矩阵初等变换

初等变换前后矩阵等价

1. 交换两行或两列
2. 用一个数乘以某一行
3. 用某个数乘以某一行加到另一行中

高斯列主元消去法

1 高斯消元法

1. 一旦遇到某个主元为0的情况，则消元过程无法继续进行
2. 主元的绝对值很小时，求出的结果误差极大

2 列主元消去法

计算步骤

1. 确定方程的$i1$，使$|a_{i1}| = \max|a_{i1}|, i \in [1, n]$，选取$|a_{i1}|$作为第一主元，交换第1个和第$i$个方程，利用第一个方程将后$n-1$个方程中的$x_1$消去
2. 在第二列中重复上述过程，消去x_2
3. 经过$n-1$次重复后，将原方程变为上三角形方程，按n递减顺序回代x_n求得结果

编程步骤，对于增广矩阵：$[A, b]$

1. 对$k \in [1, n-1]$
   (a) 选择主元，确定r，使$|a_{r1}| = \max|a_{r1}|, r \in [1, n]$
   (b) 交换$[A^{(k)}, b^{(k)}]$中的r, k两行
   (c) 对$i \in [k+1, n]$，计算$m_{ik} = \frac{a_{ik}}{a_{kk}}$
   (d) $a_{ij} \leftarrow a_{ij} - m_{ik}a_{kj}, \quad b_i \leftarrow b_i - m_{ik}b_k$
2. $x_n = b_n/a_{nn}, \quad x_i = \left(b_i - \sum_{k=i+1}^{n} a_{ik}x_k\right)/a_{ii} \quad (i = n-1, n-2, \cdots, 1),$
   $m_{ik} = a_{ik}/a_{rk} \quad a_{ij} \leftarrow a_{ij} - m_{ik}a_{kj}, b_i \leftarrow b_i - m_{ik}b_k$

追赶法

考虑如下形式的三对角方程组

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \tag{1}$$

其元素满足

$$\begin{cases} |b_1| > |c_1| > 0 \\ |b_i| \geq |a_i| + |c_i| \text{ 且 } a_i c_i \neq 0 \quad (i = 2, 3, \cdots, n-1) \end{cases} \tag{2}$$

$$\left. |b_n| > |a_n| > 0 \right.$$

具有这种特殊系数矩阵的方程组具有唯一解，可以顺序消元法求解

计算步骤如下，

第一步，取$\beta_1 = c_1/b_1, y_1 = f_1/b_1$，将方程增广矩阵第一行主元单位化，有

$$\bar{A}^{(1)} = \begin{bmatrix} 1 & & \beta_1 & & & & y_1 \\ a_2 & & b_2 & & c_2 & & f_2 \\ & \ddots & & \ddots & & \ddots & \vdots \\ & & & a_{n-1} & & b_{n-1} & c_{n-1} & f_{n-1} \\ & & & & & a_n & b_n & f_n \end{bmatrix} \tag{3}$$

第二步，作$n-1$次初等变换，使得矩阵中第k行元素乘以$-a_{k+1}$再加到第$k+1$行元素上，然后将第$k+1$行主元单位化，使得增广矩阵变形为

$$\bar{A}^{(n)} = \begin{bmatrix} 1 & \beta_1 & & & & y_1 \\ & 1 & & \beta_2 & & y_2 \\ & & \ddots & & \ddots & \vdots \\ & & & 1 & \beta_{n-1} & y_{n-1} \\ & & & & 1 & y_n \end{bmatrix} \tag{4}$$

其中，

$$\begin{aligned} &\beta_i = c_i / (b_i - a_i \beta_{i-1}) \quad (i = 2, 3, \cdots, n-1) \\ &y_i = (f_i - a_i y_{i-1}) / (b_i - a_i \beta_{i-1}) \quad (i = 2, 3, \cdots, n) \end{aligned} \tag{5}$$

则，原方程等价于

$$\begin{bmatrix} 1 & \beta_1 & & & & \\ & 1 & & \beta_2 & & \\ & & \ddots & & \ddots & \\ & & & 1 & & \beta_{n-1} \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} \tag{6}$$

对于这种上三角方程，在回代过程中只需要进行$n-1$次乘法就能得到方程的解

三对角方程组消元法又称追赶法

迭代法

对于变量极多的线性方程组，常用迭代方法，

对于方程组

$$A\vec{x} = \vec{b} \tag{7}$$

设计一迭代公式，任选一初始向量，使得

$$\vec{x}^{(0)}, \vec{x}^{(1)}, \cdots, \vec{x}^{(k)}, \cdots, \tag{8}$$

若该向量序列收敛，其极限值为原方程组的解，即

$$\lim_{k \to \infty} x_i^{(k)} = x_i^* \quad i = 1, 2, \cdots, n \tag{9}$$

记$\vec{x}^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$，有

$$\lim_{k\to\infty}\vec{x}^{(k)}=\vec{x}^{*} \tag{10}$$

1 雅可比迭代法

对于方程组

$$\begin{cases} a_{11}x_1+a_{12}x_2+a_{13}x_3+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+a_{23}x_3+\cdots+a_{2n}x_n=b_2 \\ \qquad\qquad\vdots \\ a_{n1}x_1+a_{n2}x_2+a_{n3}x_3+\cdots+a_{nn}x_n=b_n \end{cases} \quad\Rightarrow\quad A\vec{x}=\vec{b} \tag{11}$$

若$a_{ii}\neq 0 \quad (i=1,2,\dots,n)$，有

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

将上述方程组分解成下述三个矩阵，

$$D=\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix},\quad L=\begin{bmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & & & \ddots & \\ a_{11} & a_{12} & a_{13} & \cdots & 0 \end{bmatrix},$$
$$U=\begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & 0 & \ddots & \\ & & & \ddots & a_{n-1n} \\ & & & & 0 \end{bmatrix} \tag{13}$$

则上述方程组的解可以写为

$$\vec{x}=-D^{-1}(L+U)\vec{x}+D^{-1}\vec{b} \tag{14}$$

令，$B=-D^{-1}(L+U), \quad d=D^{-1}b$，则

$$\vec{x}=B\vec{x}+d \tag{15}$$

取，$\vec{x}^{(0)}=\left(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)}\right)^T$，代入上述迭代公式，有

$$\begin{cases} x_1^{(1)}=\frac{1}{a_{11}}\left(-a_{12}x_2^{(0)}-a_{13}x_3^{(0)}\cdots-a_{1n}x_n^{(0)}+b_1\right) \\ x_2^{(1)}=\frac{1}{a_{22}}\left(-a_{21}x_1^{(0)}-a_{23}x_3^{(0)}\cdots-a_{2n}x_n^{(0)}+b_2\right) \\ \qquad\qquad\vdots \\ x_n^{(1)}=\frac{1}{a_{nn}}\left(-a_{n1}x_1^{(0)}-a_{n2}x_2^{(0)}\cdots-a_{n,n-1}{x_{n-1}}^{(0)}+b_n\right) \end{cases} \tag{16}$$

即

$$\vec{x}^{(1)}=B\vec{x}^{(0)}+d \tag{17}$$

反复迭代有，

$$\vec{x}^{(k+1)}=B\vec{x}^{(k)}+d \quad k=0,1,2,\cdots \tag{18}$$

若，

$$\lim_{k\to\infty}\vec{x}^{(k)}=\vec{x}^{*} \tag{19}$$

即，$\vec{x}$收敛，则$\vec{x}^{*}$就是方程组的解

综上所述，有雅可比迭代法的迭代公式

$$\begin{aligned} x_i^{(k+1)} &= \frac{1}{a_{it}}\left(-\sum_{j=1}^{k-1} a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n} a_{ij}x_j^{(k)}+b_i\right) \\ &= x_i^{(k)}+\frac{1}{a_{ii}}\left(b_i-\sum_{j=1}^{n} a_{ij}x_j^{(k)}\right) \end{aligned} \tag{20}$$

其中，$i=1,2,\cdots,n;\quad k=0,1,2,\cdots$

2　高斯-塞德尔迭代法

雅可比迭代的每一步计算中，需要对$\vec{x}^{(k)}$的全部分量进行计算，

当计算第i个分量时，已经算出的$i-1$个分量都未被利用

若在迭代过程中，用新计算得到的分量代替旧的分量计算，就是高斯-塞德尔迭代法

取初始向量：$\vec{x}^{(0)}=\left(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)}\right)^T$

第一次迭代有结果

$$\begin{cases} x_1^{(1)}=\frac{1}{a_{11}}\left(-a_{12}x_2^{(0)}-a_{13}x_3^{(0)}-\cdots-a_{1n}x_n^{(0)}+b_1\right) \\ x_2^{(1)}=\frac{1}{a_{22}}\left(-a_{21}x_1^{(1)}-a_{23}x_3^{(0)}-\cdots-a_{2n}x_n^{(0)}+b_2\right) \\ \vdots \\ x_n^{(1)}=\frac{1}{a_{nn}}\left(-a_{n1}x_1^{(1)}-a_{n2}x_2^{(1)}-\cdots-a_{n,n-1}x_{n-1}^{(1)}+b_n\right) \end{cases} \tag{21}$$

可以看出每一个解都用了前一个解的本次迭代结果

简化有

$$\begin{aligned} \vec{x}^{(k+1)} &= -D^{-1}\left(L\vec{x}^{(k+1)}+U\vec{x}^{(k)}\right)+D^{-1}\vec{b} \\ D\vec{x}^{(k+1)} &= -L\vec{x}^{(k+1)}-U\vec{x}^{(k)}+\vec{b} \\ (D+L)\vec{x}^{(k+1)} &= -U\vec{x}^{(k)}+\vec{b} \end{aligned} \tag{22}$$

有

$$\vec{x}^{(k+1)}=-(D+L)^{-1}U\vec{x}^{(k)}+(D+L)^{-1}\vec{b} \tag{23}$$

令，$G=-(D+L)^{-1}U,d_1=(D+L)^{-1}b$，则矩阵形式为

$$\vec{x}^{(k+1)}=G\vec{x}^{(k)}+d_1 \tag{24}$$

或写作

$$\begin{cases} x_1^{(k+1)}=\frac{1}{a_{11}}\left(-a_{12}x_2^{(k)}-a_{13}x_3^{(k)}-\cdots-a_{1n}x_n^{(k)}+b_1\right) \\ x_2^{(k+1)}=\frac{1}{a_{22}}\left(-a_{21}x_1^{(k+1)}-a_{23}x_3^{(k)}-\cdots-a_{2n}x_n^{(k)}+b_2\right) \\ \vdots \\ x_n^{(k+1)}=\frac{1}{a_{nn}}\left(-a_{n1}x_1^{(k+1)}-a_{n2}x_2^{(k+1)}-\cdots-a_{n,n-1}x_{n-1}^{(k+1)}+b_n\right) \end{cases} \tag{25}$$

综上，其计算公式为

$$
\begin{aligned}
x_i^{(k+1)} &= \frac{1}{a_{ii}}\left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x^{(k)} + b_i\right) \\
&= x_i^{(k)} + \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right)
\end{aligned} \tag{26}
$$

其中，$i = 1, 2, .., n, \quad k = 0, 1, 2, ...$

该算法相比雅可比迭代收敛速度快的多

3　超松弛迭代法

采用雅可比迭代法或高斯-塞德尔迭代法求解时，其收敛速度往往很慢，为加快收敛速度，可以采用超松弛迭代法

对于给定的线性方程组

$$
A \cdot \vec{x} = \vec{b} \tag{27}
$$

将A分解为$A = I - B$，则该方程组等价于

$$
\begin{aligned}
&\vec{x} = B\vec{x} + \vec{b} \quad (B = I - A) \\
&\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{b} \\
&\qquad = (I - A)\vec{x}^{(k)} + \vec{b} \\
&\qquad = \vec{x}^{(k)} + \vec{b} - A\vec{x}^{(k)} \\
&\qquad = \vec{x}^{(k)} + \vec{r}^{(k)} \quad (k = 0, 1, 2, \cdots)
\end{aligned} \tag{28}
$$

其中，$\vec{r}^{(k)} = \vec{b} + A\vec{x}^{(k)}$称为剩余向量，

注意剩余向量$\vec{r}^{(k)}$并不是方程的解，

给$\vec{r}^{(k)}$加上一适当因子ω，从而得到一个加速迭代公式

$$
\vec{x}^{(k+1)} = \vec{x}^{(k)} + \omega\left(\vec{b} - A\vec{x}^{(k)}\right) \tag{29}
$$

其中，ω称为松弛因子

则上式的分量形式为

$$
\begin{aligned}
&x_i^{(k+1)} = x_i^{(k)} + \omega\left(b_i - \textstyle\sum_{j=1}^{n} a_{ij}x_j^{(k)}\right) \\
&(i = 1, 2, \cdots, n; k = 0, 1, 2, \cdots)
\end{aligned} \tag{30}
$$

这种加速方法称为**带松弛因子的同时迭代法**，该方法对ω的要求很高

同样考虑高斯-塞德尔算法的思想，可以得到**逐次超松弛迭代法SOR法**

其迭代公式为

$$
\begin{aligned}
&x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right) \\
&(i = 1, 2, \cdots, n; k = 0, 1, 2, \cdots)
\end{aligned} \tag{31}
$$

相当于高斯法公式增加一松弛因子，即逐次超松弛迭代公式

若$\omega < 1$，称为低松弛法

若$\omega > 1$，称为超松弛法

超松弛法常用于求解大型稀疏矩阵方程组，但需要慎重选择松弛因子

积分方程的数值解法

1　积分方程的定义及分类

1.1　Fredholm方程（弗氏方程）

第一类

$$f(x)=\lambda\int_a^b G(x,s)y(s)ds \tag{32}$$

第二类

$$f(x)=y(x)-\lambda\int_a^b G(x,s)y(s)ds \tag{33}$$

其中，f,G已知，G是积分方程的核，λ为参数，y为未知函数

1.2　Volterra方程（渥氏方程）

第一、二类弗氏方程中上限b换为变量x。

2　有限求和方法

将求解积分方程变换为求解线性方程组

1. 积分方程离散化
2. 用有限求和代替积分

以第二类弗氏方程为例

$$f(x)=y(x)-\lambda\int_a^b G(x,s)y(s)ds \tag{34}$$

积分方程离散化：将积分区间$[a,b]$按步长离散，得离散化积分方程

$$y_i-\lambda\int_a^b G\left(x_i,s\right)y_i(s)ds=f_i \tag{35}$$

有限求和代替积分，有

$$\int_a^b G\left(x_i,s\right)y_i(s)ds\approx h\sum_{j=1}^{N}C_jG_{ij}y_j \tag{36}$$

其中，C_j则对应于不同积分方法所采用的系数，

辛普森法对应系数如下

$$C_1=C_N=1/3\quad C_j=\begin{cases}4/3 & j=2,4,\cdots\\ 2/3 & j=3,5,\cdots\end{cases} \tag{37}$$

则积分方程化为

$$y_i-\lambda h\sum^{N}C_jG_{ij}y_j=f_i\quad(i=1,2,\cdots,N) \tag{38}$$

$j=1$

该方程可以看作一个关于y的线性方程组，

由此可以求出$y(x)$在各点出的数值解