第二讲　线性方程组的直接解法

一般来说, 求解线性方程组的数值方法可以分为两类: 直接法与迭代法. 本章介绍直接法, 即 Gauss 消去法. 直接法相对比较稳定, 因此在工程计算中很受欢迎. 但由于运算量是 $\mathcal{O}(n^3)$, 当问题规模较大时, 时间会很长 (这里 n 表示未知量的个数). 目前, 直接法主要用于小规模或中等规模线性方程组的数值求解.

2.1 Gauss 消去法和 LU 分解

2.1.1 LU 分解

考虑线性方程组

$$Ax = b, \tag{2.1}$$

其中 $A \in \mathbb{R}^{n\times n}$ 非奇异, $b \in \mathbb{R}^n$ 为给定的右端项. Gauss 消去法本质上就是对系数矩阵 A 进行 LU 分解, 即将 A 分解成两个矩阵的乘积

$$A = LU, \tag{2.2}$$

其中 L 是单位下三角矩阵, U 为非奇异上三角矩阵. 这个分解就称为 **LU 分解**.

假定矩阵 A 存在 LU 分解 (2.2), 则方程组 (2.1) 就转化为求解下面两个三角方程组

$$\begin{cases} Ly = b, \\ Ux = y. \end{cases}$$

显然, 这两个方程组都非常容易求解.

基于 LU 分解的 Gauss 消去法描述如下:

算法 2.1. Gauss 消去法

1: 将 A 进行 LU 分解: $A = LU$, 其中 L 为单位下三角矩阵, U 为非奇异上三角矩阵;
2: 利用向前回代, 求解 $Ly = b$, 即得 $y = L^{-1}b$;
3: 利用向后回代, 求解 $Ux = y$, 即得 $x = U^{-1}y = (LU)^{-1}b = A^{-1}b$.

我们知道, 当系数矩阵 A 非奇异时, 方程组 (2.1) 总是存在唯一解. 但是, 并不是每个非奇异矩阵都存在 LU 分解.

定理 2.1 (LU 分解的存在性和唯一性) 设 $A \in \mathbb{R}^{n\times n}$. 则存在唯一的单位下三角矩阵 L 和非奇异上三角矩阵 U, 使得 $A = LU$ 的充要条件是 A 的所有顺序主子矩阵 $A_k = A(1:k, 1:k)$ 都非奇异, $k = 1, 2, \ldots, n$.

证明. 必要性: 设 A_{11} 是 A 的 k 阶顺序主子矩阵, 将 $A = LU$ 写成分块形式

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}.$$

可得 $A_{11} = L_{11}U_{11}$. 由于 L_{11} 和 U_{11} 均非奇异, 所以 A_{11} 也非奇异.

充分性: 用归纳法.

当 $n = 1$ 时, 结论显然成立.

假设结论对 $n-1$ 阶矩阵都成立, 即对任意 $n-1$ 阶矩阵, 如果其所有的顺序主子矩阵都非奇异, 则存在 LU 分解.

考虑 n 阶的矩阵 A, 写成分块形式

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $A_{11} \in \mathbb{R}^{(n-1)\times(n-1)}$ 是 A 的 $n-1$ 阶顺序主子矩阵. 由归纳假设可知, A_{11} 存在 LU 分解, 即存在单位下三角矩阵 L_{11} 和非奇异上三角矩阵 U_{11} 使得

$$A_{11} = L_{11}U_{11}.$$

令

$$L_{21} = A_{21}U_{11}^{-1}, \quad U_{12} = L_{11}^{-1}A_{12}, U_{22} = A_{22} - L_{21}U_{12},$$

则

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & U_{22} + L_{21}U_{12} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \triangleq LU.$$

易知 U 非奇异, 所以 A 存在 LU 分解.

下面证明唯一性. 设 A 存在两个不同的 LU 分解:

$$A = LU = \tilde{L}\tilde{U},$$

其中 L 和 $\tilde{L}$ 为单位下三角矩阵, U 和 $\tilde{U}$ 为非奇异上三角矩阵. 则有

$$L^{-1}\tilde{L} = U\tilde{U}^{-1},$$

该等式左边为下三角矩阵, 右边为上三角矩阵, 所以只能是对角矩阵. 由于单位下三角矩阵的逆仍然是单位下三角矩阵, 所以 $L^{-1}\tilde{L}$ 的对角线元素全是 1, 故

$$L^{-1}\tilde{L} = I,$$

即 $\tilde{L} = L, \tilde{U} = U$.

由归纳法可知, 结论成立. □

2.1.2 LU 分解的实现

给定一个矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

我们可以通过矩阵初等变换来构造 A 的 LU 分解.

- 第一步: 假定 $a_{11} \neq 0$, 构造矩阵

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad \text{其中} \quad l_{i1} = \frac{a_{i1}}{a_{11}}, i = 2, 3, \ldots, n.$$

易知 L_1 的逆为

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ -l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

用 L_1^{-1} 左乘 A, 并将所得到的矩阵记为 $A^{(1)}$, 则

$$A^{(1)} = L_1^{-1} A \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}.$$

即左乘 L_1^{-1} 后, A 的第一列中除第一个元素外其它都变为 0.

- 第二步: 类似地, 我们可以将上面的操作作用在 $A^{(1)}$ 的子矩阵 $A^{(1)}(2:n, 2:n)$ 上, 将其第一列除第一个元素外都变为 0. 也就是说, 假定 $a_{22}^{(1)} \neq 0$, 构造矩阵

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & l_{n2} & 0 & \cdots & 1 \end{bmatrix}, \quad 其中 \quad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, i = 3, 4, \ldots, n.$$

用 L_2^{-1} 左乘 $A^{(1)}$, 并将所得到的矩阵记为 $A^{(2)}$, 则

$$A^{(2)} = L_2^{-1} A = L_2^{-1} L_1^{-1} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

- 依此类推, 假定 $a_{kk}^{(k-1)} \neq 0$ $(k = 3, 4, \ldots, n-1)$, 则我们可以构造一系列的矩阵 $L_3, L_4, \ldots, L_{n-1}$, 使得

$$L_{n-1}^{-1} \cdots L_2^{-1} L_1^{-1} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{bmatrix}$$

为一个上三角矩阵. 我们将这个上三角矩阵记为 U, 并记

$$L = L_1 L_2 \cdots L_{n-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}, \tag{2.3}$$

则可得

$$A = LU,$$

这就是 A 的 LU 分解.

将上面的过程写成算法, 描述如下:

算法 2.2. LU 分解

Set $L = I, U = 0$ % 将 L 设为单位矩阵, U 设为零矩阵
for $k = 1$ to $n - 1$ **do**
  for $i = k + 1$ to n **do**
    $l_{ik} = a_{ik}/a_{kk}$ % 计算 L 的第 k 列
  end for
  for $j = k$ to n **do**
    $u_{kj} = a_{kj}$ % 计算 U 的第 k 行
  end for
  for $i = k + 1$ to n **do**
    for $j = k + 1$ to n **do**
      $a_{ij} = a_{ij} - l_{ik}u_{kj}$ % 更新 $A(k + 1 : n, k + 1 : n)$
    end for
  end for
end for

Gauss 消去法的运算量

由算法 2.2 可知, LU 分解的运算量 (含加减乘除) 为

$$\sum_{i=1}^{n-1}\left(\sum_{j=i+1}^{n} 1 + \sum_{j=i+1}^{n}\sum_{k=i+1}^{n} 2\right) = \sum_{i=1}^{n-1}\left(n - i + 2(n - i)^2\right) = \frac{2}{3}n^3 + O(n^2).$$

由于回代过程的运算量为 $O(n^2)$, 所以 Gauss 消去法的总运算量为 $\frac{2}{3}n^3 + O(n^2)$.

☞ 评价算法的一个主要标准是执行时间, 但这依赖于计算机硬件和编程技巧等, 因此直接给出算法执行时间是不太现实的. 所以我们通常是统计算法中算术运算 (加减乘除) 的次数. 在数值算法中, 大多仅仅涉及加减乘除和开方运算. 一般地, 加减运算次数与乘法运算次数具有相同的量级, 而除法运算和开方运算次数具有更低的量级.

☞ 为了尽可能地减少运算量, 在实际计算中, 数, 向量和矩阵做乘法运算时的先后执行次序为: 先计算数与向量的乘法, 然后计算矩阵与向量的乘法, 最后才计算矩阵与矩阵的乘法.

矩阵 L 和 U 的存储

当 A 的第 i 列被用于计算 L 的第 i 列后, 在后面的计算中不再被使用. 同样地, A 的第 i 行被用于计算 U 的第 i 行后, 在后面的计算中也不再被使用. 因此, 为了节省存储空间, 我们可以在计算过程中将 L 的第 i 列存放在 A 的第 i 列, 将 U 的第 i 行存放在 A 的第 i 行, 这样就不需要另外分配空间存储 L 和 U. 计算结束后, A 的上三角部分为 U, 其绝对下三角部分为 L 的绝对下三角部分 (L 的对角线全部为 1, 不需要存储). 此时算法可以描述为：

算法 2.3. LU 分解

for $k = 1$ to $n - 1$ **do**

    for $i = k+1$ to n **do**
        $a_{ik} = a_{ik}/a_{kk}$
        for $j = k+1$ to n **do**
            $a_{ij} = a_{ij} - a_{ik}a_{kj}$
        end for
    end for
end for

☞ 根据指标的循环次序, 算法 2.3 也称为 KIJ 型 LU 分解. 在实际计算中, 我们一般不建议使用这个算法. 因为对于指标 k 的每次循环, 都需要更新 A 的第 $k+1$ 至第 n 行. 这种反复读取数据的做法会使得计算效率大大降低.

对于按行存储的数据结构, 我们一般采用后面介绍的 IKJ 型 LU 分解.

MATLAB 源代码 2.1. LU 分解的 MATLAB 代码 (KIJ 型)

```
function A = mylu(A)
n=size(A,1);
for k=1:n-1
  if A(k,k) == 0
    fprintf('Error: A(%d,%d)=0!\n', k, k);
    return;
  end
  for i=k+1:n
    A(i,k)=A(i,k)/A(k,k);
    for j=k+1:n
      A(i,j)=A(i,j)-A(i,k)*A(k,j);
    end
  end
end
```

为了充分利用 MATLAB 的向量运算优势, 提高运算效率, 上面的程序可改写为

MATLAB 源代码 2.2. LU 分解 (KIJ 型)

```
function A = mylu(A)
n=size(A,1);
for k=1:n-1
  if A(k,k) == 0
    fprintf('Error: A(%d,%d)=0!\n', k, k);
    return;
  end
  A(k+1:n,k)=A(k+1:n,k)/A(k,k);
  A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-A(k+1:n,k)*A(k,k+1:n);
end
```

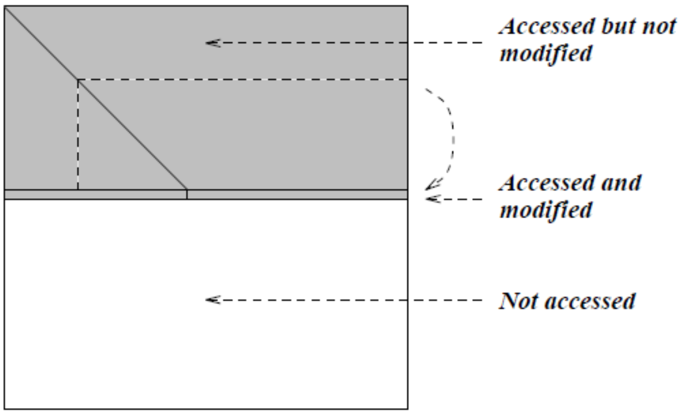
2.1.3 IKJ 型 LU 分解

如果数据是按行存储的, 如 C/C++, 我们一般采用下面的 IKJ 型 LU 分解.

算法 2.4. LU 分解 (IKJ 型)

for $i = 2$ to n **do**
    for $k = 1$ to $i - 1$ **do**
        $a_{ik} = a_{ik}/a_{kk}$
        for $j = k + 1$ to n **do**
            $a_{ij} = a_{ij} - a_{ik}a_{kj}$
        end for
    end for
end for

上述算法可以用下图来描述.



♣ 如果数据是按列存储的, 如 FORTRAN 或 MATLAB, 则该怎样设计算法?

2.1.4 待定系数法计算 LU 分解

我们也可以利用待定系数法来实现矩阵的 LU 分解. 假设 A 存在 LU 分解, 即 $A = LU$, 或

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{2n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix}.$$

通过比较等式两边的元素来计算 L 和 U 中的各元素的值. 具体计算过程如下:

(1) 比较等式两边的第一行, 可得

$$u_{1j} = a_{1j}, \quad j = 1, 2, \ldots, n.$$

再比较等式两边的第一列, 可得

$$a_{i1} = l_{i1}u_{11} \Rightarrow l_{i1} = a_{i1}/u_{11}, \quad i = 2, 3, \ldots, n.$$

(2) 比较等式两边的第二行, 可得

$$a_{2j} = l_{21}u_{1j} \;\Rightarrow\; u_{2j} = a_{2j} - l_{21}u_{ij}, \quad j = 2, 3, \ldots, n.$$

再比较等式两边的第二列, 可得

$$a_{i2} = l_{i1}u_{12} + l_{i2}u_{22} \;\Rightarrow\; l_{i1} = (a_{i2} - l_{i1}u_{12})/u_{22}, \quad i = 3, 4, \ldots, n.$$

(3) 以此类推, 第 k 步时, 比较等式两边的第 k 行, 可得

$$u_{kj} = a_{kj} - (l_{k1}u_{1j} + \cdots + l_{k,k-1}u_{k-1,j}), \quad j = k, k+1, \ldots, n.$$

比较等式两边的第 k 列, 可得

$$l_{ik} = (a_{ik} - l_{i1}u_{1k} - \cdots - l_{i,k-1}u_{k-1,k})/u_{kk}, \quad i = k+1, k+2, \ldots, n.$$

直到第 n 步, 即可计算出 L 和 U 的所有元素.

同样, 我们可以利用 A 来存储 L 和 U. 算法描述如下:

算法 2.5. LU 分解 (待定系数法或 Doolittle 方法)

for $k = 1$ to n **do**

$\quad a_{kj} = a_{kj} - \sum_{i=1}^{k-1} a_{ki}a_{ij}, \quad j = k, k+1, \ldots, n$

$\quad a_{ik} = \dfrac{1}{a_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} a_{ij}a_{jk} \right), \quad i = k+1, k+2, \ldots, n$

end for

相应的 MATLAB 程序为:

MATLAB 源代码 2.3. 待定系数法 LU 分解

```matlab
function A = mylu2(A)
[n,n]=size(A);
for k=1:n
  A(k,k)=A(k,k)-A(k,1:k-1)*A(1:k-1,k);
  if (A(k,k)==0)
    fprintf('Error: A(%d,%d)=0!\n', i,i); return;
  end
  A(k,k+1:n)=A(k,k+1:n)-A(k,1:k-1)*A(1:k-1,k+1:n);
  A(k+1:n,k)=(A(k+1:n,k)-A(k+1:n,1:k-1)*A(1:k-1,k))/A(k,k);
end
```

2.1.5 三角方程求解

得到 A 的 LU 分解后, 我们最后需要用回代法求解两个三角方程组

$$Ly = b, \quad Ux = y,$$

其中 L 是单位下三角矩阵, U 为非奇异上三角矩阵.

下面是关于一般下三角方程组的求解算法 (行存储方式).

算法 2.6. 向前回代求解 $Ly = b$ (行存储方式)

$y_1 = b_1/l_{11}$
for $i = 2 : n$ **do**
    for $j = 1 : i-1$ **do**
        $b_i = b_i - l_{ij}y_j$
    end for
    $y_i = b_i/l_{ii}$
end for

如果数据是按列存储的, 则采用列存储方式效率会高一些. 下面是按列存储方式求解上三角方程组.

算法 2.7. 向后回代求解 $Ux = y$ (列存储方式)

$x_n = y_n/u_{nn}$
for $i = n-1 : -1 : 1$ **do**
    for $j = 1 : i$ **do**
        $y_j = y_j - x_{i+1}u_{i+1,j}$
    end for
    $x_i = y_i/u_{ii}$
end for

☞ 这两个算法的运算量均为 $n^2 + \mathcal{O}(n)$.

以上两个算法都是向后稳定的 (componentwise backward stable) [35].

2.1.6 选主元 LU 分解

在 LU 分解算法 2.2 中, 我们称 $a_{kk}^{(k-1)}$ 为主元. 如果 $a_{kk}^{(k-1)} = 0$, 则算法就无法进行下去. 即使 $a_{kk}^{(k-1)}$ 不为零, 但如果 $|a_{kk}^{(k-1)}|$ 的值很小, 由于舍入误差的原因, 也可能会给计算结果带来很大的误差. 此时我们就需要通过选主元来解决这个问题.

例 2.1 用 LU 分解求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0.02 & 61.3 \\ 3.43 & -8.5 \end{bmatrix}$, $b = \begin{bmatrix} 61.5 \\ 25.8 \end{bmatrix}$, 要求在运算过程中保留 3 位有效数字.

解. 根据 LU 分解算法 2.2, 我们可得

$$
\begin{aligned}
& l_{11} = 1.00,\ l_{21} = a_{21}/a_{11} = 1.72 \times 10^2,\ l_{22} = 1.00, \\
& u_{11} = a_{11} = 2.00 \times 10^{-2},\ u_{12} = a_{12} = 6.13 \times 10, \\
& u_{22} = a_{22} - l_{21}u_{12} \approx -8.5 - 1.05 \times 10^4 \approx -1.05 \times 10^4,
\end{aligned}
$$

即

$$
A \approx \begin{bmatrix} 1.00 & 0 \\ 1.72 \times 10^2 & 1.00 \end{bmatrix} \begin{bmatrix} 2.00 \times 10^{-2} & 6.12 \times 10 \\ 0 & -1.05 \times 10^4 \end{bmatrix}.
$$

解方程组 $Ly=b$ 可得

$$y_1 = 6.15\times 10,\quad y_2 = b_2 - l_{21}y_1 \approx -1.06\times 10^4.$$

解方程组 $Ux=y$ 可得

$$x_2 = y_2/u_{22} \approx 1.01,\quad x_1 = (y_1 - u_{12} * x_2)/u_{11} \approx -0.413/u_{11} \approx -20.7$$

□

易知, 方程的精确解为 $x_1 = 10.0$ 和 $x_2 = 1.00$. 我们发现 x_1 的误差非常大. 导致这个问题的原因就是 $|a_{11}|$ 太小, 用它做主元时会放大舍入误差. 所以我们需要通过置换矩阵来选主元.

首先介绍置换矩阵的一些基本性质.

引理 2.1 设 $P\in\mathbb{R}^{n\times n}$ 为置换矩阵, $X\in\mathbb{R}^{n\times n}$ 为任意矩阵, 则

(1) PX 相当于将 X 的行进行置换; XP 相当于将 X 的列进行置换;

(2) $P^{-1}=P^{\intercal}$, 即 P 是正交矩阵;

(3) $\det(P)=\pm 1$;

(4) 置换矩阵的乘积仍然是置换矩阵.

定理 2.2 (选主元 LU 分解的存在性) 设 $A\in\mathbb{R}^{n\times n}$ 非奇异, 则存在置换矩阵 P_1, P_2, 以及单位下三角矩阵 L 和非奇异上三角矩阵 U, 使得 $P_1AP_2 = LU$. 其中 P_1 和 P_2 中只有一个是必需的.

证明. 用归纳法.

当 $n=1$ 时, 取 $P_1=P_2=L=1, U=A$ 即可.

假设结论对 $n-1$ 成立.

设 $A\in\mathbb{R}^{n\times n}$ 是 n 阶非奇异矩阵, 则 A 至少存在一个非零元, 取置换矩阵 $\hat{P}_1$ 和 $\hat{P}_2$ 使得

$$\hat{P}_1A\hat{P}_2 = \begin{bmatrix} a_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix},$$

其中 $a_{11}\neq 0$, $A_{22}\in\mathbb{R}^{(n-1)\times(n-1)}$.

$$u_{11}=a_{11},\quad U_{12}=A_{12},\quad L_{21}=A_{21}/a_{11},\quad U_{22}=A_{22}-L_{21}U_{12}.$$

则有

$$\begin{bmatrix} 1 & 0\\ L_{21} & I\end{bmatrix}\begin{bmatrix} u_{11} & U_{12}\\ 0 & U_{22}\end{bmatrix} = \begin{bmatrix} a_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix} = \hat{P}_1A\hat{P}_2.$$

两边取行列式可得

$$0\neq \det(P_1A\hat{P}_2) = \det\left(\begin{bmatrix} 1 & 0\\ L_{21} & I\end{bmatrix}\right)\cdot\det\left(\begin{bmatrix} u_{11} & U_{12}\\ 0 & U_{22}\end{bmatrix}\right) = a_{11}\det(U_{22}).$$

所以 $\det(U_{22})\neq 0$, 即 $U_{22}\in\mathbb{R}^{(n-1)\times(n-1)}$ 非奇异. 由归纳假设可知, 存在置换矩阵 $\tilde{P}_1$ 和 $\tilde{P}_2$ 使得

$$\tilde{P}_1U_{22}\tilde{P}_2 = \tilde{L}_{22}\tilde{U}_{22},$$

其中 $\tilde{L}_{22}$ 为单位下三角矩阵, $\tilde{U}_{22}$ 为非奇异上三角矩阵. 取

$$P_1 = \begin{bmatrix} 1 & 0\\ 0 & \tilde{P}_1\end{bmatrix}\hat{P}_1,\quad P_2 = \hat{P}_2\begin{bmatrix} 1 & 0\\ 0 & \tilde{P}_2\end{bmatrix},$$

则有

$$
\begin{aligned}
P_1AP_2 &= \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \tilde{P}_1^{\intercal} \tilde{L}_{22} \tilde{U}_{22} \tilde{P}_2^{\intercal} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1^{\intercal} \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \tilde{U}_{22} \tilde{P}_2^{\intercal} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \tilde{P}_2 \\ 0 & \tilde{U}_{22} \end{bmatrix} \\
&\triangleq LU,
\end{aligned}
$$

其中 L 为单位下三角矩阵, U 为非奇异上三角矩阵.

由归纳法可知, 结论成立. □

第 k 步时, 如何选取置换矩阵 P_1 和 P_2?

1. 选取 P_1 和 P_2 使得主元为剩下的矩阵中绝对值最大, 这种选取方法称为 "全主元 Gauss 消去法", 简称 GECP (Gaussian elimination with complete pivoting);
2. 选取 P_1 和 P_2 使得主元为第 k 列中第 k 到第 n 个元素中, 绝对值最大, 这种选取方法称为 "部分选主元 Gauss 消去法", 简称 GEPP (Gaussian elimination with partial pivoting), 此时 $P_2 = I$, 因此也称为列主元 Gauss 消去法.

- GECP 比 GEPP 更稳定, 但工作量太大, 在实际应用中通常使用 GEPP 算法.
- GEPP 算法能保证 L 所有的元素的绝对值都不超过 1.

算法 2.8. 部分选主元 LU 分解

$p = 1 : n;$ % 用于记录置换矩阵
for $i = 1$ to $n - 1$ **do**
  $a_{ki} = \max_{i \le j \le n} |a_{ji}|$ % 选列主元
  if $k \sim= i$ **then**
    for $j = 1$ to n **do**
      $t = a_{ij}$
      $a_{ij} = a_{kj}$
      $a_{kj} = t$ % 交换 A 的第 i 行与第 k 行
    end for
    $p(k) = i$
    $p(i) = k$ % 更新置换矩阵
  end if
  for $j = i + 1$ to n **do**
    $a_{ji} = a_{ji}/a_{ii}$ % 计算 L 的第 i 列
  end for
  for $j = i + 1$ to n **do**
    for $k = i + 1$ to n **do**
      $a_{jk} = a_{jk} - a_{ji} * a_{ik}$ % 更新 $A(i+1:n, i+1:n)$

```
            end for
        end for
    end for
```

相应的 MATLAB 程序如下:

MATLAB **源代码** 2.4. 部分选主元 LU 分解

```
function [A,p] = myplu(A)
[n,n]=size(A);
p=1:n;
for i=1:n-1
    [a,k]=max(abs(A(i:n,i)));
    if a==0
        error('Error: 第 %d 步的列主元为 0!\n', i);
    end
    k=k+i-1;
    if k~=i
        Atmp=A(i,:); A(i,:)=A(k,:); A(k,:)=Atmp;
        ptmp=p(i); p(i)=p(k); p(k)=ptmp;
    end
    A(i+1:n,i)=A(i+1:n,i)/A(i,i);
    A(i+1:n,i+1:n)=A(i+1:n,i+1:n)-A(i+1:n,i)*A(i,i+1:n);
end
```

例 2.2 用部分选主元 LU 分解求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0.02 & 61.3 \\ 3.43 & -8.5 \end{bmatrix}$, $b = \begin{bmatrix} 61.5 \\ 25.8 \end{bmatrix}$, 要求在运算过程中保留 3 位有效数字.

解. 由于 $|a_{21}| > |a_{11}|$, 根据部分选主元 LU 分解算法, 我们需要将第一行与第二行交换, 即取 $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, 然后计算 $\tilde{A} = PA$ 的 LU 分解, 可得

$$
\begin{aligned}
& l_{11} = 1.00,\ l_{21} = \tilde{a}_{21}/\tilde{a}_{11} = 5.83 \times 10^{-3},\ l_{22} = 1.00, \\
& u_{11} = \tilde{a}_{11} = 3.43,\ u_{12} = \tilde{a}_{12} = -8.50, \\
& u_{22} = \tilde{a}_{22} - l_{21}u_{12} \approx 6.13 \times 10 + 4.96 \times 10^{-2} \approx 6.13 \times 10,
\end{aligned}
$$

即

$$
PA \approx \begin{bmatrix} 1.00 & 0 \\ 5.83 \times 10^{-3} & 1.00 \end{bmatrix} \begin{bmatrix} 3.43 & -8.50 \\ 0 & 6.13 \times 10 \end{bmatrix}.
$$

解方程组 $Ly = P^{\intercal} b$ 可得

$$
y_1 = 2.58 \times 10, \quad y_2 \approx 6.12 \times 10.
$$

解方程组 $Ux = y$ 可得

$$
x_2 = y_2/u_{22} \approx 0.998, \quad x_1 = (y_1 - u_{12} * x_2)/u_{11} \approx 34.3/u_{11} \approx 10.0
$$

所以, 数值解具有 3 位有效数字. □

2.1.7 矩阵求逆

我们可以通过部分选主元 LU 分解来计算矩阵的逆. 设 $PA = LU$, 则

$$A^{-1} = U^{-1}L^{-1}P,$$

等价于求解下面 $2n$ 个三角线性方程组

$$Ly_i = Pe_i, \quad Ux_i = y_i, \quad i = 1, 2, \ldots, n.$$

2.2 特殊方程组的求解

如果系数矩阵具有一定的特殊结构, 则可以充分利用这些特殊结构来构造高效的算法. 本节考虑以下特殊方程组的求解:

- 对称正定情形
- 对称不定情形
- 三对角矩阵
- 带状矩阵
- Toeplitz 矩阵

2.2.1 对称正定线性方程组

考虑线性方程组

$$Ax = b$$

其中 $A \in \mathbb{R}^{n\times n}$ 对称正定的.

我们首先给出对称正定矩阵的几个基本性质.

定理 2.3 设 $A \in \mathbb{R}^{n\times n}$.

- A 对称正定当且仅当 A 对称且所有特征值都是正的;
- A 对称正定当且仅当 $X^{\intercal}AX$ 对称正定, 其中 $X \in \mathbb{R}^{n\times n}$ 是一个任意的非奇异矩阵;
- 若 A 对称正定, 则 A 的任意主子矩阵都对称正定;
- 若 A 对称正定, 则 A 的所有对角线元素都是正的, 且 $\max\limits_{i\neq j}\{|a_{ij}|\} < \max\limits_{i}\{a_{ii}\}$, 即绝对值最大的元素出现在对角线上.

定理 2.4 (Cholesky 分解) 设 $A \in \mathbb{R}^{n\times n}$ 对称正定, 则存在唯一的对角线元素为正的下三角矩阵 L, 使得

$$A = LL^{\intercal}.$$

该分解称为 Cholesky 分解.

证明. 首先证明存在性, 我们用数学归纳法来构造矩阵 L.

当 $n = 1$ 时, 由 A 的对称正定性可知 $a_{11} > 0$. 取 $l_{11} = \sqrt{a_{11}}$ 即可.

假定结论对所有不超过 $n-1$ 阶的对称正定矩阵都成立. 设 $A \in \mathbb{R}^{n\times n}$ 是 n 阶对称正定, 则 A 可分解为

$$A = \begin{bmatrix} a_{11} & A_{12} \\ A_{12}^{\intercal} & A_{22} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}^{\intercal},$$

其中 $\tilde{A}_{22} = A_{22} - A_{12}^{\mathsf{T}} A_{12}/a_{11}$. 由定理 2.3 可知, $\begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix}$ 对称正定, 故 $\tilde{A}_{22}$ 是 $n-1$ 阶对称正定矩阵. 根据归纳假设, 存在唯一的对角线元素为正的下三角矩阵 $\tilde{L}$, 使得 $\tilde{A}_{22} = \tilde{L}\tilde{L}^{\mathsf{T}}$. 令

$$L = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}} A_{12}^{\mathsf{T}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}} A_{12}^{\mathsf{T}} & \tilde{L} \end{bmatrix}.$$

易知, L 是对角线元素均为正的下三角矩阵, 且

$$LL^{\mathsf{T}} = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}} A_{12}^{\mathsf{T}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}} A_{12}^{\mathsf{T}} & I \end{bmatrix}^{\mathsf{T}} = A.$$

由归纳法可知, 对任意对称正定实矩阵 A, 都存在一个对角线元素为正的下三角矩阵 L, 使得

$$A = LL^{\mathsf{T}}.$$

唯一性可以采用反证法, 留做作业. □

☞ 该定理也可以通过 LU 分解的存在唯一性来证明.

Cholesky 分解的实现

设 $A = LL^{\mathsf{T}}$, 即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{bmatrix}.$$

直接比较等式两边的元素可得

$$a_{ij} = \sum_{k=1}^{n} l_{ik} l_{jk} = l_{jj} l_{ij} + \sum_{k=1}^{j-1} l_{ik} l_{jk}, \quad i, j = 1, 2, \ldots, n.$$

根据这个计算公式即可得下面的算法:

算法 2.9. Cholesky 分解算法

for $j = 1$ to n **do**
  $l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{1/2}$
  for $i = j+1$ to n **do**
    $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}$
  end for
end for

关于 Cholesky 算法的几点说明

- 与 LU 分解一样, 可以利用 A 的下三角部分来存储 L;
- Cholesky 分解算法的运算量为 $\frac{1}{3}n^3 + \mathcal{O}(n^2)$, 大约为 LU 分解的一半;
- Cholesky 分解算法是稳定的（稳定性与全主元 Gauss 消去法相当）, 故不需要选主元.

改进的 Cholesky 分解算法

为了避免开方运算, 我们可以将 A 分解为: $A = LDL^{\intercal}$, 即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}.$$

通过待定系数法可得

$$a_{ij} = \sum_{k=1}^{n} l_{ik} d_k l_{jk} = d_j l_{ij} + \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}, \quad i, j = 1, 2, \ldots, n.$$

基于以上分解来求解对称正定线性方程组的算法称为**改进的平方根法**:

算法 2.10. 改进的平方根法

% 先计算分解
for $j = 1$ to n **do**
  $d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k$
  for $i = j+1$ to n **do**
    $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk})/d_j$
  end for
end for
% 解方程组: $Ly = b$ 和 $DL^{\intercal}x = y$
$y_1 = b_1$
for $i = 2$ to n **do**
  $y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k$
end for
$x_n = y_n/d_n$
for $i = n-1$ to 1 **do**
  $x_i = y_i/d_i - \sum_{k=i+1}^{n} l_{ki} x_k$
end for

2.2.2 对称不定线性方程组

设 $A \in \mathbb{R}^{n \times n}$ 是非奇异的对称不定矩阵. 若 A 存在 LU 分解, 即 $A = LU$, 则可写成

$$A = LDL^{\intercal},$$

其中 D 是由 U 的对角线元素构成的对角矩阵. 然而, 当 A 不定时, 其 LU 分解不一定存在. 若采用选主元 LU 分解, 则其对称性将被破坏. 为了保持对称性, 在选主元时必须对行列进行同样的置换, 即选取置换矩阵 P, 使得

$$PAP^{\intercal} = LDL^{\intercal}. \tag{2.4}$$

通常称 (2.4) 为对称矩阵的 LDL$^{\mathrm{T}}$ 分解. 不幸的是, 这样的置换矩阵可能不一定存在, 即分解 (2.4) 不一定存在.

例 2.3 设对称矩阵

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

由于 A 的对角线元素都是 0, 对任意置换矩阵 P, 矩阵 $PAP^{\intercal}$ 的对角线元素仍然都是 0. 因此, 矩阵 A 不存在分解 (2.4).

Aasen 算法

基于以上原因, Aasen[1] 提出了下面的分解

$$PAP^{\intercal} = LTL^{\intercal}, \tag{2.5}$$

其中 P 为置换矩阵, L 为单位下三角矩阵, T 为对称三对角矩阵. 分解 (2.5) 本质上与部分选主元 LU 分解是一样的,

具体实施细节可参见 [35, 77].

块 LDL$^{\mathrm{T}}$ 分解

设 A 对称非奇异, 则存在置换矩阵 P 使得

$$PAP^{\intercal} = \begin{bmatrix} B & E^{\intercal} \\ E & C \end{bmatrix},$$

其中 $B \in \mathbb{R}$ 或 $B \in \mathbb{R}^{2\times 2}$, 且非奇异. 因此可以对 $PAP^{\intercal}$ 进行块对角化, 即

$$PAP^{\intercal} = \begin{bmatrix} I & 0 \\ EB^{-1} & I \end{bmatrix} \begin{bmatrix} B & 0 \\ 0 & C - EA^{-1}E^{\intercal} \end{bmatrix} \begin{bmatrix} I & B^{-1}E^{\intercal} \\ 0 & I \end{bmatrix},$$

其中 $C - EA^{-1}E^{\intercal}$ 是 Schur 补.

不断重复以上过程, 就可以得到 A 的块 LDL$^{\mathrm{T}}$ 分解:

$$PAP^{\intercal} = L\tilde{D}L^{\intercal},$$

其中 $\tilde{D}$ 是拟对角矩阵, 即块对角矩阵且对角块的大小为 1 或 2.

与选主元 LU 分解类似, 我们需要考虑块 LDL$^{\mathrm{T}}$ 分解的选主元策略, 即如何选取置换矩阵 P. Kahan 于 1965 年首先考虑了选主元块 LDL$^{\mathrm{T}}$ 分解. 目前常用的策略有

- 全主元策略: Bunch 和 Parlett [13] 于 1971 年提出了全主元策略来选取置换矩阵, 并证明了其稳定性 [11]. 但需要进行 $n^3/6$ 次比较运算, 代价比较昂贵.

- 部分选主元策略: 由 Bunch 和 Kaufman [12] 于 1977 年提出, 将比较运算复杂度降低到 $O(n^2)$ 量级, 而且具有较满意的向后稳定性. 因此被广泛使用. 具体实施也可以参见 [35], 或 [18] (分块版本).
- Rook 策略: 该策略由 Ashcraft, Grimes 和 Lewis [3] 提出, 整体上与部分选主元类似, 但在选主元时加了一层迭代, 从而能够提供更高的精度.

☞ 目前大部分软件都采用部分选主元块 $\mathrm{LDL^T}$ 分解来求解对称线性方程组. 关于 Aasen 算法和块 $\mathrm{LDL^T}$ 分解比较也可参见 [3, 8].

2.2.3 三对角线性方程组

考虑三对角线性方程组 $Ax = f$, 其中 A 是三对角矩阵:

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix}.$$

我们假定

$$|b_1| > |c_1| > 0, \quad |b_n| > |a_n| > 0, \tag{2.6}$$

且

$$|b_i| \geq |a_i| + |c_i|, \quad a_i c_i \neq 0, \quad i = 2, \dots, n-1. \tag{2.7}$$

即 A 是不可约弱对角占优的. 此时, 我们可以得到下面的三角分解

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & \\ a_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & a_n & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{bmatrix} \triangleq LU. \tag{2.8}$$

由待定系数法, 我们可以得到递推公式:

$$\begin{aligned} &\alpha_1 = b_1, \\ &\beta_1 = c_1/\alpha_1 = c_1/b_1, \\ &\begin{cases} \alpha_i = b_i - a_i\beta_{i-1}, \\ \beta_i = c_i/\alpha_i = c_i/(b_i - a_i\beta_{i-1}), \quad i = 2, 3, \dots, n-1 \end{cases} \\ &\alpha_n = b_n - a_n\beta_{n-1}. \end{aligned}$$

为了使得算法能够顺利进行下去, 我们需要证明 $\alpha_i \neq 0$.

定理 2.5 设三对角矩阵 A 满足条件 (2.6) 和 (2.7). 则 A 非奇异, 且

(1) $|\alpha_1| = |b_1| > 0$;

(2) $0 < |\beta_i| < 1, i = 1, 2, \dots, n-1$;

(3) $0 < |c_i| \leq |b_i| - |a_i| < |\alpha_i| < |b_i| + |a_i|, i = 2, 3, \dots, n$;

证明. 由于 A 是不可约且弱对角占优, 所以 A 非奇异. (见定理 ??)

结论 (1) 是显然的.

下面我们证明结论 (2) 和 (3).

由于 $0 < |c_1| < |b_1|$, 且 $\beta_1 = c_1/b_1$, 所以 $0 < |\beta_1| < 1$. 又 $\alpha_2 = b_2 - a_2\beta_1$, 所以

$$|\alpha_2| \geq |b_2| - |a_2| \cdot \|\beta_1\| > |b_2| - |a_2| \geq |c_2| > 0, \tag{2.9}$$

$$|\alpha_2| \leq |b_2| + |a_2| \cdot \|\beta_1\| < |b_2| + |a_2|. \tag{2.10}$$

再由结论 (2.9) 和 β_2 的计算公式可知 $0 < |\beta_2| < 1$. 类似于 (2.9) 和 (2.9), 我们可以得到

$$|\alpha_3| \geq |b_3| - |a_3| \cdot \|\beta_2\| > |b_3| - |a_3| \geq |c_3| > 0,$$

$$|\alpha_3| \leq |b_3| + |a_3| \cdot \|\beta_2\| < |b_3| + |a_3|.$$

依此类推, 我们就可以证明结论 (2) 和 (3). □

由定理 2.5 可知, 分解 (2.8) 是存在的. 因此, 原方程就转化为求解 $Ly = f$ 和 $Ux = y$. 由此便可得求解三对角线性方程组的**追赶法** 也称为 Thomas **算法** (1949), 其运算量大约为 $8n - 6$.

算法 2.11. 追赶法

```
1:  β_1 = c_1/b_1
2:  y_1 = f_1/b_1
3:  for i = 2 to n − 1 do
4:      α_i = b_i − a_i β_{i−1}
5:      β_i = c_i/α_i
6:      y_i = (f_i − a_i y_{i−1})/α_i
7:  end for
8:  α_n = b_n − a_n β_{n−1}
9:  y_n = (f_n − a_n y_{n−1})/α_n
10: x_n = y_n
11: for i = n − 1 to 1 do
12:     x_i = y_i − β_i x_{i+1}
13: end for
```

☞ 具体计算时, 由于求解 $Ly = f$ 与矩阵 LU 分解是同时进行的, 因此, α_i 可以不用存储. 但 β_i 需要存储.

☞ 由于 $|\beta_i| < 1$, 因此在回代求解 x_i 时, 误差可以得到有效控制.

需要指出的是, 我们也可以考虑下面的分解

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \gamma_2 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_n & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & c_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & \alpha_n \end{bmatrix}. \tag{2.11}$$

但此时 $|\gamma_i|$ 可能大于 1. 比如 $\gamma_2 = a_2/b_1$, 因此当 $|b_1| < |a_2|$ 时, $|\gamma_2| > 1$. 所以在回代求解时, 误差可能得不到有效控制. 另外一方面, 计算 γ_i 时也可能会产生较大的舍入误差 (大数除以小数). 但如果 A 是列对角占优, 则可以保证 $|\gamma_i| < 1$.

☞ 如果 A 是 (行) 对角占优, 则采用分解 (2.8); 如果 A 是列对角占优, 则采用分解 (2.11).

2.2.4 带状线性方程组

设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U, 即

$$a_{ij} = 0 \quad \text{for} \quad i > j + b_L \ \text{ or } \ i < j - b_U.$$

其形状如下图所示:

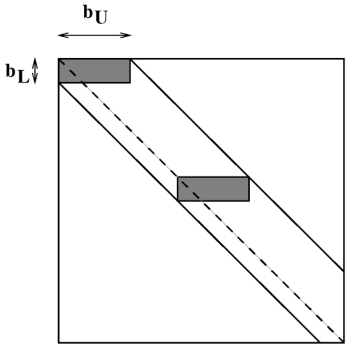


图 2.1. 带状矩阵

对于带状矩阵, 其 LU 分解有如下性质:

定理 2.6 设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U. 若 $A = LU$ 是不选主元的 LU 分解, 则 L 为下带宽为 b_L 的带状矩阵, U 为上带宽为 b_U 的带状矩阵. 求解 $Ax = b$ 的运算量大约为 $2nb_Lb_U + 2n(b_L + b_U)$.

若采用部分选主元的 LU 分解, 则有

定理 2.7 设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U. 若 $PA = LU$ 是部分选主元的 LU 分解, 则 U 为上带宽不超过 $b_L + b_U$ 的带状矩阵, L 为下带宽为 b_L 的 "基本带状矩阵", 即 L 每列的非零元素不超过 $b_L + 1$ 个.

2.2.5 Toeplitz 线性方程组

设 $T_n \in \mathbb{R}^{n\times n}$ 是 Toeplitz 矩阵, 即

$$T_n = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}.$$

易知 Toeplitz 矩阵是反向对称 (persymmetric) 矩阵, 即关于东北–西南对角线对称. 记 J_n 为 n 阶反向单位矩阵, 即

$$J_n = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}.$$

易知 $J_n^{\intercal} = J_n^{-1} = J_n$.

引理 2.2 矩阵 $A \in \mathbb{R}^{n\times n}$ 是反向对称矩阵当且仅当

$$A = J_n A^{\intercal} J_n \quad 或 \quad J_n A = A^{\intercal} J_n.$$

若 A 可逆, 则可得

$$A^{-1} = J_n^{-1} \left(A^{\intercal}\right)^{-1} J_n^{-1} = J_n \left(A^{-1}\right)^{\intercal} J_n,$$

即反向对称矩阵的逆也是反向对称矩阵.

☞ Toeplitz 矩阵的逆是反向对称矩阵, 但不一定是 Toeplitz 矩阵.

Yule-Walker 方程组的 Durbin 算法

设 T_n 对称正定, 考虑线性方程组

$$T_n x = -r_n, \tag{2.12}$$

其中 $r_n = [t_1, t_2, \ldots, t_{n-1}, t_n]^{\intercal}$. 这类线性方程组称为 **Yule-Walker 方程组**, 其中 t_n 为任意给定的实数.

由于 T_n 对称正定, 所以 $t_0 > 0$. 因此我们可以对 T_n 的对角线元素进行单位化. 不失一般性, 我们假定 T_n 的对角线元素为 1, 即

$$T_n = \begin{bmatrix} 1 & t_1 & \cdots & t_{n-1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \cdots & t_1 & 1 \end{bmatrix}.$$

由于方程组右端项的特殊性, 我们可以通过递推来求解.

设 $x^{(k)}$ 是 $T_k x = -r_k$ 的解, 下面导出 $T_{k+1} x = -r_{k+1}$ 的解 $x^{(k+1)}$. 记

$$x^{(k+1)} = \begin{bmatrix} z^{(k)} \\ \alpha_k \end{bmatrix},$$

则 $T_{k+1} x^{(k+1)} = -r_{k+1}$ 可写为

$$\begin{bmatrix} T_k & J_k r_k \\ r_k^{\intercal} J_k & 1 \end{bmatrix} \begin{bmatrix} z^{(k)} \\ \alpha_k \end{bmatrix} = - \begin{bmatrix} r_k \\ t_{k+1} \end{bmatrix}.$$

因此可得

$$z^{(k)} = T_k^{-1}(-r_k - \alpha_k J_k r_k) = x^{(k)} - \alpha_k T_k^{-1} J_k r_k, \tag{2.13}$$

$$\alpha_k = -t_{k+1} - r_k^\intercal J_k z^{(k)}. \tag{2.14}$$

由于 T_k 是反向对称矩阵, 故 $T_k^{-1} J_k = J_k T_k^{-1}$. 所以可得

$$z^{(k)} = x^{(k)} - \alpha_k T_k^{-1} J_k r_k = x^{(k)} - \alpha_k J_k T_k^{-1} r_k = x^{(k)} + \alpha_k J_k x^{(k)}.$$

代入 (2.14) 可得

$$(1 + r_k^\intercal x^{(k)})\alpha_k = -t_{k+1} - r_k^\intercal J_k x^{(k)}.$$

又

$$\begin{bmatrix} I & J_k x^{(k)} \\ 0 & 1 \end{bmatrix}^\intercal \begin{bmatrix} T_k & J_k r_k \\ r_k^\intercal J_k & 1 \end{bmatrix} \begin{bmatrix} I & J_k x^{(k)} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} T_k & 0 \\ 0 & 1 + r_k^\intercal x^{(k)} \end{bmatrix},$$

由 T_{k+1} 的对称正定性可知 $1 + r_k^\intercal x^{(k)} > 0$, 故可得 $x^{(k+1)}$ 的计算公式

$$\alpha_k = \frac{-t_{k+1} - r_k^\intercal J_k x^{(k)}}{1 + r_k^\intercal x^{(k)}}, \qquad z^{(k)} = x^{(k)} + \alpha_k J_k x^{(k)}. \quad k = 1, 2, \ldots. \tag{2.15}$$

运算量为 $\mathcal{O}(k)$. 因此, 我们就可以从一阶 Yule-Walker 方程出发, 利用递推公式 (2.15) 计算 $T_n x = -r_n$ 的解. 总的运算量大约为 $3n^2$.

为了减少运算量, 我们引入一个变量 $\beta_k \triangleq 1 + r_k^\intercal x^{(k)}$, 则

$$\begin{aligned} \beta_{k+1} &= 1 + r_{k+1}^\intercal x^{(k+1)} \\ &= 1 + [r_k^\intercal, t_{k+1}] \begin{bmatrix} x^{(k)} + \alpha_k J_k x^{(k)} \\ \alpha_k \end{bmatrix} \\ &= 1 + r_k^\intercal x^{(k)} + \alpha_k (t_{k+1} + r_k^\intercal J_k x^{(k)}) \\ &= (1 - \alpha_k^2)\beta_k. \end{aligned}$$

于是可得求解 Yule-Walker 方程组的 Durbin 算法, 总运算量大约为 $2n^2$.

算法 2.12. 求解 Yule-Walker 方程组的 Durbin 算法

1: 输入数据: $t = [t_1, t_2, \ldots, t_n]$ % 注: 这里假定 $t_0 = 1$
2: $x(1) = -t_1$, $\beta = 1$, $\alpha = -t_1$
3: **for** for $k = 1$ to $n - 1$ **do**
4: $\beta = (1 - \alpha^2)\beta$
5: $\alpha = -\left(t_{k+1} - \sum_{i=1}^{k} t_{k+1-i}\, x(i)\right) / \beta$
6: $x(1:k) = x(1:k) + \alpha x(k:-1:1)$
7: $x(k+1) = \alpha$
8: **end for**

一般右端项的 Toeplitz 线性方程组

考虑一般右端项的方程组

$$T_n x = b,$$

其中 T_n 是对称正定 Toeplitz 矩阵, $b = [b_1, b_2, \ldots, b_n]^\intercal$. 与求解 Yule-Walker 方程组类似, 我们利用递推方法来求解.

假定 $x^{(k)}$ 和 $y^{(k)}$ 分别是方程组

$$T_k x = [b_1, b_2, \ldots, b_k]^\intercal \triangleq b^{(k)}$$

和

$$T_k y = -[t_1, t_2, \ldots, t_k]^\intercal$$

的解. 设 $x^{(k+1)} = \begin{bmatrix} z^{(k)} \\ \mu_k \end{bmatrix}$ 是 $T_{k+1}x = b^{(k+1)}$ 的解, 则可得

$$\begin{bmatrix} T_k & J_k r_k \\ r_k^\intercal J_k & 1 \end{bmatrix} \begin{bmatrix} z^{(k)} \\ \mu_k \end{bmatrix} = \begin{bmatrix} b^{(k)} \\ b_{k+1} \end{bmatrix}.$$

通过计算可得

$$\begin{aligned} z^{(k)} &= T_k^{-1} b^{(k)} - \mu_k T_k^{-1} J_k r_k = x^{(k)} - \mu_k J_k T_k^{-1} r_k = x^{(k)} + \mu_k J_k y^{(k)}, \\ \mu_k &= \frac{b_{k+1} - r_k^\intercal J_k x^{(k)}}{1 + r_k^\intercal y^{(k)}}. \end{aligned}$$

所以, 我们可以先计算 $T_k x = b^{(k)}$ 和 $T_k x = -r_k$ 的解, 然后利用上述公式得到 $T_{k+1}x = b^{(k+1)}$ 的解, 这就是 Levinson 算法, 该算法的总运算量大约为 $4n^2$.

算法 2.13. 求解对称正定 Toeplitz 线性方程组的 Levinson 算法

```
输入数据: t = [t_1, t_2, ..., t_{n-1}] 和 b = [b_1, b_2, ..., b_n]     % 这里假定 t_0 = 1
y(1) = -t_1, x(1) = b_1, β = 1, α = -t_1
for k = 1 to n-1 do
    β = (1 - α^2)β
    μ = (b_{k+1} - Σ_{i=1}^{k} t_{k+1-i} x(i)) / β
    x(1:k) = x(1:k) + αx(k:-1:1), x(k+1) = α
    if k < n-1 then
        α = -(t_{k+1} - Σ_{i=1}^{k} t_{k+1-i} y(i)) / β
        y(1:k) = y(1:k) + αy(k:-1:1)
        y(k+1) = α
    end if
end for
```

在数学与工程的许多应用中都会出现 Toeplitz 线性方程组, 如样条插值, 时间序列分析, Markov 链, 排队论, 信号与图像处理等. Levinson 算法 [44] 是较早的关于对称正定 Toeplitz 线性方程组的快速算法, 但并不稳定 (只具有弱稳定性) [14]. 后来人们提出了各种各样的快速和超快速算法, 大致如下

方法	运算量	存储量
Fast stable	$\geq 20n^2$	$\geq n^2/2$
Fast but unstable	$\geq 3n^2$	$\geq 4n$
Superfast and “unstable”	$O(n\log^2 n)$	$O(n)$
Superfast preconditioner	$O(n\log n)$	$O(n)$

- Fast : Levinson-Durbin (1946), Trench (1964), ...
- Fast stable: Bareiss (1969), Gohberg, Kailath and Olshevsky (1995), Chandrasekaran and Sayed (1998), Gu (1998), ...
- Superfast: Brent, Gustavson and Yun (1980), Bitmead and Anderson (1980), Morf (1980), de Hoog (1987), Ammar and Gragg (1988), van Barel, Heinig and Kravanja (2001) [6], Stewart (2003), Codevico, Heinig and Van Barel (2005) [15], Xia, Xi and Gu (2012), ...
- Superfast Preconditioners: Strang, Chan, Chan, Tyrtyshnikov, ...

其中最后一行的 “Superfast preconditioners” 指的是预处理迭代算法. 关于 Toeplitz 线性方程组的快速求解方法介绍可参见 Stewart (2014) [60].

2.3 扰动分析

考虑线性方程组

$$Ax = b.$$

设 x_* 是精确解, $\hat{x}$ 是通过数值计算得到的近似解. 假定 $\hat{x}$ 满足线性方程组

$$(A + \delta A)\hat{x} = b + \delta b.$$

下面讨论 $\delta x \triangleq \hat{x} - x_*$ 的大小, 即向后误差分析.

2.3.1 δx 与 $\hat{x}$ 的关系

定理 2.8 设 $\|\cdot\|$ 是任一向量范数（当该范数作用在矩阵上时就是相应的导出范数）, 则 δx 与 $\hat{x}$ 满足下面的关系式

$$\frac{\|\delta x\|}{\|\hat{x}\|} \le \|A^{-1}\| \cdot \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|} \right).$$

当 $\delta b = 0$ 时, 有

$$\frac{\|\delta x\|}{\|\hat{x}\|} \le \kappa(A) \frac{\|\delta A\|}{\|A\|} \tag{2.16}$$

证明. 由等式 $(A + \delta A)\hat{x} = b + \delta b = Ax_* - \delta b$ 可知 $A(\hat{x} - x_*) = -\delta A\hat{x} + \delta b$, 即

$$\delta x = A^{-1}(-\delta A\hat{x} + \delta b).$$

所以

$$\|\delta x\| \le \|A^{-1}\| \cdot (\|\delta A\| \cdot \|\hat{x}\| + \|\delta b\|), \tag{2.17}$$

即

$$\frac{\|\delta x\|}{\|\hat{x}\|} \le \|A^{-1}\| \cdot \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|} \right).$$

若 $\delta b = 0$, 则可得

$$\boxed{\frac{\|\delta x\|}{\|\hat{x}\|} \le \kappa(A) \frac{\|\delta A\|}{\|A\|}}$$

□

2.3.2 δx 与 x_* 的关系

引理 2.3 设 $\|\cdot\|$ 是任一相容范数, $X \in \mathbb{R}^{n\times n}$. 若 $\|X\| < 1$, 则 $I - X$ 可逆, 且有

$$(I-X)^{-1} = \sum_{k=0}^{\infty} X^k \qquad \text{和} \qquad \|(I-X)^{-1}\| \le \frac{1}{1-\|X\|}.$$

证明. 先证明级数 $\sum\limits_{k=0}^{\infty} X^k$ 收敛, 即其每个分量都收敛. 记 $x_{ij}^{(k)}$ 为 X^k 的 (i,j) 元素. 由范数的等价性可知, 存在常数 c 使得对任意矩阵 $Y \in \mathbb{R}^{n\times n}$ 都有 $\|Y\|_F \le c\|Y\|$. 所以

$$\left|x_{ij}^{(k)}\right| \le \left\|X^k\right\|_F \le c\left\|X^k\right\| \le c\left\|X\right\|^k.$$

注意, 这里的常数 c 与 X 和 k 都无关. 由条件 $\|X\| < 1$ 可知, 级数 $\sum\limits_{k=0}^{\infty} c\|X\|^k$ 收敛, 所以级数 $\sum\limits_{k=0}^{\infty} x_{ij}^{(k)}$ 也收敛, 即 $\sum\limits_{k=0}^{\infty} X^k$ 收敛.

因为 $\lim\limits_{k\to\infty} \|X^k\| = 0$, 且 $(I-X)(I+X+X^2+\cdots+X^k) = I - X^{k+1}$, 所以

$$(I-X)\sum_{k=0}^{\infty} X^k = \lim_{k\to\infty}(I - X^{k+1}) = I,$$

即

$$(I-X)^{-1} = \sum_{k=0}^{\infty} X^k,$$

且

$$\|(I-X)^{-1}\| = \left\|\sum_{k=0}^{\infty} X^k\right\| \le \sum_{k=0}^{\infty}\|X^k\| \le \sum_{k=0}^{\infty}\|X\|^k = \frac{1}{1-\|X\|}.$$

□

由 $(A+\delta A)\hat{x} = b + \delta b$ 可得

$$\begin{aligned}\delta x &= (A+\delta A)^{-1}(b+\delta b - Ax_* - \delta A x_*)\\ &= (I + A^{-1}\delta A)^{-1}A^{-1}(-\delta A x_* + \delta b).\end{aligned}$$

假定 $\|\delta A\|$ 很小, 满足 $\|A^{-1}\delta A\| \le \|A^{-1}\|\cdot\|\delta A\| < 1$, 则由引理 2.3 可得

$$\begin{aligned}\frac{\|\delta x\|}{\|x_*\|} &\le \|(I+A^{-1}\delta A)^{-1}\|\cdot\|A^{-1}\|\cdot\left(\|\delta A\| + \frac{\|\delta b\|}{\|x_*\|}\right)\\ &\le \frac{\|A^{-1}\|}{1-\|A^{-1}\|\cdot\|\delta A\|}\cdot\left(\|\delta A\| + \frac{\|\delta b\|}{\|x_*\|}\right)\\ &= \frac{\|A^{-1}\|\cdot\|A\|}{1-\|A^{-1}\|\cdot\|A\|\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\|\cdot\|x_*\|}\right)\\ &\le \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right)\end{aligned}$$

当 $\|\delta A\| \to 0$ 时, 我们可得

$$\frac{\|\delta x\|}{\|x_*\|} \le \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right) \to \kappa(A)\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)$$

定理 2.9 设 $A \in \mathbb{R}^{n\times n}$ 非奇异且 $\|A^{-1}\|\cdot\|\delta A\| < 1$, 则

$$\frac{\|\delta x\|}{\|x_*\|} \le \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right). \tag{2.18}$$

如果 $\|\delta A\| = 0$, 则

$$\frac{1}{\kappa(A)}\frac{\|\delta b\|}{\|b\|} \le \frac{\|\delta x\|}{\|x_*\|} \le \kappa(A)\frac{\|\delta b\|}{\|b\|}.$$

证明. 只需证明左边一个不等式即可. 由于 $\delta A = 0$, 所以 $A\delta x = \delta b$. 两边取范数, 然后同除 $\|x_*\|$ 可得

$$\frac{\|A\|\cdot\|\delta x\|}{\|x_*\|} \ge \frac{\|A\delta x\|}{\|A^{-1}b\|} \ge \frac{\|\delta b\|}{\|A^{-1}\|\cdot\|b\|}.$$

所以结论成立. □

定理 2.10 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 则有

$$\min\left\{\frac{\|\delta A\|_2}{\|A\|_2} \;:\; A+\delta A \text{ 奇异}\right\} = \frac{1}{\kappa_2(A)}$$

证明. 记 $d \triangleq \min\left\{\|\delta A\|_2 \;:\; A+\delta A \text{ 奇异}\right\}$, 只需证明 $d = \frac{1}{\|A^{-1}\|_2}$.

先证明 $d \ge \frac{1}{\|A^{-1}\|_2}$. 若 $\|\delta A\|_2 < \|A^{-1}\|_2^{-1}$, 则

$$\|A^{-1}\delta A\|_2 \le \|A^{-1}\|_2\cdot\|\delta A\|_2 < 1.$$

由引理 2.3 可知 $I + A^{-1}\delta A$ 非奇异. 因此 $A+\delta A = A(I+A^{-1}\delta A)$ 也非奇异, 这表明使得 $A+\delta A$ 奇异的 δA 必须满足 $\|\delta A\|_2 \ge \|A^{-1}\|_2^{-1}$, 即

$$d \ge \frac{1}{\|A^{-1}\|_2}.$$

下面证明 $d \le \frac{1}{\|A^{-1}\|_2}$, 即证明存在 δA 满足 $\|\delta A\|_2 = \|A^{-1}\|_2^{-1}$ 使得 $A+\delta A$ 奇异. 由范数的定义可知

$$\|A^{-1}\|_2 = \max_{\|x\|_2=1}\|A^{-1}x\|_2,$$

故存在 x 满足 $\|x\|_2 = 1$ 使得

$$\|A^{-1}\|_2 = \|A^{-1}x\|_2.$$

令 $y = A^{-1}x/\|A^{-1}x\|_2$, 则 $\|y\|_2 = 1$, 且

$$\|xy^{\intercal}\|_2 = \max_{\|z\|_2=1}\|xy^{\intercal}z\|_2 = \max_{\|z\|_2=1}|y^{\intercal}z|\cdot\|x\|_2 = \max_{\|z\|_2=1}|y^{\intercal}z|.$$

由于 $|y^\intercal z| \le \|y\|_2 \cdot \|z\|_2 = 1$, 且当 $z = y$ 时有 $|y^\intercal z| = 1$, 所以 $\|xy^\intercal\|_2 = 1$. 构造

$$\delta A = -\frac{xy^\intercal}{\|A^{-1}\|_2},$$

则

$$\|\delta A\|_2 = \frac{\|xy^\intercal\|_2}{\|A^{-1}\|_2} = \frac{1}{\|A^{-1}\|_2}.$$

下面证明 $A+\delta A$ 奇异. 我们只需证明以 $A+\delta A$ 为系数矩阵的齐次线性方程组有非零解. 由于 $\|A^{-1}x\|_2 = \|A^{-1}\|_2$, 容易验证

$$(A+\delta A)y = A\frac{A^{-1}x}{\|A^{-1}x\|_2} - \frac{xy^\intercal}{\|A^{-1}\|_2}y = \frac{x}{\|A^{-1}\|_2} - \frac{x}{\|A^{-1}\|_2} = 0,$$

即 $A+\delta A$ 奇异, 所以 $d \le \frac{1}{\|A^{-1}\|_2}$.

综上可得

$$d = \min\left\{\|\delta A\|_2 \ : \ A+\delta A \text{ 奇异}\right\} = \frac{1}{\|A^{-1}\|_2}.$$

□

☞ 定理 2.10 的结论对所有 p-范数都成立, 参见 [24, 40].

☞ 度量

$$\mathrm{dist}_p(A) \triangleq \min\left\{\frac{\|\delta A\|_p}{\|A\|_p} \ : \ A+\delta A \text{ 奇异}\right\} = \frac{1}{\kappa_p(A)},$$

表示 A 距离奇异矩阵集合的相对距离.

2.3.3 δx 与残量的关系

这是研究线性方程组的扰动理论的一个较实用的方法.

记残量 (残差) 为 $r = b - A\hat{x}$, 则有

$$\delta x = \hat{x} - x_* = \hat{x} - A^{-1}b = A^{-1}(A\hat{x} - b) = -A^{-1}r,$$

所以可得

$$\boxed{\|\delta x\| \le \|A^{-1}\| \cdot \|r\|}$$

这个估计式的优点是不用去估计 δA 和 δb 的大小. 由于在实际计算中, r 通常是可以计算的, 因此该估计式比较实用.

定理 2.11 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $\|\cdot\|$ 为任一算子范数. 记 $r = b - A\hat{x}$, 则

(1) 若存在 $\hat{A}$ 满足 $\hat{A}\hat{x} = b$, 则 $\|\hat{A} - A\| \ge \dfrac{\|r\|}{\|\hat{x}\|}$;

(2) 存在 δA 满足 $\|\delta A\| = \dfrac{\|r\|}{\|\hat{x}\|}$, 使得 $(A+\delta A)\hat{x} = b$.

证明. (1) 由 $\hat{A}\hat{x} = b$ 可知

$$(\hat{A} - A)\hat{x} = b - A\hat{x} = r.$$

所以有

$$\|r\| = \|(\hat{A} - A)\hat{x}\| \le \|\hat{A} - A\| \cdot \|\hat{x}\|,$$

即

$$\|\hat{A} - A\| \ge \frac{\|r\|}{\|\hat{x}\|}.$$

(2) 以 2-范数为例, 取 $\delta A = \dfrac{r\hat{x}^{\intercal}}{\|\hat{x}\|_2^2}$ 即可. □

2.3.4 相对扰动分析

前面给出了解的误差 δx 的界是与条件数 $\kappa(A)$ 与 δA 和 δb 成比例的. 在许多情况下, 这个界是令人满意的. 但有时会相差很大, 这个界就不能很好的反映实际计算中解的误差.

例 2.4 设 $A = \begin{bmatrix} \gamma & 0 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} \gamma \\ 1 \end{bmatrix}$, 其中 $\gamma > 1$. 则 $Ax = b$ 的精确解为 $x_* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, 任何合理的直接法求得的解的误差都很小. 但系数矩阵的谱条件数为 $\kappa_2(A) = \gamma$, 当 γ 很大时, $\kappa_2(A)$ 也很大, 因此误差界 (2.16) 和 (2.18) 可以是很大.

针对这个问题, 我们按分量进行分析. 记

$$\delta A = \begin{bmatrix} \delta a_{11} & \\ & \delta a_{22} \end{bmatrix}, \quad \delta b = \begin{bmatrix} \delta b_1 \\ \delta b_2 \end{bmatrix},$$

并设 $|\delta a_{ij}| \le \varepsilon |a_{ij}|$, $|\delta b_i| \le \varepsilon |b_i|$. 则

$$\delta x = \begin{bmatrix} \hat{x}_1 - x_1 \\ \hat{x}_2 - x_2 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 + b_1}{\delta a_{11} + a_{11}} - 1 \\ \dfrac{\delta b_2 + b_2}{\delta a_{22} + a_{22}} - 1 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 + \gamma}{\delta a_{11} + \gamma} - 1 \\ \dfrac{\delta b_2 + 1}{\delta a_{22} + 1} - 1 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 - \delta a_{11}}{\delta a_{11} + \gamma} \\ \dfrac{\delta b_2 - \delta a_{22}}{\delta a_{22} + 1} \end{bmatrix}.$$

故

$$\|\delta x\|_\infty \le \frac{2\varepsilon}{1 - \varepsilon}.$$

如果 $\delta b = 0$, 则

$$\|\delta x\|_\infty \le \frac{\varepsilon}{1 - \varepsilon}.$$

这个界与 (2.16) 或 (2.18) 相差约 γ 倍.

相对条件数

为了得到更好误差界, 我们引入**相对条件数** $\kappa_{cr}(A)$, 即

$$\kappa_{cr}(A) \triangleq \big\| \, |A^{-1}| \cdot |A| \, \big\|,$$

有时也称为 Bauer 条件数或 Skeel 条件数.

假定 δA 和 δb 满足 $|\delta A| \leq \varepsilon|A|$ 和 $|\delta b| \leq \varepsilon|b|$. 则由 $(A+\delta A)\hat{x} = b+\delta b$ 可得

$$\begin{aligned} |\delta x| &= |A^{-1}(-\delta A\hat{x} + \delta b)| \\ &\leq |A^{-1}| \cdot (|\delta A| \cdot |\hat{x}| + |\delta b|) \\ &\leq |A^{-1}| \cdot (\varepsilon|A| \cdot |\hat{x}| + \varepsilon|b|) \\ &= \varepsilon|A^{-1}| \cdot (|A| \cdot |\hat{x}| + |b|). \end{aligned} \tag{2.19}$$

若 $\delta b = 0$, 则有

$$\|\delta x\| = \|\, |\delta x| \,\| \leq \varepsilon \|\, |A^{-1}| \cdot |A| \cdot |\hat{x}| \,\| \leq \varepsilon \|\, |A^{-1}| \cdot |A| \,\| \cdot \|\hat{x}\|,$$

即

$$\boxed{\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|\, |A^{-1}| \cdot |A| \,\| \cdot \varepsilon = \kappa_{cr}(A) \cdot \varepsilon} \tag{2.20}$$

相对条件数有下面的性质

引理 2.4 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $D \in \mathbb{R}^{n\times n}$ 为非奇异对角矩阵, 则

$$\kappa_{cr}(DA) = \kappa_{cr}(A).$$

定理 2.12 设 $A \in \mathbb{R}^{n\times n}$ 非奇异. 使得 $|\delta A| \leq \varepsilon|A|$, $|\delta b| \leq \varepsilon|b|$ 成立, 且满足

$$(A+\delta A)\hat{x} = b + \delta b$$

的最小的 $\varepsilon > 0$ 称为**按分量的相对向后误差**, 其表达式为

$$\varepsilon = \max_{1\leq i\leq n} \frac{|r_i|}{(|A| \cdot |\hat{x}| + |b|)_i},$$

其中 $r = b - A\hat{x}$.

更多关于数值计算的稳定性和矩阵扰动分析方面的知识, 可以参考 [35, 59, 75].

2.4 误差分析

2.4.1 LU 分解的舍入误差分析

关于 LU 分解的舍入误差分析, 我们有下面的结果.

定理 2.13 假定 $A \in \mathbb{R}^{n\times n}$ 的所有顺序主子式都不为 0, 则带舍入误差的 LU 分解可表示为

$$A = LU + E,$$

其中误差 E 满足

$$|E| \leq \gamma_n |L| \cdot |U|.$$

这里 $\gamma_n = \dfrac{n\varepsilon_u}{1-n\varepsilon_u}$, ε_u 表示机器精度.

证明. 见 [35, page 164]. □

2.4.2 Gauss 消去法的舍入误差分析

引理 2.5 [35] 设 $\hat{y}$ 和 $\hat{x}$ 分别是由向前回代算法 2.6 和向后回代算法 2.7 计算得到的数值解, 则

$$\begin{aligned}(L+\delta L)\hat{y} &= b, \quad |\delta L| \le \gamma_n |L| \\ (U+\delta U)\hat{x} &= \hat{y}, \quad |\delta U| \le \gamma_n |U|.\end{aligned}$$

该引理表明, $\hat{y}$ 和 $\hat{x}$ 只有很小的误差, 因此向前回代算法 2.6和向后回代算法 2.7 都是稳定的. 于是

$$\begin{aligned} b &= (L+\delta L)\hat{y} \\ &= (L+\delta L)(U+\delta U)\hat{x} \\ &= (LU + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U)\hat{x} \\ &= (A - E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U)\hat{x}. \end{aligned}$$

记 $\delta A = -E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U$, 则 $\hat{x}$ 是扰动方程 $(A+\delta A)x = b$ 精确解, 且

$$\begin{aligned} |\delta A| &= |-E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U| \\ &\le |E| + |L|\cdot|\delta U| + |\delta L|\cdot|U| + |\delta L|\cdot|\delta U| \\ &\le (3\gamma_n + \gamma_n^2)|L|\cdot|U| \\ &\le \gamma_{3n}|L|\cdot|U|, \end{aligned}$$

其中 $\gamma_{3n} = \dfrac{3n\varepsilon_u}{1-3n\varepsilon_u}$. 两边取范数后可得

$$\|\delta A\| \le 3n\varepsilon_u \|L\|\cdot\|U\|$$

对 1-范数, ∞-范数和 F-范数成立 (2-范数不成立).

根据算法向后稳定性的定义, 要说明带选主元 Gauss 消去法是向后稳定的, 必须要求 $\|\delta A\|$ 是 “小” 的, 即

$$\|\delta A\| = O(\varepsilon_u)\|A\|.$$

数值试验表明, 部分选主元 Gauss 消去法几乎总是保持

$$\|L\|\cdot\|U\| \approx \|A\|.$$

记

$$\rho_n \triangleq \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}$$

为部分选主元 Gauss 消去法的**增长因子**, 其中 $a_{ij}^{(k)}$ 是部分选主元 Gauss 消去法过程第 k 步时 a_{ij} 的值. 由于 $\|L\|_\infty \le n$, $\|U\|_\infty \le n\rho_n \|A\|_\infty$, 因此 [35]

定理 2.14 设 $\hat{x}$ 是由部分选主元 Gauss 法得到的数值解, 则 $\hat{x}$ 满足

$$(A+\delta A)\hat{x}=b,\quad \|\delta A\|_\infty \le n^2\gamma_{3n}\rho_n\|A\|_\infty. \tag{2.21}$$

所以若 ρ_n 比较小或随着 n 变大时增长比较缓慢, 则当 n 不是很大时, 部分选主元 Gauss 消去法是向后稳定的. 遗憾的是, 理论上无法保证 ρ_n 比较小 [35, page 166].

性质 2.1 部分选主元 Gauss 消去法能保证 $\rho_n \le 2^{n-1}$, 且这个界是可以达到的.

☞ 事实上, (2.21) 中的界几乎总是远远大于真正的 $\|\delta A\|$.

☞ 在绝大多数情况下, 部分选主元 Gauss 消去法是向后稳定的, 但理论上也存在失败的例子.

☞ 全主元 Gauss 消去法是数值稳定的. 在大部分实际应用中, 部分选主元 Gauss 消去法与全主元 Gauss 消去法具有同样的数值稳定性.

2.5 解的改进和条件数估计

当矩阵 A 是病态时, 即使残量 $r=b-A\hat{x}$ 很小, 所求得的数值解 $\hat{x}$ 仍可能带有较大的误差. 此时需要通过一些方法来提高解的精度.

2.5.1 高精度运算

在计算中, 尽可能采用高精度的运算. 比如, 原始数据是单精度的, 但在计算时都采用双精度运算, 或者更高精度的运算. 但更高精度的运算会带来更大的开销.

2.5.2 矩阵元素缩放 (Scaling)

如果 A 的元素在数量级上相差很大, 则在计算过程中很可能会出现大数与小数的加减运算, 这样就可能会引入更多的舍入误差. 为了避免由于这种情况而导致的舍入误差, 我们可以在求解之前先对矩阵元素进行缩放 (Scaling), 即在矩阵两边同时乘以两个适当的对角矩阵.

例 2.5 考虑线性方程组

$$\begin{bmatrix} -4000 & 2000 & 2000 \\ 2000 & 0.78125 & 0 \\ 2000 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 400 \\ 1.3816 \\ 1.9273 \end{bmatrix}.$$

用部分选主元 Gauss 消去法求解, 计算过程中保留 8 位有效数字, 最后求得的数值解为

$$\tilde{x}=[0.00096365, -0.698496, 0.90042329]^\intercal.$$

而精确解为 $x=[1.9273..., -0.698496..., 0.9004233...]^\intercal$. 数值解的第一个分量存在很大的误差.

我们考虑对矩阵元素进行缩放, 即在方程两边同时乘以一个对角矩阵 $D=\mathrm{diag}(0.00005, 1, 1)$, 然

后求解一个新的方程组

$$DADy = Db.$$

最后令 $\tilde{x} = Dy$, 即可求得比较精确的数值解.

设 β 是计算机浮点数的基 (一般为 2), 构造对角矩阵

$$D_1 = \mathrm{diag}(\beta^{r_1}, \beta^{r_2}, \ldots, \beta^{r_n}), \quad D_2 = \mathrm{diag}(\beta^{c_1}, \beta^{c_2}, \ldots, \beta^{c_n}).$$

将原线性方程组 $Ax = b$ 转化为等价方程组 $D_1^{-1}AD_2y = D_1^{-1}b$, 即缩放后的线性方程组. 在对方程组进行缩放时, 需要 $\mathcal{O}(n^2)$ 运算量, 通常不会产生较大的舍入误差.

☞ 对角矩阵 D_1 是对系数矩阵进行缩放, 而 D_2 是对未知量进行缩放.

通过求解缩放后的线性方程组, 我们可以证明 [27]

$$\frac{\|D_2^{-1}(\tilde{x} - x)\|_\infty}{\|D_2^{-1}x\|_\infty} \approx \varepsilon_u \kappa_\infty(D_1^{-1}AD_2),$$

其中 ε_u 是机器精度.

☞ 为了平衡矩阵元素的大小, 一种好的方案是左乘一个对角矩阵 D^{-1} (即对 A 行缩放), 其中 $D_{ii} = \sum\limits_{j=1}^{n} |a_{ij}|$. 然后再执行部分选主元 LU 分解.

2.5.3 迭代改进法

设近似解 $\hat{x}$, 残量 $r = b - A\hat{x}$. 当 $\hat{x}$ 没达到精度要求时, 可以考虑方程 $Az = r$. 如果 z 该方程的精确解, 则

$$A(\hat{x} + z) = A\hat{x} + Az = (b - r) + r = b,$$

因此 $\hat{x} + z$ 就是原方程的精确解. 在实际计算中, 我们只能得到近似解 $\hat{z}$, 但 $\|r - A\hat{z}\|$ 会很小, 特别地, 应该比 $\|r\|$ 更小. 因此 $\hat{x} + \hat{z}$ 应该比 $\hat{x}$ 更接近精确解.

如果新的近似解 $\hat{x} + \hat{z}$ 还不满足精度要求, 则可重复以上过程.

这就是通过迭代来提高解的精度.

算法 2.14. 通过迭代改进解的精度

1: 设 $PA = LU$, $\hat{x}$ 是 $Ax = b$ 的近似解
2: **while** 近似解 $\hat{x}$ 不满足精度要求, **do**
3: 　计算 $r = b - A\hat{x}$
4: 　求解 $Ly = Pr$, 即 $y = L^{-1}Pr$
5: 　求解 $Uz = y$, 即 $z = U^{-1}y$
6: 　令 $\hat{x} = \hat{x} + z$
7: **end while**

由于每次迭代只需计算一次残量和求解两个三角线性方程组, 因此运算量为 $\mathcal{O}(n^2)$. 所以相对来讲还是比较经济的.

☞ 为了提高计算精度, 在计算残量 r 时最好使用原始数据 A, 而不是 $P^{\intercal}LU$, 因此对 A 做 LU 分解时需要保留矩阵 A, 不能被 L 和 U 覆盖.

☞ 实际计算经验表明, 当 A 病态不是很严重时, 即 $\varepsilon_u\kappa_\infty(A) < 1$, 迭代法可以有效改进解的精度, 最后达到机器精度. 但 $\varepsilon_u\kappa_\infty(A) \geq 1$ 时, 一般没什么效果. 这里 ε_u 表示机器精度.

2.6 课后习题

练习 2.1 设 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, 且 $a_{11} \neq 0$, 经过一次 Gauss 消去法后得到 $A^{(2)} = \begin{bmatrix} a_{11} & * \\ 0 & A_2 \end{bmatrix}$.
证明: (1) 若 A 对称, 则 A_2 也对称; (2) 若 A 对称正定, 则 A_2 也对称正定.

练习 2.2 验证等式 (2.3).

练习 2.3 设矩阵 $A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 12 \end{bmatrix}$. 计算 A 的 LU 分解.

练习 2.4 设矩阵 $A = \begin{bmatrix} 4 & 2 & 4 \\ 2 & 37 & 8 \\ 4 & 8 & 14 \end{bmatrix}$. 计算 A 的 Cholesky 分解.

练习 2.5 设矩阵 $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & a \\ 0 & a & 2 \end{bmatrix}$. 问: 当 a 取何值时, A 存在 Cholesky 分解?

练习 2.6 设 $A \in \mathbb{R}^{n\times m}$, 其中 $n \geq m$, 证明: $\|A^{\intercal}A\|_2 = \|A\|_2^2$.
当 $m = n$ 时, 证明: $\kappa_2(A^{\intercal}A) = \kappa_2(A)^2$.

练习 2.7 设 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 对称正定, 证明: $a_{ij}^2 < a_{ii}a_{jj}$.

练习 2.8 证明 Cholesky 分解 (即定理 2.4) 的唯一性.

练习 2.9 设 $A \in \mathbb{R}^{n\times n}$ 对称非奇异, 且存在分解 $A = LDM^{\intercal}$, 其中 $L, M \in \mathbb{R}^{n\times n}$ 是单位下三角矩阵, $D \in \mathbb{R}^{n\times n}$ 是对角矩阵. 证明: $L = M$.

练习 2.10 将 $A \in \mathbb{R}^{n\times n}$ 写成分块形式

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $A_{11} \in \mathbb{R}^{k\times k}$ $(1 \leq k \leq n)$ 非奇异. 我们称矩阵 $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ 为 A 中 A_{11} 的 Schur 补 (通常简称 Schur 补).

(1) 假设 A 存在 LU 分解, 证明: 对于不选主元的 Gauss 消去法, 第 k 步后, A_{22} 已被 S 覆盖.

(2) 假设 $A_{21} = A_{12}^{\intercal}$, 且 A_{11} 和 $-A_{22}$ 都正定, 证明 A 非奇异.

练习 2.11 设 $A \in \mathbb{R}^{n\times n}$ 严格列对角占优, 即

$$|a_{jj}| > \sum_{i=1,\, i\neq j}^{n} |a_{ij}|.$$

证明:

(1) A 非奇异;

(2) 部分选主元 Gauss 消去法与不选主元的 Gauss 消去法是一样的, 即 Gauss 消去法进行 k 步之后, 剩余的右下角部分仍然是严格列对角占优.

练习 2.12 给定一个非奇异矩阵 A 和一个向量 b.
试证明: 对充分小的 $\|\delta A\|_2$, 存在非零的 δA 和 δb, 使得不等式 (2.17) 中的等号成立.

思考题

练习 2.13 以 1-范数或 ∞-范数为例, 证明定理 2.11 中的结论 (2).

练习 2.14 证明不等式 (2.19) 和 (2.20) 中的等号是可以达到的.

实践题

练习 2.15 写出列存储方式一般下三角方程组 $Ly = b$ 的求解算法, 并编写相应的 MATLAB 程序.
函数形式: `y=LU_Ly(L,b)`

练习 2.16 根据算法 2.11, 编写求解对角占优三对角线性方程组的追赶法程序.
函数形式: `x=LU_tridiag(a,b,c,f)`
其中 a, b, c 分别是系数矩阵的三条对角线, f 是右端项.

练习 2.17 带状矩阵的 LU 分解. 设 A 是 n 阶带状矩阵, 上带宽为 $L < n$, 下带宽为 $M < n$, 编写一个函数, 计算 A 的 LU 分解 (不带选主元), 并统计运算量.
函数形式: `[L,U]=LU_banded(A,L,M)`

参考文献

[1] J. O. Aasen, On the reduction of a symmetric matrix to tridiagonal form, BIT, 11 (1971), 233–242. Cited on page 2-16.

[2] M. Arioli, V. Pták, and Z. Strakoš, Krylov sequences of maximal length and convergence of GMRES, *BIT*, 38 (1998), 636–643. No citations.

[3] C. Ashcraft, R. G. Grimes and J. G. Lewis, Accurate symmetric indefinite linear equation solvers, SIAM Journal on Matrix Analysis and Applications, 20 (1998), 513–561. Cited on page 2-17.

[4] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994. No citations.

[5] Z.-Z. Bai, G. H. Golub and M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM Journal on Matrix Analysis and Applications*, 24 (2003), 603–626. No citations.

[6] M. Van Barel, G. Heinig and P. Kravanja, A stabilized superfast solver for nonsymmetric toeplitz systems, *SIAM Journal on Matrix Analysis and Applications*, 23 (2001), 494–510. FORTRAN code `http://people.cs.kuleuven.be/~marc.vanbarel/software/` Cited on page 2-23.

[7] R. Barrett, et.al, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1994. (`http://www.netlib.org/templates/index.html`) No citations.

[8] V. Barwell and A. George, A comparison of algorithms for solving symmetric indefinite systems of linear equations, ACM Transactions on Mathematical Software, 2 (1976), 242–251. Cited on page 2-17.

[9] Åke Björck, Solving linear least square problems by Gram-Schmidt orthogonalization, BIT, 7 (1967), 1–21. No citations.

[10] Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996. No citations.

[11] J. R. Bunch, Analysis of the diagonal pivoting method, *SIAM Journal on Numerical Analysis*, 8 (1971), 656–680. Cited on page 2-16.

[12] J. R. Bunch and L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, *Mathematics of Computation*, 31 (1977), 163–179. Cited on page 2-17.

[13] J. R. Bunch and B. N. Parlett, Direct methods for solving symmetric indefinite systems of linear equations, *SIAM Journal on Numerical Analysis*, 8 (1971), 639–655. Cited on page 2-16.

[14] R. P. Brent, Stability of fast algorithms for structured linear systems, `http://arxiv.org/pdf/1005.0671v1.pdf`, 2010. Cited on page 2-22.

[15] G. Codevico, G. Heinig and M. Van Barel, A superfast solver for real symmetric Toeplitz systems using real trigonometric transformations, *Numer. Linear Algebra Appl.*, 12 (2005), 699–713. MATLAB code `http://people.cs.kuleuven.be/~marc.vanbarel/software/` Cited on page 2-23.

[16] J. J. M. Cuppen, A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem, *Numerische Mathematik*, 36 (1981), 177–195. No citations.

[17] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997. No citations.

[18] J. J. Dongarra, I. S. Duff, D. C. Sorensen and H. A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, PA, 1998. Cited on page 2-17.

[19] Z. Drmač and K. Veselić, New fast and accurate jacobi SVD algorithm. I SIAM Journal on Matrix Analysis and Applications, 29 (2008), 1322–1342. No citations.

[20] Z. Drmač and K. Veselić, New fast and accurate jacobi SVD algorithm. II SIAM Journal on Matrix Analysis and Applications, 29 (2008), 1343–1362. No citations.

[21] V. Faber, W. Joubert, E. Knill and T. Manteuffel, Minimal residual method stronger than polynomial preconditioning, *SIAM Journal on Matrix Analysis and Applications*, 17 (1996), 707–729. No citations.

[22] K. Fernando and B. Parlett, Accurate singular values and differential qd algorithms, Numerische Mathematik, 67 (1994), 191–229. No citations.

[23] B. Fischer, *Polynomial based iteration methods for symmetric linear systems*, Wiley-Teubner Series Advances in Numerical Mathematics, John Wiley & Sons Ltd., Chichester, 1996. No citations.

[24] N. Gastinel, *Linear Numerical Analysis*, Kershaw Publishing, London, 1083. Cited on page 2-26.

[25] G. H. Golub, History of numerical linear algebra: A personal view, Stanford, 2007. Available at http://forum.stanford.edu/events/2007slides/plenary/history-revised-2007-03-19-golub.pdf No citations.

[26] G. H. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM Journal on Numerical Analysis, Series B, 2 (1965), 205–224. No citations.

[27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The 4th Editon, The Johns Hopkins University Press, Baltimore, MD, 2013. Cited on page 2-31.

[28] A. Greenbaum and L. Gurvits, Max-min properties of matrix factor norms, *SIAM Journal on Scientific Computing*, 15 (1994), 348–358. No citations.

[29] A. Greenbaum, V. Pták and Z. Strakoš, Any nonincreasing convergence curve is possible for GMRES, *SIAM Journal on Matrix Analysis and Applications*, 17 (1996), 465–469. No citations.

[30] A. Greenbaum and Z. Strakoš, Matrices that generate the same Krylov residual spaces, in *Recent Advances in Iterative Methods*, vol. 60 of IMA Vol. Math. Appl., Springer, New York, 1994, pp. 95–118. No citations.

[31] M. Gu and S. C. Eisenstat, A stable algorithm for the rank-1 modification of the symmetric eigenproblem, SIAM Journal on Matrix Analysis and Applications, 15 (1994), 1266–1276. No citations.

[32] M. Gu and S. C. Eisenstat, A Divide-and-Conquer algorithm for the bidiagonal SVD, SIAM Journal on Matrix Analysis and Applications, 16 (1995), 79–92. No citations.

[33] M. Gu and S. C. Eisenstat, A Divide-and-Conquer algorithm for the symmetric tridiagonal eigenproblem, SIAM Journal on Matrix Analysis and Applications, 16 (1995), 172–191. No citations.

[34] A. Hadjidimos, Accelerated overrelaxation method, Mathematics of Computation, 32 (1978), 149–157. No citations.

[35] Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, Second Edition, SIAM, Philadelphia, 2002. Cited on pages 2-9, 2-16, 2-17, 2-28, 2-29, and 2-30.

[36] L. Hogben, *Handbook of Linear Algebra*, 2nd, CRC Press, 2014. No citations.

[37] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985. No citations.

[38] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991. No citations.

[39] W. Joubert, A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems, *SIAM Journal on Scientific Computing*, 15 (1994), 427–439. No citations.

[40] W. Kahan Numerical Linear Algebra, Canadian Math. Bull., 9 (1966), 757–801. Cited on page 2-26.

[41] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995. No citations.

[42] D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems*, Lecture Notes in Computational Sciences and Engineering 46, Springer-Verlag, 2005. No citations.

[43] R. Lehoucq, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration*, Ph.D. thesis, Rice University, Houston, TX, 1995. No citations.

[44] N. Levinson The Wiener RMS (root mean square) error criterion in filter design and prediction, *J. Math. Phys.*, 25 (1946), 261–278. Cited on page 2-22.

[45] J. Liesen, Computable convergence bounds for GMRES, *SIAM Journal on Matrix Analysis and Applications*, 21 (2000), 882–903. No citations.

[46] J. Liesen and P. Tichý, Convergence analysis of Krylov subspace methods, *GAMM-Mitteilungen*, 27 (2004), 153–173. No citations.

[47] E. H. Moore, On the reciprocal of the general algebraic matrix, *Bull. Amer. Math. Soc.*, 26 (1920), 394–395. No citations.

[48] Christopher C. Paige, Miroslav Rozložník and Zdeněk Strakoš, Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES, *SIAM Journal on Matrix Analysis and Applications*, (28) 2006, 264–-284. No citations.

[49] B. N. Parlett, *The Symmetric Eigenvalue Problem*, The 2nd Edition, SIAM, Philadelphia, PA, 1998. No citations.

[50] D. W. Peaceman and H. H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, *Journal of the Society for Industrial and Applied Mathematics*, 3 (1955), 28–41. No citations.

[51] R. Penrose, A generalized inverse for matrices, Proc. Cambridge Philos. Soc., 51 (1955), 406–413. No citations.

[52] V. Britanak, P. Yip and K. Rao, *Discrete Cosine and Sine Transforms: General properties, Fast algorithms and Integer Approximations*, Academic Press, 2007. No citations.

[53] J. Rutter, *A Serial Implementation of Cuppen'S Divide and Conquer Algorithm for the Symmetric Eigenvalue Problem*, Master's Thesis, University of California, 1994. No citations.

[54] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual method for solving nonsymmetric linear systems, *SIAM Journal on Scientific & Statistical Computing*, 7 (1986), 856–869. No citations.

[55] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms*, Manchester University Press, Manchester, UK, 1992. No citations.

[56] D. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM Journal on Matrix Analysis and Applications*, 13 (1992), 357–385. No citations.

[57] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decomposition*, SIAM, Philadelphia, PA, 1998. No citations.

[58] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, PA, 2001. No citations.

[59] G. W. Stewart and Ji-guang Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990. Cited on page 2-28.

[60] M. Stewart, Fast algorithms for structured matrix computations, in *Handbook of Linear Algebra*, 2nd, section 62, CRC Press, 2014. Cited on page 2-23.

[61] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997. No citations.

[62] L. N. Trefethen, Numerical Analysis, in *Princeton Companion to Mathematics*, Edited by T. Gowers, J. Barrow-Green and I. Leader, Princeton University Press, 2008. No citations.

[63] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, 2007. No citations.

[64] D. S. Watkins and L. Elsner, Convergence of algorithms of decomposition type for the eigenvalue problem, *Linear Algebra and its Applications*, 143 (1991), 19–47. No citations.

[65] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford University, Oxford, 1965. No citations.

[66] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962. 2nd edition, Springer-Verlag, Berlin, 2000. No citations.

[67] D. M. Young, *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*, PhD thesis, Harward University, 1950. No citations.

[68] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971. No citations.

[69] 北京大学数学系, 高等代数 (第三版), 高等教育出版社, 2003. No citations.

[70] 陈志明, 科学计算：科技创新的第三种方法, 中国科学院院刊, 27 (2012), 161–166. No citations.

[71] 戴华, 矩阵论, 科学出版社, 2001. No citations.
[72] 胡家赣, 线性代数方程组的迭代解法, 科学出版社, 1991. No citations.
[73] 蒋尔雄, 矩阵计算, 科学出版社, 2008. No citations.
[74] 李大明, 数值线性代数, 清华大学出版社. No citations.
[75] 孙继广, 矩阵扰动分析, 科学出版社, 北京, 2001. Cited on page 2-28.
[76] 魏木生, 广义最小二乘问题的理论与计算, 科学出版社, 北京, 2006. No citations.
[77] 徐树方, 矩阵计算的理论与方法, 北京大学出版社, 北京, 1995. Cited on page 2-16.
[78] 徐树方, 钱江, 矩阵计算六讲, 高等教育出版社, 北京, 2011. No citations.