线性方程组 $Ax = b$ 的数值解法

《计算方法》课程笔记

2024 年 11 月 5 日

§1 引言

大量实际的科学与工程问题常可归结为求解含有多个未知量 $x_1, x_2, ..., x_n$ 的线性代数方程组. 即求

$$\begin{cases} \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n = \beta_1 \\ \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n = \beta_2 \\ \cdots \\ \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{nn}x_n = \beta_2 \end{cases}$$

的值. 写为矩阵形式就为 $\mathbf{A}\mathbf{x} = \mathbf{b}$. 其中,

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \alpha_{n3} & \alpha_{nn} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}.$$

在线性代数中我们学到的是精确求解方法. 下面来看几个方法:

1.1. 矩阵求逆.

a) Gauss 消元法 假设有 $\mathbf{A}\mathbf{x} = \mathbf{b}$, 只要在两端乘上对应的逆, 就可以由 $\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ 推出 $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. 这个可以使用消元法求得方程组的解. 例如

$$\begin{cases} x - y + 2z = 5 & (1) \\ x + y + z = 6 & (2) \\ 2x + 3y + 5z = 23 & (3) \end{cases}$$

假设 (x_0, y_0, z_0) 是它的解, 那么

- 任意交换 2 行, 解不变: 比如交换 (1) 和 (2), $\begin{cases} \color{blue}{x + y + z = 6} & (1) \\ \color{blue}{x - y + 2z = 5} & (2) \\ 2x + 3y + 5z = 23 & (3) \end{cases}$ 的解也满足 (x_0, y_0, z_0).

- 某一行乘上一个非 0 倍数, 解不变. 如把第二行乘上 2, $\begin{cases} x - y + 2z = 5 & (1) \\ 2x + 2y + 2z = 12 & (2) \\ 2x + 3y + 5z = 23 & (3) \end{cases}$ 的解也满足 (x_0, y_0, z_0).
- 某一行乘上一个倍数加到另一行去, 解不变. 如 (2) 乘上 2 加到 (3) 上去, 得到 $\begin{cases} x - y + 2z = 5 & (1) \\ x + y + z = 6 & (2) \\ 4x + 5y + 7z = 35 & (3) \end{cases}$ 的解也满足 (x_0, y_0, z_0).

如果把系数 $\mathbf{A}$ 和常数项 $\mathbf{b}$ 排列为矩阵的形式, 就可以由以下的方法解得方程组:

- 按照行从第一行到最后一行上对角线的位置迭代. 第 i 次迭代的目的是把第 $i+1$ 行 i 列到 n 行 i 列的数 (包含两端) 全都置为 0(红色部分). 我们可以通过上述观察的第三条性质满足.
- 如果发现 i 行 i 列的数已经为 0, 试图找一个不为 0 的行与之交换.
  - 如果找不到这样的不为 0 的行, 说明方程组无解或有无穷多解 (即 $0x = 0$ 或 $0x = b$ 的情况).
  - 如果找到了, 交换他们, 然后继续进行行的迭代.
- 最后, 左侧矩阵一定会类似于上三角, 最后一行会形如 $ax = b$(紫色). 只需要应用第二条规则就可以解得一个变量. 从而可以反推出方程的解.

例 1. 来看一个具体的例子: 增广矩阵形如

$$\left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & 3 & 5 & 23 \end{array}\right]$$

的方程组, 使用上述的消元法的过程:

消去过程的第一步, 目的是把第二行第一列到第三行第一列的所有元素消去:

$$\left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & 3 & 5 & 23 \end{array}\right] \xrightarrow{\text{第二行}+=\text{第一行}\times(-1)} \left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 0 & 2 & -1 & 1 \\ 2 & 3 & 5 & 23 \end{array}\right] \xrightarrow{\text{第三行}+=\text{第一行}\times(-2)} \left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 0 & 2 & -1 & 1 \\ 0 & 5 & 1 & 13 \end{array}\right]$$

消去过程的第二步, 目的是把第三行第二列到第三行第二列的所有元素消去:

$$\left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 0 & 2 & -1 & 1 \\ 0 & 5 & 1 & 13 \end{array}\right] \xrightarrow{\text{第三行}+=\text{第二行}\times(-\frac{5}{2})} \left[\begin{array}{ccc:c} 1 & -1 & 2 & 5 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & 7/2 & 21/2 \end{array}\right]$$

我们得到了一个阶梯型. 这时候只要使用第二条, 就有

$$\left[\begin{array}{ccc:c}1&-1&2&5\\0&2&-1&1\\0&0&7/2&21/2\end{array}\right]\xrightarrow{\text{第三行}\times=(\frac{2}{7})}\left[\begin{array}{ccc:c}1&-1&2&5\\0&2&-1&1\\0&0&1&3\end{array}\right]$$

接下来就是回带的过程. 就像刚刚一样, 这次是从后往前. 第一步的先把第三列头上的消干净:

$$\left[\begin{array}{ccc:c}1&-1&2&5\\0&2&-1&1\\0&0&1&3\end{array}\right]\xrightarrow{\text{第二行}+=\text{第三行}}\left[\begin{array}{ccc:c}1&-1&2&5\\0&2&0&4\\0&0&1&3\end{array}\right]\xrightarrow[\text{第二行}/=2]{\text{第一行}+=\text{第三行}\times(-2)}\left[\begin{array}{ccc:c}1&-1&0&-1\\0&1&0&2\\0&0&1&3\end{array}\right]$$

和刚刚一样的操作, 就有

$$\left[\begin{array}{ccc:c}1&-1&0&-1\\0&1&0&2\\0&0&1&3\end{array}\right]\xrightarrow{\text{第一行}+=\text{第二行}}\left[\begin{array}{ccc:c}1&0&0&1\\0&1&0&2\\0&0&1&3\end{array}\right]$$

于是我们得到了 $x=1, y=2, z=3$.

这就得到了方程的解. 要想得到逆矩阵, 便可以在旁边放一个单位阵. 我们知道三种初等的行变换就是可以被初等矩阵表示的. 这样, 每一次行 (列) 变换乘在一起还是一个矩阵, 便是我们的逆矩阵. 从而, 把 $\mathbf{A}$ 变为 $\mathbf{I}$ 的过程中, 实际上是左乘一个逆矩阵; 对于旁边的单位阵而言, 我们就得到的 $\mathbf{A}^{-1}$. 即

$$[\mathbf{A}|\mathbf{I}]\xrightarrow[\text{单位阵}]{\text{经过初等行列变换化为}}[\mathbf{A}^{-1}\mathbf{A}|\mathbf{A}^{-1}\mathbf{I}]\to[\mathbf{I}|\mathbf{A}^{-1}].$$

b) Cramer 法则 如果一个矩阵 $\mathbf{A}$ 的行列式不为 0, 那么就可以使用 Cramer 法则. 可以使用 $x_i=\frac{|\mathbf{A}_i|}{|\mathbf{A}|}, i=1,2,...,n$. 其中, $|\mathbf{A}|$ 是方程组对应系数矩阵对应的行列式; $|\mathbf{A}_i|$ 是以右端变量向量 $\mathbf{b}$ 替代 $\mathbf{A}$ 的第 i 列所得矩阵的行列式. 即

$$x_i=\frac{\begin{vmatrix}a_{1,1}&\cdots&a_{1,i-1}&b_1&a_{1,i+1}&\cdots&a_{1,n}\\a_{2,1}&\cdots&a_{2,i-1}&b_2&a_{2,i+1}&\cdots&a_{2,n}\\\vdots&\ldots&\vdots&\vdots&\vdots&\ldots&\vdots\\a_{n,1}&\cdots&a_{n,i-1}&b_n&a_{n,i+1}&\cdots&a_{n.n}\end{vmatrix}}{\begin{vmatrix}a_{1,1}&a_{1,2}&\cdots&a_{1,n}\\a_{2,1}&a_{2,2}&\cdots&a_{2,n}\\\vdots&\vdots&\ldots&\vdots\\a_{n,1}&a_{n,2}&\cdots&a_{n,n}\end{vmatrix}}.$$

但是, 计算一个 n 阶行列式需要做 $(n-1)(n!)$ 次乘法, 运用 Cramer 法则求解上述方程所需乘除法的运算量大约为

$$N = \underbrace{(n+1)}_{\substack{\text{分子的}n\text{个}\\ \text{和分母的1个}}} \times (n-1)n! + \underbrace{n}_{\text{求}x_i}.$$

因此, 当线性方程组的阶数 n 较高时, 计算量太大, 现实上不可行! 因此, 快速高效地数值求解线性方程组是数值线性代数研究中的核心问题, 也是目前科学计算中的重大研究课题之一.

1.2. 解线性方程组的两种方法.

直接法 只包含有限次四则运算. 在计算过程中都不发生舍入误差的假定下, 计算结果就是原方程组的精确解. 包括

- Gauss 消元法
- 三角分解法

注. 由于运算过程中舍入误差的存在, 实际上直接方法得到的解也是方程组的近似解.

迭代法 把方程组的解向量看作是某种极限过程的极限, 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. 如

- Jacobi 迭代法
- Gauss-Seidel 迭代法
- 超松弛 (SOR) 迭代法

§2 列主元 Gauss 消元法

在引言中已经知道 Gauss 消元法的一般步骤. 我们发现若方程组是“上/下三角方程组”, 可很容易求解. 从而, 我们的解法就是分为如下两步走:

- 运用初等变换构造等价的上三角方程组 $Ux = y$. (消去)
- 按照方程组的逆序求解上三角方程组.（回代）

通过三种初等变换

- Interchanges 行对换：对调方程组的两行
- Scaling 行倍乘：用非零常数乘以方程组的某一行
- Replacement 行倍加：将方程组的某一行乘以一个非零常数, 再加到另一行上

便可以得到上面的结果.

首先更加仔细地看一下 Gauss 消元的过程.

2.1. Gauss 消去法的过程. 方程组 $\mathbf{Ax} = \mathbf{b}$ 的增广矩阵记为

$$(\mathbf{A}^{(0)}\ \ \mathbf{b}^{(0)}) = \begin{pmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & \cdots & a_{1,n}^{(0)} & \beta_1^{(0)} \\ a_{2,1}^{(0)} & a_{2,2}^{(0)} & a_{2,3}^{(0)} & \cdots & a_{2,n}^{(0)} & \beta_2^{(0)} \\ a_{3,1}^{(0)} & a_{3,2}^{(0)} & a_{3,3}^{(0)} & \cdots & a_{3,n}^{(0)} & \beta_3^{(0)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1}^{(0)} & a_{n,2}^{(0)} & a_{n,3}^{(0)} & \cdots & a_{n,n}^{(0)} & \beta_n^{(0)} \end{pmatrix}.$$

其中右上角的角标 (m) 表示第 m 次迭代, 即第 m 次消元的结果.

如果 $a_{11} \neq 0$, 第 1 步消元的工作就是对于第 $2 \sim$ n 行: 将第 i 行分别减去第 1 行乘上 a_{11} 的倍数

$$l_{i.1} = \frac{a_{i,1}^{(0)}}{a_{1,1}^{(0)}}, i = 1, 2, ..., n,$$

得到

$$(\mathbf{A}^{(1)}\ \ \mathbf{b}^{(1)}) = \begin{pmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & \cdots & a_{1,n}^{(0)} & \beta_1^{(0)} \\ 0 & a_{2,2}^{(1)} & a_{2,3}^{(1)} & \cdots & a_{2,n}^{(1)} & \beta_2^{(1)} \\ 0 & a_{3,2}^{(1)} & a_{3,3}^{(1)} & \cdots & a_{3,n}^{(1)} & \beta_3^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n,2}^{(1)} & a_{n,3}^{(1)} & \cdots & a_{n,n}^{(1)} & \beta_n^{(1)} \end{pmatrix}, \text{满足} \begin{matrix} a_{i,j}^{(1)} = a_{i,j}^{(0)} - l_{i,1}a_{1.j}^{(0)} \\ \beta_i^{(1)} = \beta_i^{(0)} - l_{i,1}\beta_1^{(0)} \\ j = 2, 3, ..., n \\ i = 2, 3, ..., n \end{matrix}.$$

考虑第 k 步, 在完成第 $k-1$ 步的时候, 矩阵看上去就像

$$\begin{pmatrix} \mathbf{A}^{(k-1)} & \mathbf{b}^{(k-1)} \end{pmatrix} = \begin{pmatrix} a_{1,1}^{(0)} & \cdots & a_{1,k}^{(0)} & \cdots & a_{1,n}^{(0)} & \beta_1^{(0)} \\ & \ddots & \vdots & \ddots & \vdots & \vdots \\ & & a_{k,k}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & \beta_k^{(k-1)} \\ & & \vdots & \ddots & \vdots & \vdots \\ & & a_{n,k}^{(k-1)} & \cdots & a_{n,n}^{(k-1)} & \beta_n^{(k-1)} \end{pmatrix}$$

到了第 k 步, 我们令 $l_{i.k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$, 并且试图把 k, k 下面的列消去为 0, 其余列的系数就变成了 $\begin{matrix} a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - l_{i,k}a_{k,j}^{(k-1)} & \text{(未知数的系数)} \\ \beta_i^{(k)} = \beta_i^{(k-1)} - l_{i,k}\beta_k^{(k-1)} & \text{(常数项)} \end{matrix}$, 对于 $i, j = k+1, k+2, ..., n$. 经过这样一番操作, 就得到了第 k 步执行完了之后的样子:

$$\begin{pmatrix} \mathbf{A}^{(k)} & \mathbf{b}^{(k)} \end{pmatrix} = \begin{pmatrix} a_{1,1}^{(0)} & \cdots & a_{1,k}^{(0)} & a_{1,k+1}^{(0)} & \cdots & a_{1,n}^{(0)} & \beta_1^{(0)} \\ & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{k,k}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & \beta_k^{(k-1)} \\ & & 0 & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} & \beta_k^{(k)} \\ & & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & 0 & a_{n,k+1}^{(k)} & \cdots & a_{n,n}^{(k)} & \beta_n^{(k)} \end{pmatrix}.$$

实际上, 第 k 步的时候, 第 k 行的元素保持不变 (蓝色部分); 更新的区域是其右下方的一个小矩形, 如紫色部分所示.

在执行完了 $n-1$ 步消元之后, 便得到了

$$\begin{pmatrix} \mathbf{A}^{(n-1)} & \mathbf{b}^{(n-1)} \end{pmatrix} = \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & \beta_1^{(0)} \\ & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & \beta_2^{(1)} \\ & & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & \beta_3^{(2)} \\ & & & \ddots & \vdots & \vdots \\ & & & & a_{nn}^{(n-1)} & \beta_n^{(n-1)} \end{pmatrix}.$$

这时候便可以从 x_n 逐步回带, 依次求解. 如

$$\begin{cases} x_n = \frac{\beta_n^{(n-1)}}{a_{n,n}^{(n-1)}} \\ x_k = \frac{\beta_k^{(k-1)} - \sum_{j=k+1}^n a_{k,j}^{(k-1)} x_j}{a_{k,k}^{(k-1)}} & k = n-1, n-2, \cdots, 1 \end{cases}.$$

运算量估计 对于 Gauss 消元法, 所需要的乘除法总次数为

$$\begin{aligned} N_1 &= \sum_{k=1}^{n-1} ((n-k)(n-k+1) + n - k) \\ &= \sum_{k=1}^{n-1} (n-k)^2 + 2\sum_{k=1}^{n-1} (n-k) \\ &= \underbrace{\sum_{1 \le k \le n-1} k^2}_{k \leftarrow n-k} + 2 \underbrace{\sum_{1 \le k \le n-1} k}_{k \leftarrow n-k} \\ &= \frac{n(n-1)(2n-1)}{6} + 2\frac{n(n-1)}{2} \\ &= \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}. \end{aligned}$$

另外, 回带的时候,

- 求 x_n 需做 1 次除法; 共 1 次
- 求 x_{n-1} 需做 1 次乘法和 1 次除法; 共 2 次
- 求 x_{n-2} 需做 2 次乘法和 1 次除法; 共 3 次
- ... 如此往复
- 求 x_1 需做 $n-1$ 次乘法和 1 次除法; 共 n 次.

于是

$$N_2 = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

总运算次数为 $N = N_1 + N_2 = n^3/3 + n^2 - n/3$.

选主元减小误差 在每次消元运算中可能引入微小的误差. 这主要是因为在除法的时候不能除以一个比较小的数. 由此, 我们需要**每次消元前, 在本列元素中**

选绝对值最大的非零元作为主元 (pivot), 然后经过换行换到主对角线上, 再进行消元.

2.2. 使用计算器的附注. 使用 Fx-991 CN CW 计算器的时候, 可以通过主屏幕中的 "方程 → 线性方程组" 中找到对应的结果.

如果计算器支持矩阵运算, 在高斯消元回带过程中, 可以使用左乘一个初等矩阵的方法批量地把某一行乘上某个倍数加到另一行去.

初等矩阵

- 如果希望把 $\mathbf{A}$ 矩阵的第 i 行和第 j 列对调, 就在 $\mathbf{A}$ 左边乘上

$$\begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 0 & & 1 & \\ & & & \ddots & & \\ & & 1 & & 0 & \\ & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \text{第}i\text{行} \\ \\ \leftarrow \text{第}j\text{行} \\ \\ \end{matrix}$$

.

- 如果希望把 $\mathbf{A}$ 矩阵的第 i 行乘上 k 倍, 就在 $\mathbf{A}$ 的左边乘上

$$\begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & k & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \text{第}i\text{行} \\ \\ \\ \\ \end{matrix}$$

.

- 如果希望把 $\mathbf{A}$ 矩阵的第 i 行乘上 k 倍加到第 j 行去, 就在 $\mathbf{A}$ 的左边乘上

$$\begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ k & & 1 & & & \\ \uparrow & & & \ddots & & \\ \text{第}i\text{行}k\text{倍} & & & & 1 & \\ & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \text{第}j\text{行} \\ \\ \\ \\ \end{matrix}$$

上面的三个就叫做初等矩阵. 它捕捉了高斯消元基本的一部捕捉的运算. 总结一下, 初等矩阵有如下的性质

1. 设 $\mathbf{A}$ 是一个 $m \times n$ 的矩阵, 对 $\mathbf{A}$ 施行一次初等行变换, 相当于 $\mathbf{A}$ 左乘相应的 m 阶初等矩阵;
2. 初等矩阵都是可逆的. 这是因为做一个相反的操作就可以变回单位矩阵.

从初等矩阵的角度看 Gauss 消去法, 实际上就是

- 系数矩阵化成上三角矩阵, 实质上是做一系列初等行变换

- 相当于用一系列初等矩阵去左乘增广矩阵

例如, 第 k 步消元实际上左乘的是

$$\mathbf{L}_k = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & -l_{k+1,k} & 1 & & \\ & & & \vdots & \vdots & \ddots & \\ & & & -l_{n,k} & 0 & \cdots & 1 \end{pmatrix} \begin{array}{l} \\ \\ \\ \leftarrow 第k行 \\ \\ \\ \\ \end{array}$$

§3　三角分解法 (LU 分解)

观察上节得到的 L_k, 注意到他们是若干个初等矩阵的乘积, 因此均为非奇异矩阵. 从而可以容易地求出他们的逆矩阵 L_k^{-1}, 他们是

$$\mathbf{L}_k = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & \color{red}{l_{k+1,k}} & 1 & & \\ & & & \color{red}{\vdots} & \vdots & \ddots & \\ & & & \color{red}{l_{n,k}} & 0 & \cdots & 1 \end{pmatrix} \begin{array}{l} \\ \\ \\ \leftarrow 第k行 \\ \\ \\ \\ \end{array}$$

我们知道 $\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1 \underbrace{\left(\mathbf{A}^{(0)}, \mathbf{b}^{(0)}\right)}_{\text{初始的增广矩阵}} = \underbrace{\left(\mathbf{A}^{(n-1)}, \mathbf{b}^{(n-1)}\right)}_{\text{Gauss 消元法的结果}}$, 两边同时乘以他们的逆:

$$\begin{aligned} \color{red}{\mathbf{L}_{n-1}^{-1}\cdots\mathbf{L}_{n-2}^{-1}\mathbf{L}_{n-1}^{-1}}\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1(\mathbf{A}^{(0)}, \mathbf{b}^{(0)}) &= \color{red}{\mathbf{L}_{n-1}^{-1}\cdots\mathbf{L}_{n-2}^{-1}\mathbf{L}_{n-1}^{-1}}(\mathbf{A}^{(n-1)}, \mathbf{b}^{(n-1)}) \\ (\mathbf{A}^{(0)}, \mathbf{b}^{(0)}) &= \underbrace{\color{red}{\mathbf{L}_{n-1}^{-1}\cdots\mathbf{L}_{n-2}^{-1}\mathbf{L}_{n-1}^{-1}}}_{\text{下三角矩阵}}\underbrace{(\mathbf{A}^{(n-1)}, \mathbf{b}^{(n-1)})}_{\text{上三角矩阵}} \end{aligned}$$

实际上, 高斯法的消元过程实质上是把系数矩阵 $\mathbf{A}$ 分解成一个单位下三角阵 (Lower triangular matrix) 与上三角矩阵 (Upper triangular matrix) 乘积的过程, 即 $\mathbf{A} = \mathbf{LU}$. 当完成三角 (lower-upper) 分解后, 方程组 $\mathbf{Ax} = \mathbf{b}$ 的求解可以转化为:

$$\begin{cases} \mathbf{Ax} = \mathbf{b} \iff \mathbf{L}\color{blue}{\mathbf{Ux}} = \mathbf{b} \\ \mathbf{L}\color{blue}{\mathbf{y}} = \mathbf{b} \end{cases} \implies \begin{cases} \mathbf{Ly} = \mathbf{b} \\ \mathbf{Ux} = \mathbf{y} \end{cases}$$

而这是易于求解的. 因此, 解方程组的问题可转化为矩阵的三角分解 (LU 分解) 问题.

3.1. Dolittle 的 LU 分解法. 除了 Gauss 消元的 LU 分解法, 我们可以使用待定系数法来比较逐个元素来确定 $\mathbf{L},\mathbf{U}$ 矩阵每一个元素是什么.

考虑

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}}_{\mathbf{A}} = \underbrace{\begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & \\ l_{n1} & l_{n2} & l_{n3} & \cdot & 1 \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdot & \cdot & u_{1n} \\ & u_{22} & u_{23} & \cdot & \cdot & u_{2n} \\ & & u_{33} & \cdot & \cdot & u_{3n} \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot \\ & & & & & u_{nn} \end{pmatrix}}_{\mathbf{U}}$$

按照依赖顺序, 我们先行后列遍历 $\mathbf{A}$.

1. 依次确定矩阵 $\mathbf{U}$ 的第 1 行元素和矩阵 $\mathbf{L}$ 的第 1 列元素.
   (a) 由 $\mathbf{L}$ 的第 1 行和 $\mathbf{U}$ 的第 j 列元素相乘后与 $\mathbf{A}$ 的第一行对应元素相等, 得 $\mathbf{U}$ 的第一行元素 $u_{1,j} = a_{1,j}(j = 1, 2, ..., n)$.
   (b) 由 $\mathbf{L}$ 的第 i 行 $(i > 1)$ 和 $\mathbf{U}$ 的第 1 列元素相乘后与 $\mathbf{A}$ 的第一列对应元素相等有 $a_{i,1} = l_{i,1} \times u_{1,1}$ 而得 $\mathbf{L}$ 的第 1 列元素 $l_{i,1} = a_{i1}/u_{11}(i = 2, ..., n)$.
2. 依次确定矩阵 $\mathbf{U}$ 的第 2 行元素和矩阵 $\mathbf{L}$ 的第 2 列元素
   (a) 由 $\mathbf{L}$ 的第 2 行和 $\mathbf{U}$ 的第 j 列元素相乘后与 $\mathbf{A}$ 的第二行对应元素相等, 得 $a_{2j} = l_{21}u_{1j} + u_{2j}$, 得 $\mathbf{U}$ 的第二行元素 $u_{2j} = a_{2j} - l_{21}u_{1j}(j = 2, ..., n)$. (红色是已知元素)
   (b) 由 $\mathbf{L}$ 的第 i 行 $(i > 2)$ 和 $\mathbf{U}$ 的第 2 列元素相乘后与 $\mathbf{A}$ 的第二列对应元素相等有 $a_{i2} = l_{i1} \times u_{12} + l_{i2} \times u_{22}$, 从而得到 $\mathbf{L}$ 的第 2 列元素 $l_{i2} = (a_{i2} - l_{i1} \times u_{12})/u_{22}(i = 3, ..., n)$.

继续这样下去, 直到第 k 步:

k**.** 依次确定矩阵 $\mathbf{U}$ 的第 k 行元素和矩阵 $\mathbf{L}$ 的第 k 列元素.

1. 由 $\mathbf{A}$ 第 k 行的元素

$$a_{kj} = \sum_{r=1}^{k-1} l_{kr}u_{rj} + u_{kj}(k = j, \cdots, n)$$

   可以得到 $\mathbf{U}$ 的第 k 行元素

$$u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr}u_{ri}, (j = k, \cdots, n).$$

2. 由 $\mathbf{A}$ 第 k 列的元素

$$a_{ik} = \sum_{r=1}^{k-1} l_{ir}u_{rk} + l_{ik}u_{kk}(k = k+1, \cdots, n)$$

可以得到 $\mathbf{L}$ 的第 k 列元素

$$l_{ik} = \left(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk} \right) / u_{kk}, (j = k+1, \cdots, n).$$

我们采取如下的紧凑格式来记忆.

例 2. 使用 Dolittle 三角分解法解答线性方程组

$$\begin{pmatrix} 2 & 1 & 5 \\ 4 & 1 & 12 \\ -2 & -4 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 11 \\ 27 \\ 12 \end{pmatrix}.$$

采用紧凑格式, 将 $\mathbf{L}, \mathbf{U}$ 放在同一个矩阵里; 并且增广矩阵的结果用括号表示, 就有

$$\begin{pmatrix} (2) & (1) & (5) & (11) \\ (4) & (1) & (12) & (27) \\ (-2) & (-4) & (5) & (12) \end{pmatrix}$$

第一步: 计算第一行和第一列的值:

$$\begin{pmatrix} (2):2 & (1):1 & (5):5 & (11):11 \\ (4):4/2=2 & (1) & (12) & (27) \\ (-2):(-2)/2=-1 & (-4) & (5) & (12) \end{pmatrix}$$

第二步: 计算第二行和第二列的值

$$\begin{pmatrix} (2):2 & (1):1 & (5):5 & (11):11 \\ (4):2 & (1):1-1\times 2 = {\color{red}-1} & (12):12-5\times 2 = 2 & (27):27-11\times 2 = 5 \\ (-2):-1 & (-4):(-4-1\times(-1))/{\color{red}(-1)} = 3 & (5) & (12) \end{pmatrix}$$

第三步: 计算第三行和第三列的值

$$\begin{pmatrix} (2):2 & (1):1 & (5):5 & (11):11 \\ (4):2 & (1):-1 & (12):2 & (27):5 \\ (-2):-1 & (-4):3 & (5):5-5\times(-1)-2\times 3 = 4 & (12):12-11\times(-1)-5\times 3 = 8 \end{pmatrix}$$

于是分解为

$$\begin{pmatrix} (2):2 & (1):1 & (5):5 & (11):11 \\ (4):2 & (1):-1 & (12):2 & (27):5 \\ (-2):-1 & (-4):3 & (5):4 & (12):8 \end{pmatrix}$$

即

$$\mathbf{A} = \mathbf{LU} = \begin{pmatrix} 1 & & \\ 2 & 1 & \\ -1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 5 \\ & -1 & 2 \\ & & 4 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 11 \\ 5 \\ 8 \end{pmatrix}$$

只要 $\mathbf{Ly} = \mathbf{b}, \mathbf{Ux} = \mathbf{y}$ 即可.

§4 解线性方程组的迭代解法

与解 $f(x) = 0$ 的不动点迭代相似, 将方程组 $\mathbf{Ax} = \mathbf{b}$ 等价改写成 $\mathbf{x} = \mathbf{Bx} + \mathbf{f}$ 的形式, 建立迭代格式 $\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \mathbf{f}$. 从 $\mathbf{x}_0$ 出发, 生成迭代序列 $\{\mathbf{x}^{(k)}\}$, 直至收敛. 其优点是程序简单, 存储量小、以及特别适用于求解系数矩阵为大型稀疏矩阵的方程组.

下面考虑迭代公式的构造. 对于一般迭代法的一般迭代格式, 任给 $m+1$ 个初始向量 $\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}, ..., \mathbf{x}^{(k-m)}$, 代入公式 $\mathbf{x}^{(k+1)} = F_k(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}, \cdots, \mathbf{x}^{(k-m)}); (k = 0, 1, \cdots)$. 这里, 第 $k+1$ 步与前 $m+1$ 步都有关, 称之为多步迭代法. 倘若 $m = 0$, 原式退化为 $\mathbf{x}^{(k+1)} = F_k(\mathbf{x}^{(k)}); k = 0, 1, \cdots$, 称之为单步迭代法. 如果 F_k 是线性的, 称之为单步线性迭代法, 即 $\mathbf{x}^{(k+1)} = \mathbf{B}_k\mathbf{x}^{(k)} + \mathbf{f}_k; k = 0, 1, \cdots$. 其中 $\mathbf{B}_k$ 是迭代矩阵. 若 F_k 与 k 无关 (B_k、f_k 与 k 无关), 称为单步定常线性迭代法: $\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \mathbf{f}; k = 0, 1, \cdots$.

以上是迭代法的简单分类. 对于解线性方程组, 我们说:

设 $\mathbf{A} \in \mathbb{R}^{n\times n}$,$\mathbf{b} \in \mathbb{R}^n, \det(\mathbf{A}) \neq 0$. 如果方程组

$$\mathbf{Ax} = \mathbf{b} \iff \mathbf{x} = \mathbf{Bx} + \mathbf{f}$$

, 则可以构造单步单步定常线性迭代法 $\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \mathbf{f}$.

当迭代公式产生的序列 $\{x^{(l)}\}_{k=0}^{\infty}$ 收敛到方程组的解 x^* 时, 即 $\lim_{k\to\infty} x^{(k)} = x^*$ 时, 称该迭代法是收敛的.

什么时候迭代的结果是收敛的? 实际上有如下定理:

定理 1. *如下三个命题是等价的*

1. *迭代法* $\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \mathbf{f}$. *收敛;*
2. $\rho(\mathbf{B}) < 1$.
3. *至少存在一种从属矩阵范数* $||\cdot||$, *使得* $||\mathbf{B}|| < 1$.

矩阵的范数实际上衡量的是矩阵作为线性变换的拉伸程度. 实际上其拉伸程度只要小于 1, 就满足了上一章的压缩性质. 我们不予证明这三个条件.

4.1. Jacobi 迭代法. Jacobi 认为, 解答线性方程组可以用这样的迭代法:

例 3. 假设要求

$$\begin{aligned} 4x - y + z &= 7 \\ 4x - 8y + z &= -21 \\ -2x + y + 5z &= 15 \end{aligned}$$

我们把 x, y, z 反解出来得到:

$$
\begin{aligned}
x &= \frac{7+y-z}{4} \\
y &= \frac{21+4x+z}{8} \\
z &= \frac{15+2x-y}{5}
\end{aligned}
$$

于是我们考虑这样的迭代过程

$$
\begin{aligned}
x_{k+1} &= \frac{7+y_k-z_k}{4} \\
y_{k+1} &= \frac{21+4x_k+z_k}{8} \\
z_{k+1} &= \frac{15+2x_k-y_k}{5}
\end{aligned}
$$

从一个初值开始然后不断迭代即可达到新的值.

用线性代数的观点表示, 设方程组 $\mathbf{Ax}=\mathbf{b}$, 且 $\det(\mathbf{A}) \neq 0$. 记 $\mathbf{A}$ 的严格上三角部分为 $-\mathbf{L}$, 严格下三角的部分为 $-\mathbf{U}$(这里的负号是为了方便后续化简). 我们可以把原来的矩阵变化为 $\mathbf{A}=\mathbf{D}-\mathbf{L}-\mathbf{U}$. 其中, $\mathbf{D}=\mathrm{diag}(a_{11}, a_{22}, \cdots, a_{nn})$. 我们可以构造 $\mathbf{Dx}=(\mathbf{L}+\mathbf{U})\mathbf{x}+\mathbf{b}$.

如果 $a_{ii} \neq 0 (i=1,2,\cdots,n)$, 原方程组可化为 $\mathbf{x}=\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\mathbf{x}+\mathbf{D}^{-1}\mathbf{b}=\mathbf{Bx}+\mathbf{f}$. 这就是 Jacobi 迭代法.

4.2. Gauss-Seidel 迭代法. 在 Jacobi 迭代公式中, 计算 $x_i^{(k+1)}$ 时, 利用已经算出来的新的 $x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$ 值, 从而得到 Gauss-Seidel 迭代法. 即上例子的迭代公式变为

$$
x_{k+1}=\frac{7+y_k-z_k}{4}; y_{k+1}=\frac{21+4x_{k+1}+z_k}{8}; z_{k+1}=\frac{15+2x_{k+1}-y_{k+1}}{5}.
$$

对于这个迭代法而言, 假设线性方程组 $\mathbf{Ax}=\mathbf{b}$ 的系数矩阵 $\mathbf{A}$ 是严格对角占优的 (强对角优势矩阵), 则 Jacobi 迭代法和 Gauss-Seidel 迭代法的计算结果会收敛. (强对角优势矩阵（不可约对角优势矩阵）是非奇异方阵).

4.3. 使用计算器的附注. 若计算器具有矩阵计算功能, 将上述的内容表示为线性迭代的形式之后 (对于 Gauss-Seidel 方法, 把 x_{k+1} 带入即可) , 便可以采用如下的迭代算法, 其原理如下:

在计算完成后, 答案会储存在 `MatAns` 寄存器中.

于是第一次, 需要计算 $\underbrace{\texttt{MatA}}_{\mathbf{B}} \times \underbrace{\texttt{MatB}}_{\mathbf{x}} + \underbrace{\texttt{MatC}}_{\mathbf{f}}$; 之后的每一次迭代只要输入 $\underbrace{\texttt{MatA}}_{\mathbf{B}} \times \underbrace{\texttt{MatAns}}_{\mathbf{x}} + \underbrace{\texttt{MatC}}_{\mathbf{f}}$ 即可.