
$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2 \\ \cdots\cdots\cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\ldots+a_{nn}x_n=b_n \end{cases}$$

设$a_{ii}\neq 0$ (1,2,…,n),可将上式改写为等价方程组：
$$\begin{cases} x_1=(\quad -a_{12}x_2-a_{13}x_3-\ldots-a_{1n}x_n+b_1)/a_{11} \\ x_2=(-a_{21}x_1 \quad -a_{23}x_3-\ldots-a_{2n}x_n+b_2)/a_{22} \\ \cdots\cdots\cdots\cdots \\ x_n=(-a_{n1}x_1-a_{n2}x_2-\ldots-a_{nn-1}x_{n-1} \quad +b_n)/a_{nn} \end{cases}$$

56

由此可建立迭代格式：
$$\begin{cases} x_1^{(k+1)}=(\quad -a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\ldots-a_{1n}x_n^{(k)}+b_1)/a_{11} \\ x_2^{(k+1)}=(-a_{21}x_1^{(k)} \quad -a_{23}x_3^{(k)}-\ldots-a_{2n}x_n^{(k)}+b_2)/a_{22} \\ \cdots\cdots\cdots \\ x_n^{(k+1)}=(-a_{n1}x_1^{(k)}-a_{n2}x_2^{(k)}-\ldots-a_{ni}x_i^{(k)}-\ldots-a_{nn-1}x_{n-1}^{(k)} \quad +b_n)/a_{nn} \end{cases}$$

分量形式： $x_i^{(k+1)}=(b_i-\sum\limits_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n$

矩阵形式： $x^{(k+1)}=D^{-1}(L+U)x^{(k)}+D^{-1}b$

注： $A=D-L-U$

57

Gauss-Seidel方法

前面的Jacobi迭代格式：
$$\begin{cases} x_1^{(k+1)}=(\quad -a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\ldots-a_{1n}x_n^{(k)}+b_1)/a_{11} \\ x_2^{(k+1)}=(-a_{21}x_1^{(k)} \quad -a_{23}x_3^{(k)}-\ldots-a_{2n}x_n^{(k)}+b_2)/a_{22} \\ \cdots\cdots\cdots \\ x_n^{(k+1)}=(-a_{n1}x_1^{(k)}-a_{n2}x_2^{(k)}-\ldots-a_{ni}x_i^{(k)}-\ldots-a_{nn-1}x_{n-1}^{(k)} \quad +b_n)/a_{nn} \end{cases}$$

可作修正化为：
$$\begin{cases} x_1^{(k+1)}=(\quad -a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\ldots-a_{1n}x_n^{(k)}+b_1)/a_{11} \\ x_2^{(k+1)}=(-a_{21}x_1^{(k+1)} \quad -a_{23}x_3^{(k)}-\ldots-a_{2n}x_n^{(k)}+b_2)/a_{22} \\ \cdots\cdots\cdots \\ x_n^{(k+1)}=(-a_{n1}x_1^{(k+1)}-a_{n2}x_2^{(k+1)}-\ldots-a_{nn-1}x_{n-1}^{(k+1)} \quad +b_n)/a_{nn} \end{cases}$$

58

Gauss-Seidel方法

分量形式： $x_i^{(k+1)}=(b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum\limits_{j=i+1}^{n}a_{ij}x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n$

矩阵形式： $x^{(k+1)}=(D-L)^{-1}Ux^{(k)}+(D-L)^{-1}b$

59

SOR方法(Successive Over-Relaxation)

前面的Gauss-Seidel迭代格式：
$$\begin{cases} x_1^{(k+1)}=(\quad -a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\ldots-a_{1n}x_n^{(k)}+b_1)/a_{11} \\ x_2^{(k+1)}=(-a_{21}x_1^{(k+1)} \quad -a_{23}x_3^{(k)}-\ldots-a_{2n}x_n^{(k)}+b_2)/a_{22} \\ \cdots\cdots\cdots \\ x_n^{(k+1)}=(-a_{n1}x_1^{(k+1)}-a_{n2}x_2^{(k+1)}-\ldots-a_{ni}x_i^{(k+1)}-\ldots-a_{nn-1}x_{n-1}^{(k+1)} \quad +b_n)/a_{nn} \end{cases}$$

其分量形式为：
$$x_i^{(k+1)}=(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)})/a_{ii}=x_i^{(k)}+\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i}^{n}a_{ij}x_j^{(k)})$$

改进： $x_i^{(k+1)}=x_i^{(k)}+\dfrac{\omega}{a_{ii}}(b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum\limits_{j=i}^{n}a_{ij}x_j^{(k)})$

60

SOR方法(Successive Over-Relaxation)

矩阵形式：$x^{(k+1)}=(D-\omega L)^{-1}((1-\omega)D+\omega U)x^{(k)}+\omega(D-\omega L)^{-1}b$

分量形式：

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

例：对方程组$\begin{bmatrix}2&-1&1\\1&1&1\\1&1&-2\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1\\0\\1\end{bmatrix}$，

（1）写出 Jacobi 迭代格式；

（2）写出 Gauss-Seidel 迭代格式；

(3) 写出 SOR 迭代格式。

迭代法的收敛性

向量序列

定义：设有向量序列 $x^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T$，若存在向量 $x^*=(x_1^*,x_2^*,\cdots,x_n^*)^T$ 满足 $\lim_{k\to\infty}x_i^{(k)}=x_i^*$，则称向量序列 $\{x^{(k)}\}$ 收敛于常向量 x^*，记为 $\lim_{k\to\infty}x^{(k)}=x^*$。

定理：设有向量序列 $\{x^{(k)}\}$ 和常向量 x^*，若对某种范数有 $\lim_{k\to\infty}\|x^{(k)}-x^*\|=0$，则有 $\lim_{k\to\infty}x^{(k)}=x^*$。

矩阵序列

定义：设有矩阵序列 $A^{(k)}=(a_{ij}^{(k)})_{n\times n}$，若存在矩阵 $A=(a_{ij})_{n\times n}$ 满足 $\lim_{k\to\infty}a_{ij}^{(k)}=a_{ij}$，则称矩阵序列 $\{A^{(k)}\}$ 收敛于常矩阵 A，记为 $\lim_{k\to\infty}A^{(k)}=A$。

定理：设有矩阵序列 $\{A^{(k)}\}$ 和常矩阵 A，若对某种范数有 $\lim_{k\to\infty}\|A^{(k)}-A\|=0$，则有 $\lim_{k\to\infty}A^{(k)}=A$。

迭代法收敛的充要条件

定理：设方程组 $x=Mx+g$ 有唯一解，对任意的初始向量 $x^{(0)}$，迭代法 $x^{(k+1)}=Mx^{(k)}+g$ 收敛的充要条件是 $\lim_{k\to\infty}M^k=0$。

定理：设方程组 $x=Mx+g$ 有唯一解，对任意的初始向量 $x^{(0)}$，迭代法收敛的充要条件是 $\rho(M)<1$，这里 $\rho(M)=\max\{|\lambda|:\det(\lambda I-M)=0\}$ 为谱半径。

注：矩阵 A 的谱半径不超过 A 的任意一种矩阵范数，即有 $\rho(A)\le\|A\|$。

迭代法收敛的充分条件

定理：如果矩阵 M 的某种范数 $\|M\|<1$，则

（1）$x=Mx+g$ 的解 x^* 存在唯一；

（2）对于迭代公式 $x^{(k+1)}=Mx^{(k)}+g$，有 $\lim_{k\to\infty}x^{(k)}=x^*$，$\forall x_0\in R^n$，并且成立

$$\|x^{(k)}-x^*\|\le\frac{\|M\|}{1-\|M\|}\|x^{(k)}-x^{(k-1)}\|；$$

$$\|x^{(k)}-x^*\|\le\frac{\|M\|^k}{1-\|M\|}\|x^{(1)}-x^{(0)}\|。$$

收敛性结果

定理：设 $Ax=b$，若 A 为严格对角占优矩阵，则 Jacobi 迭代和 Gauss-Seidel 迭代均收敛。

定理：SOR 方法收敛的必要条件是 $0<w<2$。

定理：设 $Ax=b$，A 对称正定，若 $0<w<2$，则 SOR 迭代法收敛。

注：$w=1$ 对应的 SOR 方法为 Gauss-Seidel 方法，故 A 对称正定时，Gauss-Seidel 迭代法收敛。