第二讲　线性方程组的直接解法

— 矩阵分解法

矩阵分解

- 矩阵分解是矩阵计算中的一个非常重要的技术.
- 通过矩阵分解, 将原矩阵分解成若干个结构简单的矩阵的乘积, 从而将原本复杂的问题转化为若干个相对简单的问题.
- 这也是我们在实际生活中解决问题的一个基本思想 — **化繁为简**

三种分解

- **LU 分解**

$$\boxed{A = LU}, \quad L \text{ 单位下三角}, U \text{ 非奇异上三角}$$

- **Crout 分解**

$$\boxed{A = \tilde{L}\tilde{U}}, \quad \tilde{L} \text{ 非奇异下三角}, \tilde{U} \text{ 单位上三角}$$

- **LDR 分解**

$$\boxed{A = LDR}, \quad L \text{ 单位下三角}, U \text{ 单位上三角}, D \text{ 对角}$$

✍ 显然, 这三种分解在本质上没有任何区别, 在实际计算中可以根据需要选择其中的一种. 这里我们只讨论 LU 分解.

待定系数法计算 LU 分解

设 A 的 LU 分解 $A = LU$ 为:

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

待定系数法计算 LU 分解

设 A 的 LU 分解 $A = LU$ 为:

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

首先观察等式两边的第一行, 可得 U 的第一行:

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

待定系数法计算 LU 分解

设 A 的 LU 分解 $A=LU$ 为:

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

首先观察等式两边的第一行, 可得 U 的第一行:

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

再观察等式两边的第一列, 可得 L 的第一列:

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

依此类推, 不断比较两边的第 2 行和第 2 列, 第 3 行和第 3 列, 等等, 就可以把 L 和 U 的所有元素都计算出来, 这就是 **待定系数法**.

依此类推, 不断比较两边的第 2 行和第 2 列, 第 3 行和第 3 列, 等等, 就可以把 L 和 U 的所有元素都计算出来, 这就是 **待定系数法**.

一般过程: 第 k 步的计算公式

比较等式两边的第 k 行, 可得

$$u_{kj} = a_{kj} - (l_{k1}u_{1j} + l_{k2}u_{2j} + \cdots + l_{k,k-1}u_{k-1,j}), \quad j = k, k+1, \ldots.$$

比较等式两边的第 k 列, 可得

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

✍ 同样地, 我们将 L 存放在 A 的下三角部分, U 存放在 A 的上三角部分.

算法 1.1 LU 分解 (待定系数法)

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

这种通过待定系数法计算 LU 分解的算法也称为 **Doolittle 方法**.

最后, 需求解两个三角线性方程组, 即 $Ly = b$ 和 $Ux = y$. 完整求解过程如下.

算法 1.2 LU 分解求解线性方程组

计算 A 的 LU 分解 (此处省略, 可参见算法 1.1)
$y_1 = b_1$ % 向前回代求解 $Ly = b$
for $i = 2$ to n **do**
  for $k = 1$ to $i - 1$ **do**
    $b_i = b_i - a_{ik} y_k$
  end for
  $y_i = b_i$
end for
$x_n = y_n / a_{nn}$ % 向后回代求解 $Ux = y$
for $i = n - 1$ to 1 **do**
  for $k = i + 1$ to n **do**
    $y_i = y_i - a_{ik} x_k$
  end for
  $x_i = y_i / a_{ii}$
end for

列主元 LU 分解 (PLU)

$$\boxed{PA = LU},$$

其中 P 是一个置换矩阵. 因此, PA 就相当于对 A 的行进行了重新排列.

- 为了节省存储量, 我们并不存储这个置换矩阵, 而只是用一个向量来表示这个重新排列. 我们将这个向量记为 p, 其元素是 $\{1, 2, \ldots, n\}$ 的一个重排列. 比如 $p = [1, 3, 2]$ 表示交换矩阵的第 2 行和第 3 行, 而 $p = [2, 3, 1]$ 则表示将矩阵的第 2 行移到最前面, 将第 3 行移到第 2 行, 将第 1 行移到最后.
- 易知, 在计算过程中, p 的初始值为 $p = [1, 2, \ldots, n]$.
- 在选列主元的过程中, 如果出现行交换, 即交换第 i_k 行和第 k 行, 则需要相应地交换 p 的第 i_k 位置和第 k 位置上的值.

算法 1.3 PLU: 列主元 LU 分解或部分选主元 LU 分解

$p = [1, 2, \ldots, n]$ % 用于记录置换矩阵
for $k = 1$ to $n-1$ **do**
    $a_{i_k,k} = \max_{k \le i \le n} |a_{i,k}|$ % 选列主元
    if $i_k \neq k$ **then**
        for $j = 1$ to n **do**
            $a_{tmp} = a_{i_k,j},\ a_{i_k,j} = a_{k,j},\ a_{k,j} = a_{tmp}$ % 交换 A 的第 i_k 行与第 k 行
        end for
        $p_{tmp} = p_{i_k},\ p_{i_k} = p_k,\ p_k = p_{tmp}$ % 更新置换矩阵
    end if
    for $i = k+1$ to n **do**
        $a_{ik} = a_{ik}/a_{kk}$ % 计算 L 的第 i 列
        for $j = k+1$ to n **do**
            $a_{ij} = a_{ij} - a_{ik} * a_{ij}$ % 更新 $A(k+1:n, k+1:n)$
        end for
    end for
end for

完整的求解过程

算法 1.4 PLU 分解求解线性方程组

计算 A 的 PLU 分解 (此处省略, 可参见算法 1.3)
$y_1 = b_{p_1}$ % 向前回代求解 $Ly = Pb$
for $i = 2$ to n **do**
    for $j = 1$ to $i-1$ **do**
        $b_{p_i} = b_{p_i} - a_{ij}y_j$
    end for
    $y_i = b_{p_i}$
end for
$x_n = b_{p_n}/a_{nn}$ % 向后回代求解 $Ux = y$
for $i = n-1$ to 1 **do**
    for $j = i+1$ to n **do**
        $y_i = y_i - a_{ij}x_j$
    end for
    $x_i = y_i/a_{ii}$
end for

Cholesky 分解与平方根法

如果 A 是对称正定的, 则可以得到更加简洁高效的方法.

定理 (Cholesky 分解) 设 A 对称正定, 则存在唯一的对角线元素全为正的下三角矩阵 L, 使得

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

该分解称为 **Cholesky 分解**. (板书)

✍ Cholesky 分解仅针对对称正定矩阵成立.

$\mathrm{LDL}^{\intercal}$ 分解

定理 若 A 对称, 且所有顺序主子式都不为 0, 则 A 可唯一分解为

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

其中 L 为单位下三角矩阵, D 为对角矩阵. (证明留作课外练习)

如何计算 Cholesky 分解: 待定系数法

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

类似于 LU 分解, 直接比较等式两边的元素, 可得一般公式

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

根据这个计算公式即可得 l_{ij} 的表达式.

算法 1.5 Cholesky 分解 (L 存放在 A 的下三角部分)

for $j = 1$ to n **do**
    for $k = 1$ to $j-1$ **do**
        $a_{jj} = a_{jj} - a_{jk}^2$
    end for
    $a_{jj} = \sqrt{a_{jj}}$
    for $i = j+1$ to n **do**
        for $k = 1$ to $j-1$ **do**
            $a_{ij} = a_{ij} - a_{ik}a_{jk}$
        end for
        $a_{ij} = a_{ij}/a_{jj}$
    end for
end for

✍ Cholesky 分解算法的乘除运算量为 $\dfrac{1}{3}n^3 + \mathcal{O}(n^2)$, 大约为 LU 分解的一半. 另外, Cholesky 分解算法是稳定的 (稳定性与全主元 Gauss 消去法相当), 因此不

利用 Cholesky 分解求解线性方程组的方法就称为 **平方根法**, 具体描述如下:

算法 1.6 Cholesky 分解求解线性方程组

计算 Cholesky 分解 (此处省略, 参见算法 1.5)
$y_1 = b_1/a_{11}$ % 向前回代求解 $Ly = b$
for $i = 2$ to n **do**
  for $j = 1$ to $i-1$ **do**
    $b_i = b_i - a_{ij}y_j$
  end for
  $y_i = b_i/a_{ii}$
end for
$x_n = b_n/a_{nn}$ % 向后回代求解 $L^{\intercal}x = y$
for $i = n-1$ to 1 **do**
  for $j = i+1$ to n **do**
    $y_i = y_i - a_{ji}x_j$
  end for
  $x_i = y_i/a_{ii}$
end for

LDL$^\intercal$ 分解

$$\boxed{A = LDL^\intercal}, \quad L \text{ 单位下三角}, \ D \text{ 对角}.$$

- 目的: 避免在 Cholesky 分解计算平方根
- 计算方法: 由待定系数法

$$A = LDL^\intercal = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \ddots & \vdots \\ & & \ddots & l_{n,n-1} \\ & & & 1 \end{bmatrix}.$$

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

算法 1.7 改进的平方根法 (L 存放在 A 的下三角, D 存放在 A 的对角)

for $j = 1$ to n **do**
    for $k = 1$ to $j-1$ **do**
        $a_{jj} = a_{jj} - l_{jk}^2 a_{kk}$
    end for
    for $i = j+1$ to n **do**
        for $k = 1$ to $j-1$ **do**
            $a_{ij} = a_{ij} - a_{ik} a_{kk} a_{jk}$
        end for
        $a_{ij} = a_{ij}/a_{jj}$
    end for
end for
$y_1 = b_1$ % 向前回代求解 $Ly = b$
for $i = 2$ to n **do**
    for $j = 1$ to $i-1$ **do**
        $b_i = b_i - a_{ij} y_j$

    end for
    $y_i = b_i$
end for
$x_n = b_n / a_{nn}$ % 向后回代求解 $DL^{\intercal}x = y$
for $i = n-1$ to 1 **do**
    $x_i = y_i / a_{ii}$
    for $j = i+1$ to n **do**
        $x_i = x_i - a_{ji}x_j$
    end for
end for

三对角线性方程组

$$\begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}.$$

我们假定

$$|b_1| > |c_1| > 0, \quad |b_n| > |a_{n-1}| > 0, \tag{2.1}$$

且

$$|b_i| \geq |a_{i-1}| + |c_i|, \quad a_i c_i \neq 0, \quad i = 1, \ldots, n-1. \tag{2.2}$$

即 A 是**不可约 (行) 弱对角占优**的. (可以证明此时 A 是非奇异的)

计算 A 的 Crout 分解:

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & \\ a_1 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & a_{n-1} & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{bmatrix} \triangleq LU. \tag{2.3}$$

由待定系数法, 我们可以得到递推公式:

$$\alpha_1 = b_1, \quad \beta_1 = c_1/\alpha_1 = c_1/b_1,$$

$$\begin{cases} \alpha_i = b_i - a_{i-1}\beta_{i-1}, \\ \beta_i = c_i/\alpha_i = c_i/(b_i - a_{i-1}\beta_{i-1}), \quad i = 2, 3, \ldots, n-1 \end{cases}$$

$$\alpha_n = b_n - a_{n-1}\beta_{n-1}.$$

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 设三对角矩阵 A 满足条件 (2.1) 和 (2.2). 则 A 非奇异, 且

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

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

(3) $0 < |c_i| \leq |b_i| - |a_{i-1}| < |\alpha_i| < |b_i| + |a_{i-1}|, \quad i = 2, 3, \ldots, n$;

(板书)

追赶法 (Thomas 算法)

算法 1.8 追赶法 (矩阵分解与方程求解交叉进行)

$\alpha_1 = b_1$
$\beta_1 = c_1 / b_1$
$y_1 = f_1 / b_1$
for $i = 2$ to $n-1$ **do**
 $\alpha_i = b_i - a_{i-1}\beta_{i-1}$
 $\beta_i = c_i / \alpha_i$
 $y_i = (f_i - a_{i-1} y_{i-1}) / \alpha_i$ % 求解 $Ly = f$
end for
$\alpha_n = b_n - a_{n-1}\beta_{n-1}$
$y_n = (f_n - a_{n-1} y_{n-1}) / \alpha_n$
$x_n = y_n$
for $i = n-1$ to 1 **do**
 $x_i = y_i - \beta_i x_{i+1}$
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.

列对角占优情形

我们也可以考虑下面的分解

$$A=\begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix}=\begin{bmatrix} 1 & & & \\ \gamma_1 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_{n-1} & 1 \end{bmatrix}\begin{bmatrix} \alpha_1 & c_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & \alpha_n \end{bmatrix}. \tag{2.4}$$

但此时 $|\gamma_i|$ 可能大于 1. 比如 $\gamma_1=a_1/b_1$, 因此当 $|b_1|<|a_1|$ 时, $|\gamma_1|>1$. 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 (2.3);

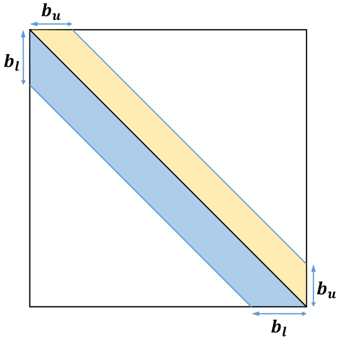
如果 A 是列对角占优, 则采用分解 (2.4).

带状线性方程组

在实际应用中会经常遇到带状线性方程组, 在设计算法时应该充分利用其中的零元素来减少运算量, 从而大幅度地提高计算效率.

带状矩阵: 下带宽为 b_l, 上带宽为 b_u

$$a_{ij} = 0 \quad \text{for} \quad i > j + b_l \text{ or } i < j - b_u$$



带状矩阵 LU 分解的性质

定理 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.

部分选主元 LU 分解的性质

定理 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. (也就是说, 非零元素的个数有限制, 但位置不限)