FDU 数值算法 1. 稠密线性方程组的直接解法

本文根据邵老师授课内容整理而成，并参考了以下教材:

- 数值线性代数 (第二版, 徐树方, 高立, 张平文) 第 1 & 2 章

欢迎批评指正!

1.1 扰动分析

1.1.1 条件数

考虑线性方程组 $Ax = b$，其中系数矩阵 $A \in \mathbb{C}^{n\times n}$ 非奇异，右端向量 $b \in \mathbb{C}^n$ 非零.
我们关心的问题是: 当 A 和 b 发生扰动时，线性方程组的解 x 将受到怎样的影响?

(数值线性代数, 定理 $2.2.1$)
设 $A \in \mathbb{C}^{n\times n}$ 非奇异，$b \in \mathbb{C}^n$ 非零，$\|\cdot\|$ 是 $\mathbb{C}^{n\times n}$ 上的一个满足 $\|I\| = 1$ 的矩阵范数.
若扰动 $\Delta A \in \mathbb{C}^{n\times n}$ 满足 $\|A^{-1}\| \cdot \|\Delta A\| < 1$，且

$$\begin{cases} Ax = b \\ (A + \Delta A)(x + \Delta x) = b + \Delta b, \end{cases}$$

则 $A + \Delta A$ 也是非奇异的，且有:

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

- **Lemma (数值线性代数, 推论 $2.1.1$):**
  设 $\|\cdot\|$ 是 $\mathbb{C}^{n\times n}$ 上的一个满足条件 $\|I\| = 1$ 的矩阵范数 (例如谱范数 $\|\cdot\|_2$).
  若 $A \in \mathbb{C}^{n\times n}$ 满足 $\|A\| < 1$,
  则 $I - A$ 可逆 (因为 $\|I - A\| \ge |\|I\| - \|A\|| = |1 - \|A\|| > 0$),
  且根据 Neumann 级数我们有:

$$\begin{aligned} \|(I - A)^{-1}\| &= \left\| I + A + A^2 + \cdots \right\| && (\text{Neumann series}) \\ &\le \|I\| + \|A\| + \|A\|^2 + \cdots && (\text{note that } \|I\| = 1) \\ &= 1 + \|A\| + \|A\|^2 + \cdots \\ &= \frac{1}{1 - \|A\|}. \end{aligned}$$

证明:
注意到 $\| - A^{-1}\Delta A\| \le \|A^{-1}\| \cdot \|\Delta A\| < 1.$
根据 **Lemma** 可知 $I + A^{-1}\Delta A$ 可逆 (因此 $A + \Delta A = A(I + A^{-1}\Delta A)$ 也是可逆的)，且有:

$$\|(I + A^{-1}\Delta A)^{-1}\| \le \frac{1}{1 - \| - A^{-1}\Delta A\|} \le \frac{1}{1 - \|A^{-1}\| \cdot \|\Delta A\|}.$$

根据

$$\begin{cases} Ax = b \\ (A + \Delta A)(x + \Delta x) = b + \Delta b \end{cases}$$

可知 $(A + \Delta A)\Delta x = \Delta b - \Delta A \cdot x$，于是我们有:

$$\begin{aligned} \Delta x &= (A + \Delta A)^{-1}(\Delta b - \Delta A \cdot x) \\ &= (I + A^{-1}\Delta A)^{-1} A^{-1}(\Delta b - \Delta A \cdot x), \end{aligned}$$

进而有:

$$\begin{aligned} \|\Delta x\| &= \|(I + A^{-1}\Delta A)^{-1} A^{-1}(\Delta b - \Delta A \cdot x)\| \\ &\le \|(I + A^{-1}\Delta A)^{-1}\| \cdot \|A^{-1}\| \cdot \|\Delta b - \Delta A \cdot x\| \\ &\le \frac{1}{1 - \|A^{-1}\| \cdot \|\Delta A\|} \cdot \|A^{-1}\| \cdot (\|\Delta b\| + \|\Delta A\| \cdot \|x\|). \end{aligned}$$

注意到 $Ax = b$ 的解 x 一定不是零向量，因此我们有:

$$
\begin{aligned}
\frac{\|\Delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\cdot\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|x\|}+\|\Delta A\|\right) \\
&= \frac{\|A^{-1}\|\cdot\|A\|}{1-\|A^{-1}\|\cdot\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|A\|\cdot\|x\|}+\frac{\|\Delta A\|}{\|A\|}\right) \quad (\text{note that } \|b\|=\|Ax\|\leq\|A\|\cdot\|x\|) \\
&\leq \frac{\|A^{-1}\|\cdot\|A\|}{1-\|A^{-1}\|\cdot\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\|\Delta A\|}{\|A\|}\right).
\end{aligned}
$$

当 $\|\Delta A\|$ 足够小时，上述定理中的不等式可近似表示为:

$$
\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A)\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\|\Delta A\|}{\|A\|}\right),
$$

其中 $\kappa(A):=\|A^{-1}\|\cdot\|A\|$ 为线性方程组 $Ax=b$ 的**条件数** (condition number).
若 $\kappa(A)$ 很小，则我们称 $Ax=b$ 的求解问题是**良态的** (well-conditioned);
若 $\kappa(A)$ 很大，则我们称 $Ax=b$ 的求解问题是**病态的** (ill-conditioned).

值得注意的是，条件数 $\kappa(A)$ 依赖于所选用的矩阵范数 $\|\cdot\|$.
然而，由于有限维向量空间中所有范数都是等价的，可以推知:
如果一个问题在某一矩阵范数下表现为病态，那么在其他矩阵范数下它同样是病态的.

在理论分析中，我们通常使用谱范数 $\|\cdot\|_2$ 定义的条件数 $\kappa_2(A)=\|A^{-1}\|_2\|A\|_2$.
注意到对于任意酉矩阵 $U\in\mathbb{C}^{n\times n}$，我们都有 $\kappa_2(U)=\|U^{-1}\|_2\|U\|_2=1\cdot 1=1$,
因此系数矩阵为酉矩阵的线性方程组是良态的.

1.1.2 病态性的直观理解

在线性方程组 $Ax=b$ 中，系数矩阵 $A\in\mathbb{C}^{n\times n}$ 越接近奇异，问题就越病态.
考虑极限情况，假设 A 是奇异矩阵，给定特解 x_0 满足 $Ax_0=b$,
对于任意 $\Delta x\in\mathrm{Ker}(A)\backslash\{0_n\}$，我们都有 $A(x_0+\Delta x)=b+0_n=b$.
这说明，即使 A,b 什么扰动都不加，问题的解都会面目全非.

再从奇异值分解的角度看待这个问题.
设 $A\in\mathbb{C}^{n\times n}$ 的奇异值分解为 $A=U\Sigma V^{\mathrm{H}}=\sum_{i=1}^{n}u_i\sigma_i v_i^{\mathrm{H}}$,
其中 $U=[u_1,\ldots,u_n],V=[v_1,\ldots,v_n]$ 为酉矩阵,
对角阵 $\Sigma=\mathrm{diag}\{\sigma_1,\ldots,\sigma_n\}$ 的对角元为 A 的奇异值，满足 $\sigma_1\geq\cdots\geq\sigma_n$.
若 A 非奇异，则其奇异值 $\sigma_1,\ldots,\sigma_n$ 均为正实数，于是对角阵 Σ 可逆.
此时我们有:

$$
\begin{aligned}
A^{-1} &= (U\Sigma V^{\mathrm{H}})^{-1} \\
&= V^{-\mathrm{H}}\Sigma^{-1}U^{\mathrm{H}} \\
&= V\Sigma^{-1}U^{\mathrm{H}} \\
&= \sum_{i=1}^{n} v_i\sigma_i^{-1}u_i^{\mathrm{H}}.
\end{aligned}
$$

取 $\Delta x=v_n$，则可对右端向量 b 构造如下扰动:

$$
\begin{cases}
\Delta b = A\Delta x = Av_n = u_n\sigma_n \\
\|\Delta x\| = \|v_n\| = 1 \\
\|\Delta b\| = \|u_n\sigma_n\| = \sigma_n\|u_n\| = \sigma_n.
\end{cases}
$$

这表明问题的解对右端向量 b 的扰动的敏感程度是与 σ_n 有关的.
系数矩阵 A 越接近奇异，最小奇异值 σ_n 就越接近于 0,
解 x 对右端向量 b 的扰动就越敏感，求解 $Ax=b$ 的问题就越病态.

事实上，在谱范数下定义的条件数具有更直观的几何意义.
(Kahan 公式, 数值线性代数, 定理 $2.2.2$)
若 $A\in\mathbb{C}^{n\times n}$ 非奇异，则我们有:

$$
\min_{\det(A+\Delta A)=0}\|\Delta A\|_2=\frac{1}{\|A^{-1}\|_2}=\sigma_n,
$$

其中 $\sigma_n > 0$ 为 A 的最小奇异值.

- **Insight:**
  Kahan 公式表明非奇异阵 A 与全体奇异阵的最小距离可以用 A 的最小奇异值 σ_n 来刻画.
  最小奇异值 σ_n 越小，A 就越接近奇异，即越容易通过一个小扰动变成奇异阵.

  将 Kahan 公式等价表示为:

  $$\min_{\det(A+\Delta A)=0} \frac{\|\Delta A\|_2}{\|A\|_2} = \frac{1}{\|A\|_2\|A^{-1}\|_2} = \frac{1}{\kappa(A)},$$

  这表明条件数的倒数 $1/\kappa(A)$ 刻画了非奇异阵 A 与全体奇异阵的最小相对距离.
  因此当 A 的条件数 $\kappa(A)$ 很大时，A 就已经接近奇异了.

证明:
若 A 非奇异，则对于足够小的 ΔA，扰动后的矩阵 $A+\Delta A$ 仍是非奇异的.
具体来说，根据 **数值线性代数 推论** $2.1.1$ 可知:
只要 $\| -A^{-1}\Delta A\|_2 \le \|A^{-1}\|_2 \cdot \|\Delta A\|_2 < 1$,
矩阵 $I + A^{-1}\Delta A$ 就是非奇异的，于是 $A+\Delta A = A(I+A^{-1}\Delta A)$ 也是非奇异的.
因此我们有:

$$\inf_{\det(A+\Delta A)=0} \|\Delta A\|_2 \ge \frac{1}{\|A^{-1}\|_2}.$$

下面我们证明上述不等式是取等的.
根据谱范数的定义我们有 $\|A^{-1}\|_2 = \sup_{\|x\|_2=1}\|A^{-1}x\|_2$,
故存在满足 $\|x\|_2 = 1$ 的 $x \in \mathbb{C}^n$ 使得 $\|A^{-1}x\|_2 = \|A^{-1}\|_2$.
若构造:

$$y := \frac{A^{-1}x}{\|A^{-1}x\|_2}$$
$$\Delta A := -\frac{xy^{\mathrm{T}}}{\|A^{-1}\|_2},$$

则我们有:

$$\begin{aligned}(A+\Delta A)y &= Ay + \Delta A \cdot y \\ &= A\frac{A^{-1}x}{\|A^{-1}x\|_2} - \frac{xy^{\mathrm{T}}y}{\|A^{-1}\|_2} \quad (\text{note that } \|y\|_2 = 1) \\ &= \frac{x}{\|A^{-1}x\|_2} - \frac{x}{\|A^{-1}\|_2} \quad (\text{note that } \|A^{-1}x\|_2 = \|A^{-1}\|_2) \\ &= 0_n\end{aligned}$$

由于 y 是非零向量，故 $A+\Delta A$ 是奇异矩阵.
因此我们有:

$$\max_{\det(A+\Delta A)=0} \|\Delta A\|_2 = \frac{1}{\|A^{-1}\|_2}.$$

值得说明的是，我们并非拿病态问题毫无办法.
例如系数矩阵 $A = \mathrm{diag}\{1, 2, \dots, 2^n\}$ 的线性方程组非常病态 ($\kappa(A) = 2^n$),
但只要我们不给它在非对角元上扰动的机会，那么这个病态问题还是解得准的.
可如果非对角元上的扰动控制不住，那就真的完蛋了.

1.2 三角方程组的解法

三角方程组易于求解，它是用分解方法求解稠密线性方程组的基础.

1.2.1 前代法

考虑下三角方程组 $Ly = b$.

其中 $L \in \mathbb{C}^{n\times n}$ 为已知的非奇异下三角阵 (满足 $\begin{cases} l_{i,j} = 0 \ (\forall\, i < j) \\ l_{i,i} \neq 0 \ (i = 1, \ldots, n) \end{cases}$), $b \in \mathbb{C}^n$ 为已知的向量.

$$Ly = \begin{bmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & \vdots & \ddots & \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = b \tag{1}$$

- 由第一个方程 $l_{1,1}y_1 = b_1$ 可知 $y_1 = b_1/l_{1,1}$.
- 由第二个方程 $l_{2,1}y_1 + l_{2,2}y_2 = b_2$ 可知 $y_2 = (b_2 - l_{2,1}y_1)/l_{2,2}$.
- 一般地，如果我们已经求出 $y_1, \ldots, y_{i-1}$，
  则可根据 (1) 的第 i 个方程 $l_{i,1}y_1 + \cdots + l_{i,i-1}y_{i-1} + l_{i,i}y_i = b_i$ 得到 $y_i = (b_i - \sum_{j=1}^{i-1} l_{i,j}y_j)/l_{i,i}$.

这种求解下三角方程组的方法称为前代法.

在实际计算中，我们将 y_i 存放在 b_i 所用的存储单元中，并调整运算次序.

(前代法, 数值线性代数, 算法 1.1.1)

$$\begin{aligned} &\text{function} : y = \text{ForwardSweep}[L, b] \\ &\quad n = \text{length}(b) \\ &\quad \text{for } i = 1 : n-1 \\ &\qquad b(i) = b(i)/L(i,i) \\ &\qquad b(i+1:n) = b(i+1:n) - b(i)L(i+1:n, i) \\ &\quad \text{end} \\ &\quad b(n) = b(n)/L(n,n) \\ &\quad \text{return } b \end{aligned}$$

最终 $Ly = b$ 的解 y 存储在 b 中.

第 $1 \le i \le n-1$ 步浮点运算次数为 $1 + (n-i) + (n-i) = 2(n-i) + 1$,

最后一步浮点运算次数为 1.

因此总浮点运算次数为:

$$\begin{aligned} \left\{\sum_{i=1}^{n-1}(2(n-i)+1)\right\} + 1 &= \sum_{k=1}^{n-1}(2k+1) + 1 \\ &= (n-1)\cdot\frac{3+2n-1}{2} + 1 \\ &= n^2 - 1 + 1 \\ &= n^2 \\ &= O(n^2) \end{aligned}$$

其 MATLAB 代码如下:

```
function y = Forward_Sweep(L, b)
    % 前代法求解 Ly = b
    n = length(b);
    for i = 1:n-1
        b(i) = b(i) / L(i, i);  % 对角线归一化
        b(i+1:n) = b(i+1:n) - b(i) * L(i+1:n, i);  % 消去
    end
    b(n) = b(n) / L(n, n);  % 处理最后一列
    y = b;  % 返回结果
end
```

1.2.2 回代法

考虑上三角方程组 $Ux = y$.

其中 $U \in \mathbb{C}^{n\times n}$ 为已知的非奇异上三角阵 (满足 $\begin{cases} u_{i,j} = 0 \ (\forall\, i > j) \\ u_{i,i} \neq 0 \ (i = 1, \ldots, n) \end{cases}$), $y \in \mathbb{C}^n$ 为已知的向量.

$$Ux = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ & u_{2,2} & \cdots & u_{2,n} \\ & & \ddots & \vdots \\ & & & u_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y \tag{2}$$

- 由第 n 个方程 $u_{n,n}x_n = y_n$ 可知 $x_n = y_n/u_{n,n}$.
- 由第 $n-1$ 个方程 $u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = y_{n-1}$ 可知 $x_{n-1} = (y_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1}$.
- 一般地，如果我们已经求出 $x_{i+1}, \ldots, x_n$，
  则可根据 (2) 的第 i 个方程 $u_{i,i}x_i + \cdots + u_{i,n-1}x_{n-1} + u_{i,n}x_n = y_i$ 得到 $x_i = (y_i - \sum_{j=i+1}^{n} u_{i,j}x_j)/u_{i,i}$.

这种求解上三角方程组的方法称为回代法.

在实际计算中，我们将 x_i 存放在 y_i 所用的存储单元中，并调整运算次序.

(回代法, 数值线性代数, 算法 1.1.2)

$$\begin{aligned} &\text{function: } x = \text{BackwardSweep}[U, y] \\ &\quad n = \text{length}(y) \\ &\quad \text{for } i = n : -1 : 2 \\ &\qquad y(i) = y(i)/U(i,i) \\ &\qquad y(1 : i-1) = y(1 : i-1) - y(i)U(1 : i-1, i) \\ &\quad \text{end} \\ &\quad y(1) = y(1)/U(1,1) \\ &\quad \text{return } y \end{aligned}$$

最终 $Ux = y$ 的解 x 存储在 y 中.

第 $2 \le i \le n$ 步浮点运算次数为 $1 + (i-1) + (i-1) = 2i - 1$,

最后一步浮点运算次数为 1.

因此总浮点运算次数为:

$$\begin{aligned} \sum_{i=2}^{n}(2i-1) + 1 &= \sum_{k=1}^{n-1}(2k+1) + 1 \\ &= (n-1) \cdot \frac{3 + 2n - 1}{2} + 1 \\ &= n^2 - 1 + 1 \\ &= n^2 \\ &= O(n^2) \end{aligned}$$

其 MATLAB 代码如下:

```
function x = Backward_Sweep(U, y)
    % 回代法求解 Ux = y
    n = length(y);
    for i = n:-1:2
        y(i) = y(i) / U(i, i);  % 对角线归一化
        y(1:i-1) = y(1:i-1) - y(i) * U(1:i-1, i);  % 消去
    end
    y(1) = y(1) / U(1, 1);  % 处理第一行
    x = y;  % 返回结果
end
```

1.2.3 舍入误差分析

> **(邵言邵语)**
> **凡是我们算出的解，我们都坚决维护;**
> **凡是输入的数据，我们都可以把脏水泼上去.**

(数值线性代数, 引理 2.4.2)
若下三角阵 $L=[l_{ij}]\in\mathbb{R}^{n\times n}$ 非奇异 (即满秩),
则使用前代法求解 $Ly=b$ 得到的 $\tilde{y}$ 满足 $\begin{cases}(L+\Delta_L)\tilde{y}=b\\|\Delta_L|\le\gamma_n|L|\end{cases}$
其中 γ_n 代表 n 层浮点运算的累积相对误差的绝对值.

- 上述引理很容易推广至上三角阵和回代法.
- **证明:** 对 n 使用数学归纳法.
  当 $n=1$ 时，命题显然成立.
  假设命题对于所有 $n-1$ 阶下三角方程组都成立，现在考虑 n 阶的情形.

  将 L,b 和 $\tilde{y}$ 分块为:

$$L=\begin{bmatrix}l_{11}&\\l_1&L_1\end{bmatrix}\qquad\tilde{y}=\begin{bmatrix}\tilde{y}_1\\\tilde{x}\end{bmatrix}\qquad b=\begin{bmatrix}b_1\\c\end{bmatrix}$$

  对于 $\tilde{y}$ 的第 1 个分块，我们有 $\tilde{y}_1=\mathrm{fl}(\frac{b_1}{l_{11}})=\frac{b_1}{l_{11}(1+\delta_1)}\quad(|\delta_1|\le\mathrm{eps})$
  对于 $\tilde{y}$ 的第 2 个分块，由归纳假设我们有 $\begin{cases}(L_1+\Delta_1)\tilde{x}=\mathrm{fl}(c-\mathrm{fl}(\tilde{y}_1l_1))\\|\Delta_1|\le\gamma_{n-1}|L_1|\end{cases}$

  关于 $\mathrm{fl}(c-\mathrm{fl}(\tilde{y}_1l_1))$ 我们有:

$$\begin{aligned}\mathrm{fl}(c-\mathrm{fl}(\tilde{y}_1l_1))&=\begin{bmatrix}(b_2-\tilde{y}_1l_{21}(1+\delta_2))\cdot\frac{1}{1+\varepsilon_2}\\\vdots\\(b_n-\tilde{y}_1l_{n1}(1+\delta_n))\cdot\frac{1}{1+\varepsilon_n}\end{bmatrix}\quad\left(\text{note that } c=\begin{bmatrix}b_2\\\vdots\\b_n\end{bmatrix}\text{ and } l_1=\begin{bmatrix}l_{21}\\\vdots\\l_{n1}\end{bmatrix}\right)\\&=(I+D_\varepsilon)^{-1}[c-(I+D_\delta)\tilde{y}_1l_1]\end{aligned}$$

  其中 $\begin{cases}D_\varepsilon=\mathrm{diag}(\varepsilon_2,\dots,\varepsilon_n)\\D_\delta=\mathrm{diag}(\delta_2,\dots,\delta_n)\\|\varepsilon_i|,|\delta_i|\le\mathrm{eps}\qquad(i=2,\dots,n)\end{cases}$
  因此我们有 $(L_1+\Delta_1)\tilde{x}=\mathrm{fl}(c-\mathrm{fl}(\tilde{y}_1l_1))=(I+D_\varepsilon)^{-1}[c-(I+D_\delta)\tilde{y}_1l_1]$
  于是 $c=(I+D_\varepsilon)(L_1+\Delta_1)\tilde{x}+(I+D_\delta)\tilde{y}_1l_1$
  联立 $\begin{cases}l_{11}(1+\delta_1)\cdot\tilde{y}_1=b_1\\(I+D_\delta)l_1\cdot\tilde{y}_1+(I+D_\varepsilon)(L_1+\Delta_1)\cdot\tilde{x}=c\end{cases}$ 可知:

$$(L+\Delta_L)\tilde{y}=\begin{bmatrix}l_{11}(1+\delta_1)&\\(I+D_\delta)l_1&(I+D_\varepsilon)(L_1+\Delta_1)\end{bmatrix}\begin{bmatrix}\tilde{y}_1\\\tilde{x}\end{bmatrix}=\begin{bmatrix}b_1\\c\end{bmatrix}=b$$

  其中我们记 $\Delta_L=\begin{bmatrix}\delta_1l_{11}&\\D_\delta l_1&(I+D_\varepsilon)\Delta_1+D_\varepsilon L_1\end{bmatrix}$
  对于 Δ_L 我们有:

$$\begin{aligned}|\Delta_L|&=\begin{bmatrix}|\delta_1l_{11}|&\\|D_\delta l_1|&|(I+D_\varepsilon)\Delta_1+D_\varepsilon L_1|\end{bmatrix}\\&\le\begin{bmatrix}|\delta_1||l_{11}|&\\|D_\delta||l_1|&|\Delta_1|+|D_\varepsilon|(|\Delta_1|+|L_1|)\end{bmatrix}\\&\le\begin{bmatrix}\mathrm{eps}|l_{11}|&\\\mathrm{eps}|I||l_1|&\gamma_{n-1}|L_1|+\mathrm{eps}|I|(\gamma_{n-1}|L_1|+|L_1|)\end{bmatrix}\\&=\begin{bmatrix}\mathrm{eps}|l_{11}|&\\\mathrm{eps}|l_1|&[\gamma_{n-1}+\mathrm{eps}(\gamma_{n-1}+1)]|L_1|\end{bmatrix}\quad(\text{note that }\gamma_{n-1}+\mathrm{eps}(\gamma_{n-1}+1)\approx(1+\mathrm{eps})(1+\gamma_{n-1})=\gamma_n)\\&\approx\begin{bmatrix}\mathrm{eps}|l_{11}|&\\\mathrm{eps}|l_1|&\gamma_n|L_1|\end{bmatrix}\\&\le\gamma_n|L|\end{aligned}$$

  这样我们就有 $\begin{cases}(L+\Delta_L)\tilde{y}=b\\|\Delta_L|\le\gamma_n|L|\end{cases}$，命题得证.

1.3 三角分解

1.3.1 Gauss 变换

我们想将给定矩阵 A 分解为一个下三角阵 L 和一个上三角阵 U 的乘积，
这样求解 $Ax=(LU)x=b$ 可以分两步进行:

- 用前代法解 $Ly=b$ 得到 y
- 用回代法解 $Ux=y$ 得到 x

得到 $A=LU$ 的最自然的做法是通过一系列的初等变换，
逐步将 A 变换为一个上三角阵 U，同时保证变换矩阵的乘积是一个下三角阵 L.

给定 $x\in\mathbb{C}^n$，考虑初等下三角阵

$$L_k=I-l_ke_k^{\mathrm{T}}=\begin{bmatrix}1&&&&\\&\ddots&&&\\&&1&&\\&&-l_{k+1,k}&1&\\&&\vdots&&\ddots\\&&-l_{n,k}&&1\end{bmatrix},$$

其中 $l_k=[0,\dots,0,l_{k+1,k},\dots,l_{n,k}]^{\mathrm{T}}$，$e_k$ 是 $\mathbb{R}^n$ 的第 k 个单位基向量.
我们希望

$$L_kx=\begin{bmatrix}x_1\\\vdots\\x_k\\x_{k+1}-x_kl_{k+1,k}\\\vdots\\x_n-x_nl_{n,k}\end{bmatrix}$$

的第 $k+1$ 至第 n 个分量均为零，
因此可令 $l_{i,k}=x_i/x_k\ (i=k+1,\dots,n)$ (当然我们要求 $x_k\neq0$)，即得到

$$l_k=\frac{1}{x_k}\begin{bmatrix}0\\\vdots\\0\\x_{k+1}\\\vdots\\x_n\end{bmatrix},$$

称为 **Gauss 向量**.
而初等下三角阵

$$L_k=I-l_ke_k^{\mathrm{T}}=\begin{bmatrix}1&&&&\\&\ddots&&&\\&&1&&\\&&-\frac{x_{k+1}}{x_k}&1&\\&&\vdots&&\ddots\\&&-\frac{x_n}{x_k}&&1\end{bmatrix}$$

称为 **Gauss 变换矩阵**.

Gauss 变换 $L_k=I-l_ke_k^{\mathrm{T}}$ 具有许多良好的性质:

- 其逆矩阵 $L_k^{-1}=I+l_ke_k^{\mathrm{T}}$，这是因为 $e_k^{\mathrm{T}}l_k=0$，于是有

$$L_k(I+l_ke_k^{\mathrm{T}})=(I-l_ke_k^{\mathrm{T}})(I+l_ke_k^{\mathrm{T}})=I-l_ke_k^{\mathrm{T}}l_ke_k^{\mathrm{T}}=I.$$

- Gauss 变换作用于一个矩阵相当于对其进行秩 1 修正.
  设 $A \in \mathbb{C}^{n\times n}$，则我们有 $L_kA = (I - l_ke_k^{\mathrm{T}})A = A - l_k(e_k^{\mathrm{T}}A)$.

1.3.2 Gauss 消去法

给定方阵 $A \in \mathbb{C}^{n\times n}$.
在一定条件下，我们可以构造 $n-1$ 个 Gauss 变换 $L_1, \dots, L_{n-1}$，使得 $L_{n-1}\cdots L_1A$ 为上三角阵.

记 $A^{(0)} = A$，并假定已求出 $k-1$ $(k < n)$ 个 Gauss 变换 $L_1, \dots, L_{k-1} \in \mathbb{R}^{n\times n}$ 使得

$$A^{(k-1)} = L_{k-1}\cdots L_1A = \begin{bmatrix} A_{1,1}^{(k-1)} & A_{1,2}^{(k-1)} \\ 0_{(k-1)\times(n-k+1)} & A_{2,2}^{(k-1)} \end{bmatrix},$$

其中 $A_{1,1}^{(k-1)}$ 是 $k-1$ 阶上三角阵，记 $A^{(k-1)}$ 的 $(2,2)$ 分块为

$$A_{2,2}^{(k-1)} = \begin{bmatrix} a_{k,k}^{(k-1)} & \dots & a_{k,n}^{(k-1)} \\ \vdots & & \vdots \\ a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} \end{bmatrix}.$$

若 $a_{k,k}^{(k-1)} \neq 0$，则我们又可以确定一个 Gauss 变换 L_k，
使得 $L_kA^{(k-1)}$ 的第 k 列的最后 $n-k$ 个元素均为 0:

$$l_k = \frac{1}{a_{k,k}^{(k-1)}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{k+1,k}^{(k-1)} \\ \vdots \\ a_{n,k}^{(k-1)} \end{bmatrix}, \quad L_k = I - l_ke_k^{\mathrm{T}} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\frac{a_{k+1,k}^{(k-1)}}{a_{k,k}^{(k-1)}} & 1 & \\ & & \vdots & & \ddots \\ & & -\frac{a_{n,n}^{(k-1)}}{a_{k,k}^{(k-1)}} & & 1 \end{bmatrix}.$$

如此进行 $n-1$ 步，最终得到 $A^{(n-1)}$ 和 $L_1, L_2, \dots, L_{n-1}$ (其中 $A^{(n-1)} = L_{n-1}\cdots L_2L_1A^{(0)}$).
我们记

$$\begin{cases} L := (L_{n-1}\cdots L_2L_1)^{-1} \\ U := A^{(n-1)}, \end{cases}$$

则有 $LU = A^{(0)} = A$.

> 以 $n = 4$ 的情况为例:

$$
\begin{aligned}
A &= \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & \\ * & 1 & & \\ * & & 1 & \\ * & & & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & \\ * & 1 & & \\ * & & 1 & \\ * & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & * & 1 & \\ & * & & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & \\ * & 1 & & \\ * & & 1 & \\ * & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & * & 1 & \\ & * & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & * & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \end{bmatrix} \\
&= L_3^{-1} L_2^{-1} L_1^{-1} U \\
&= LU
\end{aligned}
$$

根据 Gauss 变换的特点，我们有:

$$
\begin{aligned}
L &= (L_{n-1} \cdots L_2 L_1)^{-1} \\
&= L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} \\
&= (I + l_1 e_1^{\mathrm{T}})(I + l_2 e_2^{\mathrm{T}}) \cdots (I + l_{n-1} e_{n-1}^{\mathrm{T}}) \quad (\text{note that } e_j^{\mathrm{T}} l_i = 0 \text{ for all } j < i) \\
&= I + l_1 e_1^{\mathrm{T}} + l_2 e_2^{\mathrm{T}} + \cdots + l_{n-1} e_{n-1}^{\mathrm{T}}
\end{aligned}
$$

即 L 是形如

$$
L = I + [l_1, l_2, \ldots, l_{n-1}, 0] = \begin{bmatrix} 1 & & & & \\ l_{2,1} & 1 & & & \\ l_{3,1} & l_{3,2} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n,1} & l_{n,2} & l_{n,3} & \ldots & 1 \end{bmatrix}
$$

的**单位下三角阵**.

上述计算三角分解的方法称为 **Gauss 消去法**.

- L_k 作用于 $A^{(k-1)}$ 得到 $A^{(k)}$,
  即 $A^{(k)} = L_k A^{(k-1)} = (I - l_k e_k^{\mathrm{T}}) A^{(k-1)} = A^{(k-1)} - l_k e_k^{\mathrm{T}} A^{(k-1)}$.
  注意到 $e_k^{\mathrm{T}} A^{(k-1)}$ 是 $A^{(k-1)}$ 的第 k 行，且 l_k 的前 k 个分量为 0，因此我们有:
  - $A^{(k)}$ 和 $A^{(k-1)}$ 前 k 行完全相同
  - $$
    \begin{cases} a_{i,k}^{(k)} = 0 & (\forall\ i = k+1, \ldots, n) \\ a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - l_{i,k} a_{k,j}^{(k-1)} = a_{i,j}^{(k-1)} - \frac{1}{a_{k,k}^{(k-1)}} a_{i,k}^{(k-1)} a_{k,j}^{(k-1)} & (\forall\ i, j = k+1, \ldots, n) \end{cases}
    $$
- **$A^{(k)}$ 和 L_k 的存储:**
  首先，$A^{(k-1)}$ 的第 $k+1$ 行至第 n 行的元素在计算出 $A^{(k)}$ 后不再有用，
  因此 $A^{(k)}$ 在 $(k+1:n, k+1:n)$ 位置的元素可覆写到 $A^{(k-1)}$ 中的相应位置.
  其次，由于 $A^{(k)}$ 的第 k 列对角元以下的元素 $a_{i,k}^{(k)}$ $(i = k+1, \ldots, n)$ 为零，无需储存，
  故我们可将 l_k 的非零元存储在这些位置上.
  例如一个 4×4 的矩阵 A 经过两步 Gauss 消元后，其形式为

$$
\begin{bmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & a_{1,4}^{(0)} \\ l_{2,1} & a_{2,2}^{(1)} & a_{2,3}^{(1)} & a_{2,4}^{(1)} \\ l_{3,1} & l_{3,2} & a_{3,3}^{(2)} & a_{3,4}^{(2)} \\ l_{4,1} & l_{4,2} & a_{4,3}^{(2)} & a_{4,4}^{(2)} \end{bmatrix}.
$$

(Gauss 消去法, 数值线性代数, 算法 $1.1.3$)

$$
\begin{aligned}
&\text{function: } [L,U]=\text{GaussianElimination}(A)\\
&\quad n=\dim(A)\\
&\quad \text{for } k=1:n-1\\
&\qquad A(k+1:n,k)=A(k+1:n,k)/A(k,k)\\
&\qquad A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-A(k+1:n,k)A(k,k+1:n)\\
&\quad \text{end}\\
&\quad L=I_n+A\odot(\text{strictly lower triangular matrix with all ones})\\
&\quad U=A\odot(\text{upper triangular matrix with all ones})\\
&\quad \text{return } [L,U]
\end{aligned}
$$

其中 $\odot$ 代表 Hadamard 乘积，即逐元素乘积.
总浮点运算数为:

$$
\begin{aligned}
\sum_{i=1}^{n-1}\left((n-i)+2(n-i)^2\right)&=\sum_{k=1}^{n-1}(k+2k^2)\\
&=\frac{1}{2}(n-1)n+2\cdot\frac{1}{6}(n-1)n(2n-1)\\
&=\frac{2}{3}n^3-\frac{1}{2}n^2-\frac{1}{6}n\\
&=O\left(n^3\right)
\end{aligned}
$$

Gauss 消去法的 MATLAB 实现参见 **Homework 01 Problem 05**.

显然当且仅当前 $n-1$ 个主元 $a_{k,k}^{(k-1)}$ $(k=1,\dots,n-1)$ 均不为零时，Gauss 消去法才能进行到底.
我们自然要问: 矩阵 A 满足什么条件，才能保证前 $n-1$ 个主元均不为零?

(前 k 个主元不为零的充要条件, 数值线性代数, 定理 $1.1.1$)
给定 $k\in\{1,\dots,n-1\}$.
对 A 进行 Gauss 消去法过程中的前 k 个主元 $a_{i,i}^{(i-1)}$ $(i=1,\dots,k)$ 均不为零，
当且仅当 A 的前 k 个顺序主子阵 A_i $(i=1,\dots,k)$ 都是非奇异的.

- **推论 (三角分解存在且唯一的充分条件, 数值线性代数, 定理 $1.1.2$):**
  若 $A\in\mathbb{C}^{n\times n}$ 的前 $n-1$ 个顺序主子阵 A_k $(k=1,\dots,n-1)$ 都是非奇异的，
  则存在唯一的一对单位下三角阵 $L\in\mathbb{C}^{n\times n}$ 和上三角阵 $U\in\mathbb{C}^{n\times n}$ 使得 $A=LU$.

证明:
对 k 应用数学归纳法.
当 $k=1$ 时，$A_1=[a_{1,1}^{(0)}]$，命题显然成立.

假设命题直至 $k-1$ 成立，且 $A_1,\dots,A_{k-1}$ 非奇异.
根据归纳假设可知 $a_{i,i}^{(i-1)}\neq 0$ $(i=1,\dots,k-1)$.
因此 Gauss 消去过程至少可以进行 $k-1$ 步，得到

$$
A^{(k-1)}=L_{k-1}\cdots L_1A=\begin{bmatrix}A_{1,1}^{(k-1)} & A_{1,2}^{(k-1)}\\ & A_{2,2}^{(k-1)}\end{bmatrix},
$$

其中 $A_{1,1}^{(k-1)}$ 是 $k-1$ 阶上三角阵.

因此 $A^{(k-1)}$ 的 k 阶顺序主子阵具有形状

$$
\begin{bmatrix}A_{1,1}^{(k-1)} & *\\ & a_{k,k}^{(k-1)}\end{bmatrix},
$$

其中 $a_{k,k}^{(k-1)}$ 是 $A_{2,2}^{(k-1)}$ 左上角的元素.

记 $L_1,\dots,L_{k-1}$ 的 k 阶顺序主子阵为 $\widetilde{L}_1,\dots,\widetilde{L}_{k-1}$，则我们有

$$
\widetilde{L}_{k-1}\cdots\widetilde{L}_1A_k=\begin{bmatrix}A_{1,1}^{(k-1)} & *\\ & a_{k,k}^{(k-1)}\end{bmatrix}.
$$

注意到 $L_1, \dots, L_{k-1}$ 均为单位下三角阵，
故 $\widetilde{L}_1, \dots, \widetilde{L}_{k-1}$ 也均为单位下三角阵，
因此我们有

$$\det(A_k) = \frac{1}{\det(\widetilde{L}_1)\dots\det(\widetilde{L}_{k-1})} \det\left(\begin{bmatrix} A_{1,1}^{(k-1)} & * \\ & a_{k,k}^{(k-1)} \end{bmatrix}\right) = a_{k,k}^{(k-1)} \det\left(A_{1,1}^{(k-1)}\right),$$

从而 $a_{k,k}^{(k-1)} \neq 0$ 当且仅当 A_k 非奇异.
因此命题对 k 的情况也成立.
根据数学归纳法，命题得证.

1.3.3 主元的重要性

对于线性方程组 $Ax = b$ 来说，只要 A 非奇异，方程组就存在唯一解.
然而 A 非奇异不代表其前 $n-1$ 个顺序主子阵 A_k $(k = 1, \dots, n-1)$ 都是非奇异的，
因此不能保证 Gauss 消去法能够进行到底.

此外，若主元非零但是很小，则也会对算法造成不良影响.
(数值线性代数, 例 1.2.1)
考虑线性方程组

$$Ax = \begin{bmatrix} \varepsilon & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} = b,$$

其中 $\varepsilon = 1 \times 10^{-7}$.
可以解出其精确解为

$$\begin{aligned} x_1 &= \frac{1}{1-2\varepsilon} \approx 1.0000002 \\ x_2 &= \frac{1-3\varepsilon}{1-2\varepsilon} \approx 0.9999999, \end{aligned}$$

而在单精度浮点数下，通过 Gauss 消去法解得

$$\begin{aligned} x_1^{(1)} &= 1.1920929 \\ x_2^{(1)} &= 0.9999999. \end{aligned}$$

可以看出 x 在第一个元素上有较大的舍入误差.
现考虑对原来的线性方程组进行行置换，得到

$$PAx = \begin{bmatrix} 1 & 2 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} = Pb,$$

其中置换矩阵 P 为

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

用于置换系数矩阵的第一行和第二行.
此时，在单精度浮点数下，通过 Gauss 消去法解得

$$\begin{aligned} x_1^{(1)} &= 1.0000002 \\ x_2^{(1)} &= 0.9999999. \end{aligned}$$

可以看出此计算解与精确解的差距很小，
说明选择合适的主元能够减少 Gauss 消去法的舍入误差.
我们自然要问: 如何修改 Gauss 消去法使之适应所有非奇异矩阵呢?

上述示例的 MATLAB 代码如下:

```
% 系数矩阵和右端向量 (使用单精度浮点数)
epsilon = single(1e-7);
A = single([epsilon, 1;
```

```
            1,    2]);
b = single([1, 3]');

% 精确解 (解析公式)
x_exact = [1/(1-2*epsilon), (1-3*epsilon)/(1-2*epsilon)]';

% 计算解 (原始顺序)
x_cal_1 = Solve_Linear_System(A, b);

% 交换方程顺序 (相当于置换矩阵行)
A_tilde = [A(2,:); A(1,:)];
b_tilde = [b(2); b(1)];   % 注意不要转置, 保持列向量
x_cal_2 = Solve_Linear_System(A_tilde, b_tilde);

% 误差
error1 = x_cal_1 - x_exact;
err_norm1 = norm(error1);

error2 = x_cal_2 - x_exact;
err_norm2 = norm(error2);

% ----------------- 输出结果 -----------------
fprintf('精确解:\n');
fprintf('\t%.7f\n', x_exact);

fprintf('\n计算解 (原始顺序):\n');
fprintf('\t%.7f\n', x_cal_1);
fprintf('误差向量:\n');
fprintf('\t%.7f\n', error1);
fprintf('误差范数: %.7e\n', err_norm1);

fprintf('\n计算解 (行置换后):\n');
fprintf('\t%.7f\n', x_cal_2);
fprintf('误差向量:\n');
fprintf('\t%.7f\n', error2);
fprintf('误差范数: %.7e\n', err_norm2);
```

其中 `Solve_Linear_System` 函数的实现参见 **Homework 01 Problem 05**.
运行结果:

```
精确解:
    1.0000002
    0.9999999

计算解 (原始顺序):
    1.1920929
    0.9999999
误差向量:
    0.1920927
    0.0000000
误差范数: 1.9209266e-01

计算解 (行置换后):
    1.0000002
    0.9999999
误差向量:
    0.0000000
    0.0000000
误差范数: 0.0000000e+00
```

1.3.4 全选主元

在第 k 步中，若 $a_{k,k}^{(k-1)}$ 为零 (或者非零但是太小)，
则我们可以选择某个 $a_{p,q}^{(k-1)} \neq 0$ 作为主元 (为不破坏已经引入的零元素，要求 $p,q \geq k$)，
即需要交换第 k 行和第 p 行，再交换第 k 列和第 q 列.

> 任意给定 $i \neq j$
> 初等置换矩阵 $P_{i\leftrightarrow j} = [e_1, \dots, e_{i-1}, \underline{e_j}, e_{i+1}, \dots, e_{j-1}, \underline{e_i}, e_{j+1}, \dots, e_n]$
> 它是单位矩阵 I 的第 i,j 列交换得到的矩阵.
> 用 $P_{i\leftrightarrow j}$ 左乘 A 便可交换 A 的第 i,j 行，用 $P_{i\leftrightarrow j}$ 右乘 A 便可交换 A 的第 i,j 列.
>
> $P_{i\leftrightarrow j}$ 的一种更便捷的表示方法是:

$$P_{i\leftrightarrow j} = I_n - (e_ie_i^{\mathrm{T}} + e_je_j^{\mathrm{T}}) + (e_ie_j^{\mathrm{T}} + e_je_i^{\mathrm{T}}) = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 0 & \cdots & 1 & & & & \\ & & & 1 & & & & & \\ & & \vdots & & \ddots & & \vdots & & \\ & & & & & 1 & & & \\ & & 1 & \cdots & 0 & & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

假定 Gauss 消去法已经进行了 $k-1$ 步，
即已经确定了 $k-1$ 个 Gauss 变换 $L_1, \dots, L_{k-1}$ 和 $2(k-1)$ 个置换矩阵 $\begin{cases} P_1, \dots, P_{k-1} \\ Q_1, \dots, Q_{k-1} \end{cases}$ 使得

$$A^{(k-1)} = L_{k-1}P_{k-1}\cdots L_1P_1A^{(0)}Q_1\cdots Q_{k-1} = \begin{bmatrix} A_{1,1}^{(k-1)} & A_{1,2}^{(k-1)} \\ & A_{2,2}^{(k-1)} \end{bmatrix},$$

其中 $A_{1,1}^{(k-1)}$ 为 $k-1$ 阶上三角阵，记 $A^{(k-1)}$ 的 $(2,2)$ 分块为

$$A_{2,2}^{(k-1)} = \begin{bmatrix} a_{k,k}^{(k-1)} & \dots & a_{k,n}^{(k-1)} \\ \vdots & & \vdots \\ a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} \end{bmatrix}.$$

我们在 $A_{2,2}^{(k-1)}$ 中选取绝对值尽可能大的主元，即取 $(p,q) \in \arg\max_{k\leq i,j\leq n} |a_{i,j}^{(k-1)}|$.

- 若 $a_{p,q}^{(k-1)} = 0$，则说明 $A_{22}^{(k-1)}$ 为全零矩阵，消去过程终止.
- 若 $a_{p,q}^{(k-1)} \neq 0$，则我们交换 $A^{(k-1)}$ 的第 k 行和第 p 行，以及第 k 列和第 q 列.
  取 $\begin{cases} P_k = P_{k\leftrightarrow p} \\ Q_k = P_{k\leftrightarrow q} \end{cases}$，记交换后的矩阵为 $\widetilde{A}^{(k-1)} = P_kA^{(k-1)}Q_k$，记其 $(2,2)$ 分块为

  $$\widetilde{A}_{2,2}^{(k-1)} = \begin{bmatrix} \tilde{a}_{k,k}^{(k-1)} & \dots & \tilde{a}_{k,n}^{(k-1)} \\ \vdots & & \vdots \\ \tilde{a}_{n,k}^{(k-1)} & \dots & \tilde{a}_{n,n}^{(k-1)} \end{bmatrix}.$$

  然后计算 Gauss 变换

$$l_k = \frac{1}{\tilde{a}_{k,k}^{(k-1)}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tilde{a}_{k+1,k}^{(k-1)} \\ \vdots \\ \tilde{a}_{n,k}^{(k-1)} \end{bmatrix}, \quad L_k = I - l_k e_k^{\mathrm{T}} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\frac{\tilde{a}_{k+1,k}^{(k-1)}}{\tilde{a}_{k,k}^{(k-1)}} & 1 & \\ & & \vdots & & \ddots \\ & & -\frac{\tilde{a}_{n,n}^{(k-1)}}{\tilde{a}_{k,k}^{(k-1)}} & & 1 \end{bmatrix},$$

这样便有

$$A^{(k)} = L_k \widetilde{A}^{(k)} = L_k (P_k A^{(k-1)} Q_k) = \begin{bmatrix} A_{1,1}^{(k)} & A_{1,2}^{(k)} \\ & A_{2,2}^{(k)} \end{bmatrix},$$

其中 $A_{1,1}^{(k)}$ 为 k 阶上三角阵.

上述过程称为**全选主元 Gauss 消去法**.

(全选主元 Gauss 消去法, 数值线性代数, 算法 1.2.1)

$\text{function: } [P, Q, L, U] = \text{GaussianEliminationCompletePivoting}(A)$
$n = \dim(A)$
$P = Q = I_n$
$\text{for } k = 1 : n - 1$
$\text{choose } (p, q) \in \arg\max_{k \le i, j \le n} |A(i, j)| \quad (\text{确定主元位置})$
$A(k, 1 : n) \leftrightarrow A(p, 1 : n) \quad (\text{交换第 } k, p \text{ 行})$
$A(1 : n, k) \leftrightarrow A(1 : n, q) \quad (\text{交换第 } k, q \text{ 列})$
$P(k, 1 : n) \leftrightarrow P(p, 1 : n) \quad (\text{记录置换矩阵 } P_k)$
$Q(1 : n, k) \leftrightarrow Q(1 : n, q) \quad (\text{记录置换矩阵 } Q_k)$
$\text{if } A(k, k) \neq 0$
(进行 Gauss 消去)
$A(k + 1 : n, k) = A(k + 1 : n, k + 1 : n)/A(k, k)$
$A(k + 1 : n, k + 1 : n) = A(k + 1 : n, k + 1 : n) - A(k + 1 : n, k)A(k, k + 1 : n)$
else
$\text{break} \quad (\text{矩阵奇异})$
end
$L = I_n + A \odot (\text{strictly lower triangular matrix with all ones})$
$U = A \odot (\text{upper triangular matrix with all ones})$
$\text{return } [P, Q, L, U]$

其中 $\odot$ 代表 Hadamard 乘积，即逐元素乘积.
与不选主元的 Gauss 消去法一样，L 的严格下三角部分和 U 分别存储在 A 的严格下三角部分和上三角部分.
其 MATLAB 实现参见 **Homework 02 Problem 04**.

假设全主元 Gauss 消去法进行 r 步后终止，
则我们得到 r 个 Gauss 变换 $L_1, \dots, L_r$ 和 $2r$ 个初等置换矩阵 $\begin{cases} P_1, \dots, P_r \\ Q_1, \dots, Q_r \end{cases}$
使得 $U := A^{(r)} = L_r P_r \cdots L_1 P_1 A Q_1 \cdots Q_r$ 为上三角阵.

取 $\begin{cases} Q = Q_1 \cdots Q_r \\ P = P_r \cdots P_1 \\ L = P(L_r P_r \cdots L_1 P_1)^{-1} \end{cases}$ 则我们有 $PAQ = LU$.
根据 Gauss 消元法的特性，我们知道 L 是一个单位下三角阵 (对角元均为 1).
而且由于我们采用全选主元策略，故 L 的严格下三角元的绝对值都小于等于 1.
根据算法的终止条件我们可知 r 是方阵 A 的秩，因此 U 非零对角元个数为 r.

(数值线性代数, 定理 1.2.1)
设 $A \in \mathbb{R}^{n \times n}$，记 $r = \text{rank}(A)$.
则存在排列矩阵 $P, Q \in \mathbb{R}^{n \times n}$，单位下三角阵 $L \in \mathbb{C}^{n \times n}$ 和上三角阵 $U \in \mathbb{C}^{n \times n}$ 使得 $PAQ = LU$.
其中 L 的严格下三角元的模长都小于等于 1，且 U 的非零对角元个数为 r.

设 $A \in \mathbb{C}^{n\times n}$ 非奇异 (即满秩)，则线性方程组 $Ax = b$ 可以这样求解:

- 用全选主元 Gauss 消去法得到 $PAQ = LU$
- 用前代法求解 $Ly = Pb$ 得到 y
- 用回代法求解 $Uz = y$ 得到 z
- 计算 $x = Qz$

1.3.5 部分主元

虽然全主元 Gauss 消去法弥补了不选主元的 Gauss 消去法的不足，但其代价极其昂贵.
在 A 非奇异 (即满秩) 的情况下，
全选主元必须进行 $\sum_{k=1}^{n-1}(n-k+1)^2 = \sum_{i=2}^{n} i^2 = \frac{1}{3}n^3 + O(n^2)$ 次比较操作.
为尽可能减少所进行的比较，我们提出**部分主元的 Gauss 消去法**:
第 k 步只在 $A_{2,2}^{(k-1)}$ 的第 k 列上寻找绝对值最大元，
取 $p \in \arg\max_{k\le i\le n} |a_{i,k}^{(k-1)}|$，则选取的主元是 $a_{p,k}^{(k-1)}$.
这样我们只需行交换而不需要列交换，即 $\begin{cases} P_k = P_{k\leftrightarrow p} \\ Q_k = I \end{cases}$.

(部分主元 Gauss 消去法, 数值线性代数, 算法 1.2.2)

$$
\begin{aligned}
&\text{function: } [P, L, U] = \text{GaussianEliminationPartialPivoting}(A) \\
&\quad n = \dim(A) \\
&\quad P = I_n \\
&\quad \text{for } k = 1 : n-1 \\
&\qquad \text{choose } p \in \arg\max_{k\le i\le n} |A(i,k)| \quad (\text{确定主元位置}) \\
&\qquad A(k, 1:n) \leftrightarrow A(p, 1:n) \quad (\text{交换第 } k, p \text{ 行}) \\
&\qquad P(k, 1:n) \leftrightarrow P(p, 1:n) \quad (\text{记录置换矩阵 } P_k) \\
&\qquad \text{if } A(k,k) \neq 0 \\
&\qquad\quad (\text{进行 Gauss 消去}) \\
&\qquad\quad A(k+1:n, k) = A(k+1:n, k+1:n)/A(k,k) \\
&\qquad\quad A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - A(k+1:n, k)A(k, k+1:n) \\
&\qquad \text{else} \\
&\qquad\quad \text{break} \quad (\text{矩阵奇异}) \\
&\qquad \text{end} \\
&\quad L = I_n + A \odot (\text{strictly lower triangular matrix with all ones}) \\
&\quad U = A \odot (\text{upper triangular matrix with all ones}) \\
&\quad \text{return } [P, L, U]
\end{aligned}
$$

其中 $\odot$ 代表 Hadamard 乘积，即逐元素乘积.
与不选主元的 Gauss 消去法一样，L 的严格下三角部分和 U 分别存储在 A 的严格下三角部分和上三角部分.
其 MATLAB 实现参见 **Homework 02 Problem 04**.

设 $A \in \mathbb{C}^{n\times n}$ 非奇异 (即满秩)，则线性方程组 $Ax = b$ 可以这样求解:

- 用部分主元 Gauss 消去法得到 $PA = LU$
- 用前代法求解 $Ly = Pb$ 得到 y
- 用回代法求解 $Ux = y$ 得到 x

实际计算的经验和理论分析的结果表明，
部分主元 Gauss 消去法在数值稳定性方面完全可以媲美全主元 Gauss 消去法，同时具有更小的计算复杂度.
它是目前求解中小型稠密线性方程组最受欢迎的方法之一.

部分主元 Gauss 消去法可用于计算行列式.
设 $A \in \mathbb{C}^{n\times n}$ 的部分主元 Gauss 消去法的结果是 $PA = LU$，
注意到置换矩阵 P 的行列式 $\det(P) = 1$，单位下三角阵 L 的行列式 $\det(L) = 1$，
因此 $\det(A) = \det(L)\det(U)/\det(P) = \det(U)$ 很容易计算，就是 U 所有主对角元的乘积.
这个算法的计算复杂度是 $O(2n^3/3)$.

- 如果使用 Laplace 展开计算行列式 $\det(A)$，
  则根据计算量的递归关系

$$\begin{cases} T(n) = (n-1)T(n-1) + O(n) \\ T(1) = O(1) \end{cases}$$

可知 $T(n) = O(n!)$.
因此使用 Laplace 展开计算行列式 $\det(A)$ 的计算复杂度是 $O(n!)$.

若使用 Cramer 法则求解线性方程组 $Ax = b$，则我们需要计算 $n+1$ 个 n 阶行列式.
若使用部分主元 Gauss 消去法来计算行列式，则总计算复杂度是 $O(n \cdot 2n^3/3)$.
若使用 Laplace 展开计算行列式，则总计算复杂度是 $O(n \cdot n!)$.
当然，直接使用部分主元 Gauss 消去法结合回代法和前代法求解线性方程组的总计算复杂度是 $O(2n^3/3)$.

1.3.6 分块三角分解

基于计算机的存储层次结构，
在编制程序时应当尽量减少主存与磁盘、寄存器与主存之间的数据传输次数.

假定完成某计算任务的运算量为 f，数据传输次数为 m，
我们记平均每次数据传输可做的运算量为 $q = \frac{f}{m}$
记 t_{arith} 为做一次运算所需的时间，t_{mem} 为一次数据传输所需的时间
这样完成该计算任务所需的时间为 $f \cdot t_{\text{arith}} + m \cdot t_{\text{mem}} = f \cdot t_{\text{arith}}(1 + \frac{1}{q} \cdot \frac{t_{\text{mem}}}{t_{\text{arith}}})$
由此可见，q 越大，执行该任务的时间越少，效率越高.

数值线性代数的算法一般主要是由一些向量运算、矩阵-向量运算和矩阵-矩阵运算组成的.
我们已将这三种类型的一些最常用的基本运算编制成了**数值线性代数基础子程序**，
称为 BLAS (Basic Linear Algebra Subroutines):

- $\mathrm{BLAS1}$: 常用的向量运算，例如 $\begin{cases} y \leftarrow y + \alpha x \\ \alpha \leftarrow x^{\mathrm{T}} y \\ x \leftrightarrow y \end{cases}$
- $\mathrm{BLAS2}$: 常用的矩阵-向量运算，例如 $\begin{cases} y \leftarrow y + Ax \\ A \leftarrow A + xy^{\mathrm{T}} \\ x \leftarrow U^{-1}x \end{cases}$
- $\mathrm{BLAS3}$: 常用的矩阵-矩阵运算，例如 $\begin{cases} Y \leftarrow Y + AX \\ A \leftarrow A + XY^{\mathrm{T}} \\ X \leftarrow U^{-1}X \end{cases}$

典型运算	**运算量 f**	**数据传输次数 t**	**平均每次数据传输可做的运算量 $q = \frac{f}{t}$**
$y \leftarrow y + \alpha x$	$2n$	$3n+1$	$\frac{2}{3}$
$y \leftarrow y + Ax$	$2n^2 + n$	$n^2 + 3n$	2
$Y \leftarrow Y + AX$	$2n^3 + n^2$	$4n^2$	$\frac{1}{2}n$

显然矩阵-矩阵运算的效率是最高的，因此编程时应尽可能多地使用矩阵-矩阵运算 $\mathrm{BLAS3}$.
(以前是 GotoBLAS 为硬件开发 BLAS 运算，但现在是中国的 OpenBLAS 团队在维护)

下面我们简要介绍如何在计算一个矩阵的三角分解时尽可能多地使用 $\mathrm{BLAS3}$
设 $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ 我们令:

$$LU = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & \widetilde{A}_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = A$$

比较两侧的对应分块可得:

$$\begin{cases} A_{11} = L_{11}U_{11} \\ A_{12} = L_{11}U_{12} \\ A_{21} = L_{21}U_{11} \\ A_{22} = L_{21}U_{12} + \widetilde{A}_{22} \end{cases}$$

于是我们可按照以下步骤计算各个分块矩阵:

- 使用 Gauss 消去法计算分块 A_{11} 的三角分解 $A_{11}=L_{11}U_{11}$，得到 L_{11} 和 U_{11}
  (实际计算时应当使用部分主元的 Gauss 消去法)
- 使用前代法和回代法解方程组 $\begin{cases}L_{11}U_{12}=A_{12}\\L_{21}U_{11}=A_{21}\end{cases}$ 得到 U_{12} 和 L_{21}
- 计算 $\widetilde{A}_{22}=A_{22}-L_{21}U_{12}$

最后两步均为 $\mathrm{BLAS3}$ 运算，而且我们可以对 $\widetilde{A}_{22}$ 重复上述过程，
得到一个尽可能多地使用 $\mathrm{BLAS3}$ 运算的三角分解算法.
假设分块的平均阶数为 n_b，则计算复杂度约为 $\frac{n}{n_b}\cdot O(n_b^3)=O(n\cdot n_b^2)$

设 $A=\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}$ 我们令:

$$LU=\begin{bmatrix}L_{11}&\\L_{21}&I\end{bmatrix}\begin{bmatrix}U_{11}&U_{12}\\&\widetilde{A}_{22}\end{bmatrix}=\begin{bmatrix}P_{11}&\\&I\end{bmatrix}\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}=PA$$

> 邵老师说这里的做法是不正确的，部分主元应对整个矩阵的列进行，而不仅仅是 A_{11} 分块的列
> 因此追求精度 (部分主元) 和追求高性能计算 (分块算法) 是冲突的.
> But I choose to do it anyway.

比较两侧的对应分块可得:

$$\begin{cases}P_{11}A_{11}=L_{11}U_{11}\\P_{11}A_{12}=L_{11}U_{12}\\A_{21}=L_{21}U_{11}\\A_{22}=L_{21}U_{12}+\widetilde{A}_{22}\end{cases}$$

于是我们可按照以下步骤计算各个分块矩阵:

- ① 使用部分主元 Gauss 消去法计算分块 A_{11} 的三角分解 $P_{11}A_{11}=L_{11}U_{11}$，得到 P_{11},L_{11} 和 U_{11}
- ② 使用前代法和回代法解方程组 $\begin{cases}L_{11}U_{12}=P_{11}A_{12}\\L_{21}U_{11}=A_{21}\end{cases}$ 得到 U_{12} 和 L_{21}
  为规避 $P_{11}A_{12}$ 的矩阵乘法，可以不显式地计算 $P_{11}A_{12}$，而是融合到 ① 中的行交换中
  也就是说，将 ① 中的行交换应用到 A_{11},A_{12} 所在的整个区域上.
- ③ 计算 $\widetilde{A}_{22}=A_{22}-L_{21}U_{12}$

最后两步均为 $\mathrm{BLAS3}$ 运算，而且我们可以对 $\widetilde{A}_{22}$ 重复上述过程，
得到一个尽可能多地使用 $\mathrm{BLAS3}$ 运算的三角分解算法.

设 $A=\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}$ 我们令:

$$LU=\begin{bmatrix}L_{11}&\\L_{21}&I\end{bmatrix}\begin{bmatrix}U_{11}&U_{12}\\&\widetilde{A}_{22}\end{bmatrix}=P\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}=PA$$

比较两侧的对应分块可得:

$$\begin{cases}P\begin{bmatrix}A_{11}\\A_{21}\end{bmatrix}=\begin{bmatrix}L_{11}U_{11}\\L_{21}U_{11}\end{bmatrix}\\P\begin{bmatrix}A_{12}\\A_{22}\end{bmatrix}=\begin{bmatrix}\hat{A}_{12}\\\hat{A}_{22}\end{bmatrix}=\begin{bmatrix}L_{11}U_{12}\\L_{21}U_{12}+\widetilde{A}_{22}\end{bmatrix}\end{cases}$$

于是我们可按照以下步骤计算各个分块矩阵:

- ① 使用部分主元 Gauss 消去法计算分块 $\begin{bmatrix}A_{11}\\A_{21}\end{bmatrix}$ 的长方三角分解 $P\begin{bmatrix}A_{11}\\A_{21}\end{bmatrix}=\begin{bmatrix}L_{11}&\\L_{21}&I\end{bmatrix}\begin{bmatrix}U_{11}\\0\end{bmatrix}$，得到 P,L_{11},L_{21} 和 U_{11}
- ② 使用前代法解方程组 $L_{11}U_{12}=\hat{A}_{12}$ 得到 U_{12}
  为规避 $P\begin{bmatrix}A_{12}\\A_{22}\end{bmatrix}$ 的矩阵乘法，可以不显式地计算它，而是融合到 ① 中的行交换中，得到 $\begin{bmatrix}\hat{A}_{12}\\\hat{A}_{22}\end{bmatrix}$
- ③ 计算 $\widetilde{A}_{22}=\hat{A}_{22}-L_{21}U_{12}$

最后两步均为 BLAS3 运算，而且我们可以对 $\widetilde{A}_{22}$ 重复上述过程，
得到一个尽可能多地使用 BLAS3 运算的三角分解算法.

更极端地，我们有时会牺牲计算量 (提升 $2\sim3$ 倍)，减少算法的计算时间.

1.3.7 舍入误差分析

(数值线性代数, 引理 2.4.1)
若 $A=[a_{ij}]\in\mathbb{R}^{n\times n}$ 存在三角分解 (即前 $n-1$ 个顺序主子阵都正定)
则用不选主元的 Gauss 消去法得到的单位下三角阵 $\widetilde{L}$ 和上三角阵 $\widetilde{U}$ 满足 $\begin{cases}\widetilde{L}\widetilde{U}=A+\Delta_A\\|\Delta_A|\le\gamma_n|\widetilde{L}||\widetilde{U}|\end{cases}$
其中 γ_n 代表 n 层浮点运算的累积相对误差的绝对值.

> **(Gauss 消去法, 数值线性代数, 算法 1.1.3)**
>
> for $k=1:n-1$
> $\quad A(k+1:n,k)\leftarrow A(k+1:n,k)/A(k,k)$
> $\quad A(k+1:n,k+1:n)\leftarrow A(k+1:n,k+1:n)-A(k+1:n,k)A(k,k+1:n)$
> end

证明: 设 $\widetilde{L}=[\tilde{l}_{ij}]$ 和 $\widetilde{U}=[\tilde{u}_{ij}]$

一方面，我们有 $\begin{cases}a_{ij}^{(0)}=a_{ij}\\a_{ij}^{(k)}=\mathrm{fl}(a_{ij}^{(k-1)}-\mathrm{fl}(\tilde{l}_{ik}\tilde{u}_{kj}))&(k=1,\dots,i-2)\\\tilde{u}_{ij}=a_{ij}^{(i-1)}&(i\le j)\end{cases}$

于是有 $a_{ij}^{(k)}=[a_{ij}^{(k-1)}-\tilde{l}_{ik}\tilde{u}_{kj}(1+r_k)](1+\delta_k)\ \ (|r_k|,|\delta_k|\le\mathrm{eps},k=1,\dots,i-1)$

对于任意给定的 $1\le i\le j\le n$，我们都有:

$$\begin{aligned}\tilde{u}_{ij}&=a_{ij}^{(i-1)}\\&=[a_{ij}^{(i-2)}-\tilde{l}_{i,i-1}\tilde{u}_{i-1,j}(1+r_{i-1})](1+\delta_{i-1})\\&=a_{ij}^{(i-2)}(1+\delta_{i-1})-\tilde{l}_{i,i-1}\tilde{u}_{i-1,j}(1+r_{i-1})(1+\delta_{i-1})\\&=\cdots\\&=a_{ij}^{(0)}\prod_{p=1}^{i-1}(1+\delta_p)-\sum_{k=1}^{i-1}[\tilde{l}_{ik}\tilde{u}_{kj}(1+r_k)\prod_{k=1}^{i-1}(1+\delta_p)]\\&=a_{ij}(1+\varepsilon_i)-\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}(1+\varepsilon_k)\end{aligned}$$

其中 $\begin{cases}1+\varepsilon_i=\prod_{p=1}^{i-1}(1+\delta_p)\\1+\varepsilon_k=(1+r_k)\prod_{p=k}^{i-1}(1+\delta_p)\quad(k=1,\dots,i-1)\end{cases}$ (我们有 $|\varepsilon_k|:=\begin{cases}\gamma_{i+1-k}&k=1,\dots,k-1\\\gamma_{i-1}&k=i\end{cases}$)

我们可以反解出:

$$\begin{aligned}a_{ij}&=\frac{1}{1+\varepsilon_i}[\tilde{u}_{ij}+\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}(1+\varepsilon_k)]\\&=\frac{\tilde{u}_{ij}}{1+\varepsilon_i}+\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{1+\varepsilon_k}{1+\varepsilon_i}\quad(\text{note that }\tilde{l}_{ii}=1)\\&=\frac{\tilde{l}_{ii}\tilde{u}_{ij}}{1+\varepsilon_i}+\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{1+\varepsilon_k}{1+\varepsilon_i}\\&=\sum_{k=1}^{i}\tilde{l}_{ik}\tilde{u}_{kj}-\tilde{l}_{ii}\tilde{u}_{ij}\frac{\varepsilon_i}{1+\varepsilon_i}-\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{\varepsilon_i-\varepsilon_k}{1+\varepsilon_i}\end{aligned}$$

于是我们有:

$$
\begin{aligned}
|a_{ij}-\sum_{k=1}^{i}\tilde{l}_{ik}\tilde{u}_{kj}| &= |\tilde{l}_{ii}\tilde{u}_{ij}\frac{\varepsilon_i}{1+\varepsilon_i}+\sum_{k=1}^{i-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{\varepsilon_i-\varepsilon_k}{1+\varepsilon_i}| \\
&\le |\tilde{l}_{ii}||\tilde{u}_{ij}|\frac{|\varepsilon_i|}{|1+\varepsilon_i|}+\sum_{k=1}^{i-1}|\tilde{l}_{ik}||\tilde{u}_{kj}|\frac{|\varepsilon_i|+|\varepsilon_k|}{|1+\varepsilon_i|} \quad (|1+\varepsilon_i|\approx 1 \text{ and } |\varepsilon_k| := \begin{cases}\gamma_{i+1-k} & k=1,\dots,i-1\\ \gamma_{i-1} & k=i\end{cases}) \\
&\approx |\tilde{l}_{ii}||\tilde{u}_{ij}|\gamma_{i-1}+\sum_{k=1}^{i-1}|\tilde{l}_{ik}||\tilde{u}_{kj}|(\gamma_{i-1}+\gamma_{i+1-k}) \quad (\text{note that the max index of } \gamma \text{ is } i\le n) \\
&\le \gamma_n\sum_{k=1}^{i}|\tilde{l}_{ik}||\tilde{u}_{kj}|
\end{aligned}
$$

另一方面，我们有 $\begin{cases}a_{ij}^{(0)}=a_{ij}\\ a_{ij}^{(k)}=\mathrm{fl}(a_{ij}^{(k-1)}-\mathrm{fl}(\tilde{l}_{ik}\tilde{u}_{kj})) & (k=1,\dots,j-1)\\ \tilde{l}_{ij}=\mathrm{fl}(a_{ij}^{(j-1)}/\tilde{u}_{jj}) & (i>j)\end{cases}$

于是有 $\begin{cases}a_{ij}^{(k)}=[a_{ij}^{(k-1)}-\tilde{l}_{ik}\tilde{u}_{kj}(1+r_k)](1+\delta_k) & (|r_k|,|\delta_k|\le \mathrm{eps}, k=1,\dots,j-1)\\ \tilde{l}_{ij}=(a_{ij}^{(j-1)}/\tilde{u}_{jj})(1+\zeta) & (|\zeta|\le\mathrm{eps})\end{cases}$

对于任意给定的 $1\le j<i\le n$，我们都有:

$$
\begin{aligned}
\tilde{l}_{ij} &= (a_{ij}^{(j-1)}/\tilde{u}_{jj})(1+\zeta) \\
&= \frac{(1+\zeta)}{\tilde{u}_{jj}}[a_{ij}^{(j-2)}-\tilde{l}_{i,j-1}\tilde{u}_{j-1,j}(1+r_{j-1})](1+\delta_{j-1}) \\
&= \frac{(1+\zeta)}{\tilde{u}_{jj}}[a_{ij}^{(j-2)}(1+\delta_{j-1})-\tilde{l}_{i,j-1}\tilde{u}_{j-1,j}(1+r_{j-1})(1+\delta_{j-1})] \\
&= \cdots \\
&= \frac{1}{\tilde{u}_{jj}}\{a_{ij}^{(0)}(1+\zeta)\prod_{p=1}^{j-1}(1+\delta_p)-\sum_{k=1}^{j-1}[\tilde{l}_{ik}\tilde{u}_{kj}(1+\zeta)(1+r_k)\prod_{p=k}^{j-1}(1+\delta_p)]\} \\
&= \frac{1}{\tilde{u}_{jj}}\{a_{ij}(1+\varepsilon_j)-\sum_{k=1}^{j-1}\tilde{l}_{ik}\tilde{u}_{kj}(1+\varepsilon_k)\}
\end{aligned}
$$

其中 $\begin{cases}1+\varepsilon_j=(1+\zeta)\prod_{p=1}^{j-1}(1+\delta_p)\\ 1+\varepsilon_k=(1+\zeta)(1+r_k)\prod_{p=k}^{j-1}(1+\delta_p) & (k=1,\dots,j-1)\end{cases}$ (我们有 $|\varepsilon_k| := \begin{cases}\gamma_{j+2-k} & k=1,\dots,j-1\\ \gamma_j & k=j\end{cases}$)

我们可以反解出:

$$
\begin{aligned}
a_{ij} &= \frac{1}{1+\varepsilon_j}[\tilde{l}_{ij}\tilde{u}_{jj}+\sum_{k=1}^{j-1}\tilde{l}_{ik}\tilde{u}_{kj}(1+\varepsilon_k)] \\
&= \tilde{l}_{ij}\tilde{u}_{jj}\frac{1}{1+\varepsilon_j}+\sum_{k=1}^{j-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{1+\varepsilon_k}{1+\varepsilon_j} \\
&= \sum_{k=1}^{j}\tilde{l}_{ik}\tilde{u}_{kj}-\tilde{l}_{ij}\tilde{u}_{jj}\frac{\varepsilon_j}{1+\varepsilon_j}-\sum_{k=1}^{j-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{\varepsilon_j-\varepsilon_k}{1+\varepsilon_j}
\end{aligned}
$$

于是我们有:

$$
\begin{aligned}
|a_{ij}-\sum_{k=1}^{j}\tilde{l}_{ik}\tilde{u}_{kj}| &= |\tilde{l}_{ij}\tilde{u}_{jj}\frac{\varepsilon_j}{1+\varepsilon_j}+\sum_{k=1}^{j-1}\tilde{l}_{ik}\tilde{u}_{kj}\frac{\varepsilon_j-\varepsilon_k}{1+\varepsilon_j}| \\
&\le |\tilde{l}_{ij}||\tilde{u}_{jj}|\frac{|\varepsilon_j|}{|1+\varepsilon_j|}+\sum_{k=1}^{j-1}|\tilde{l}_{ik}||\tilde{u}_{kj}|\frac{|\varepsilon_j|+|\varepsilon_k|}{|1+\varepsilon_j|} \quad (|1+\varepsilon_j|\approx 1 \text{ and } |\varepsilon_k| := \begin{cases}\gamma_{j+2-k} & k=1,\dots,j-1\\ \gamma_j & k=j\end{cases}) \\
&\approx |\tilde{l}_{ij}||\tilde{u}_{jj}|\gamma_j+\sum_{k=1}^{j-1}|\tilde{l}_{ik}||\tilde{u}_{kj}|(\gamma_j+\gamma_{j+2-k}) \quad (\text{note that the max index of } \gamma \text{ is } j+1\le n) \\
&\le \gamma_n\sum_{k=1}^{j}|\tilde{l}_{ik}||\tilde{u}_{kj}|
\end{aligned}
$$

综上所述，命题得证.

注意到交换浮点数矩阵的行或列并不会引入舍入误差，因此我们有:
(数值线性代数, 推论 2.4.1)
若 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 非奇异 (即满秩)，
则用部分主元的 Gauss 消去法得到的单位下三角阵 $\widetilde{L}$、上三角阵 $\widetilde{U}$ 以及排列矩阵 $\widetilde{P}$ 满足: $\begin{cases}\widetilde{L}\widetilde{U} = \widetilde{P}A + \Delta_A \\ |\Delta_A| \le \gamma_n |\widetilde{L}||\widetilde{U}|\end{cases}$
其中 γ_n 代表 n 层浮点运算的累积相对误差的绝对值.

当对 A 完成三角分解后，求解线性方程组 $Ax = b$ 的问题就归结为:

- 用前代法求解 $\widetilde{L}y = \widetilde{P}b$ 得到 y
- 用回代法求解 $\widetilde{U}x = y$ 得到 x

当对 A 完成三角分解 $\begin{cases}\widetilde{L}\widetilde{U} = \widetilde{P}A + \Delta_{\text{Gauss}} \\ |\Delta_{\text{Gauss}}| \le \gamma_n |\widetilde{L}||\widetilde{U}|\end{cases}$ 后，求解线性方程组 $Ax = b$ 的问题就归结为:

- 用前代法求解 $\widetilde{L}y = \widetilde{P}b$ 得到 y
- 用回代法求解 $\widetilde{U}x = y$ 得到 x

根据**数值线性代数 引理** $2.4.2$ 可知最后的解 $\tilde{x}$ 满足 $\begin{cases}(\widetilde{L} + \Delta_L)(\widetilde{U} + \Delta_U)\tilde{x} = \widetilde{P}b \\ |\Delta_L| \le \gamma_n |\widetilde{L}| \\ |\Delta_U| \le \gamma_n |\widetilde{U}|\end{cases}$

联立 $\begin{cases}\widetilde{L}\widetilde{U} = \widetilde{P}A + \Delta_{\text{Gauss}} \\ (\widetilde{L} + \Delta_L)(\widetilde{U} + \Delta_U)\tilde{x} = (\widetilde{L}\widetilde{U} + \widetilde{L}\Delta_U + \Delta_L\widetilde{U} + \Delta_L\Delta_U)\tilde{x} = \widetilde{P}b\end{cases}$ 就得到:

$$
\begin{gathered}
(\widetilde{P}A + \Delta_{\text{Gauss}} + \widetilde{L}\Delta_U + \Delta_L\widetilde{U} + \Delta_L\Delta_U)\tilde{x} = \widetilde{P}b \\
\Leftrightarrow \\
(A + \widetilde{P}^{\mathrm{T}}(\Delta_{\text{Gauss}} + \widetilde{L}\Delta_U + \Delta_L\widetilde{U} + \Delta_L\Delta_U))\tilde{x} = b \\
\Leftrightarrow \\
(A + \Delta_A)\tilde{x} = b \text{ where } \Delta_A = \widetilde{P}^{\mathrm{T}}(\Delta_{\text{Gauss}} + \widetilde{L}\Delta_U + \Delta_L\widetilde{U} + \Delta_L\Delta_U)
\end{gathered}
$$

关于 Δ_A 我们有:

$$
\begin{aligned}
|\Delta_A| &= |\widetilde{P}^{\mathrm{T}}(\Delta_{\text{Gauss}} + \widetilde{L}\Delta_U + \Delta_L\widetilde{U} + \Delta_L\Delta_U)| \\
&\le \widetilde{P}^{\mathrm{T}}(|\Delta_{\text{Gauss}}| + |\widetilde{L}||\Delta_U| + |\Delta_L||\widetilde{U}| + |\Delta_L||\Delta_U|) \\
&\le \widetilde{P}^{\mathrm{T}}(\gamma_n|\widetilde{L}||\widetilde{U}| + |\widetilde{L}| \cdot \gamma_n|\widetilde{U}| + |\widetilde{U}| \cdot \gamma_n|\widetilde{L}| + \gamma_n|\widetilde{L}| \cdot \gamma_n|\widetilde{U}|) \\
&= (3\gamma_n + \gamma_n^2)\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}| \\
&\le (3\gamma_n + 3\gamma_n^2 + \gamma_n^3)\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}| \\
&= [(1+\gamma_n)^3 - 1]\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}| \\
&= [(1+\gamma_{3n}) - 1]\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}| \\
&= \gamma_{3n}\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}|
\end{aligned}
$$

注意到 $\widetilde{L}$ 是单位下三角阵，且部分主元策略保证了 $\widetilde{L}$ 的严格下三角元的绝对值都小于等于 1，
因此我们有 $\|\widetilde{L}\|_\infty \le n$ (部分主元的算法成功将 $\|\widetilde{L}\|_\infty$ 按下去了)
但是 $\|\widetilde{U}\|_\infty$ 最坏的情况是 $O(2^n)$，例如以下的情况:

$$
A = \begin{bmatrix}
1 & & & & 1 \\
& 1 & & & 2 \\
& & \ddots & & \vdots \\
& & & 1 & 2^{n-2} \\
& & & & 2^{n-1}
\end{bmatrix}
$$

这个上估计太大了 (尽管它能够取到)
(值得一提的是，全选主元策略的数值稳定性是很强的，Wilkinson 给出了一个误差上界，但通常取不到)

为给出 $\|\widetilde{U}\|_\infty$ 的更紧的上估计，我们定义**部分主元 Gauss 消去法的增长因子** $\rho = \frac{\max\limits_{i,j}|\tilde{u}_{ij}|}{\|A\|_\infty}$
(值得注意的是，增长因子的定义是不统一的，有些教材上的分子是迭代过程中 $A^{(k)}$ 元素的模最大值，即 $\max\limits_{i,j,k}|a_{ij}^{(k)}|$)
我们有 $\|\widetilde{U}\|_\infty \leq n\max\limits_{i,j}|\tilde{u}_{ij}| \leq n\rho\|A\|_\infty$
因此我们有:

$$\begin{aligned}\|\Delta_A\|_\infty &\leq \|\gamma_{3n}\widetilde{P}^{\mathrm{T}}|\widetilde{L}||\widetilde{U}|\|_\infty \\ &\leq \gamma_{3n}\|\widetilde{P}^{\mathrm{T}}\|_\infty\|\widetilde{L}\|_\infty\|\widetilde{U}\|_\infty \\ &\leq \gamma_{3n}\cdot 1\cdot n\cdot n\rho\|A\|_\infty \\ &= \gamma_{3n}n^2\rho\|A\|_\infty\end{aligned}$$

于是我们有 $\frac{\|\Delta_A\|_\infty}{\|A\|_\infty} \leq \gamma_{3n}n^2\rho$，即得到一个重要定理:
(数值线性代数, 定理 $2.4.1$)
若 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 非奇异 (即满秩)，
则用部分主元的 Gauss 消去法解线性方程组 $Ax = b$ 得到的计算解 $\tilde{x}$ 满足 $\begin{cases}(A+\Delta A)\tilde{x} = b \\ \frac{\|\Delta_A\|_\infty}{\|A\|_\infty} \leq n^2\rho\gamma_{3n}\end{cases}$
其中 γ_{3n} 代表 $3n$ 层浮点运算的累积相对误差的绝对值.

- 这个定理表明:
  部分主元的 Gauss 消去法得到的计算解相当于系数矩阵作某些扰动而得到的扰动方程组的精确解.
  一般来说，Δ_A 的元素 (舍入误差) 比起 A 的元素的初始误差 (测量误差、模型误差等) 来是很小的.
  从这个意义上来说，部分主元 Gauss 消去法是数值稳定的.
- 理论上可以证明部分主元 Gauss 消去法的增长因子 $\rho \leq 2^{n-1}$，且上界可以达到.
  但在实际问题中，ρ 通常很小 (满足 $\rho \leq n$)
  此外，上界 $n^2\rho\gamma_{3n}$ 一般比 $\frac{\|\Delta_A\|_\infty}{\|A\|_\infty}$ 大得多.
  在实际问题中，通常都有 $\frac{\|\Delta_A\|_\infty}{\|A\|_\infty} \approx \mathrm{eps}$ 成立.

1.4 Cholesky 分解

Cholesky 分解法是求解正定线性方程组最常用的方法之一.

1.4.1 存在唯一性

(Cholesky 分解的存在唯一性, 数值线性代数, 定理 $1.3.1$)
若方阵 $A \in \mathbb{C}^{n\times n}$ Hermite 正定，
则存在唯一的对角元均为正实数的下三角阵 $L \in \mathbb{C}^{n\times n}$ 使得 $A = LL^{\mathrm{H}}$.

- 我们称 $A = LL^{\mathrm{H}}$ 为 A 的 Cholesky 分解，而 L 称为 A 的 Cholesky 因子.

存在性证明:
由于 A 正定，故其所有的顺序主子阵都正定 (自然非奇异).
根据**数值线性代数 定理** $1.1.2$ 可知，
存在唯一的一对单位下三角阵 $\widetilde{L} \in \mathbb{C}^{n\times n}$ 和上三角阵 $U \in \mathbb{C}^{n\times n}$ 使得 $A = \widetilde{L}U$.

令 $\begin{cases}D = \mathrm{diag}(u_{1,1},\dots,u_{n,n}) \\ \widetilde{U} = D^{-1}U\end{cases}$ (显然 $\widetilde{U}$ 是单位上三角阵)
则有 $\widetilde{U}^{\mathrm{H}}D\widetilde{L}^{\mathrm{H}} = A^{\mathrm{H}} = A = \widetilde{L}D\widetilde{U}$.
等式两端同时左乘 $D^{-1}\widetilde{U}^{-\mathrm{H}}$ 并右乘 $\widetilde{U}^{-1}$，即有 $\widetilde{L}^{\mathrm{H}}\widetilde{U}^{-1} = D^{-1}\widetilde{U}^{-\mathrm{H}}\widetilde{L}D$.

注意到新等式的左端是单位上三角阵，右端是下三角阵，因此二者都是单位矩阵，
即有 $\widetilde{L}^{\mathrm{H}}\widetilde{U}^{-1} = D^{-1}\widetilde{U}^{-\mathrm{H}}\widetilde{L}D = I$.
因此 $\widetilde{L}^{\mathrm{H}} = \widetilde{U}$,
从而 $A = \widetilde{L}U = \widetilde{L}D\widetilde{U} = \widetilde{L}D\widetilde{L}^{\mathrm{H}}$.

根据 A 的正定性可知 D 的对角元都为正数.

> 如若不然，不妨设 $u_{1,1} \leq 0$,
> 则对于使得 $\widetilde{L}^{\mathrm{H}}x = e_1$ 的 x 都有 $x^{\mathrm{H}}Ax = x^{\mathrm{H}}\widetilde{L}D\widetilde{L}^{\mathrm{H}}x = (\widetilde{L}^{\mathrm{T}}x)^{\mathrm{T}}D(\widetilde{L}^{\mathrm{T}}x) = e_1^{\mathrm{T}}De_1 = u_{1,1} \leq 0$ 成立，
> 与 A 的正定性矛盾.

因此我们可定义 $D^{1/2} := \mathrm{diag}(\sqrt{u_{1,1}}, \dots, \sqrt{u_{n,n}})$.
并令 $L = \widetilde{L}D^{1/2}$，其对角元 $l_{i,i} = \sqrt{u_{i,i}} > 0 \ (i = 1, \dots, n)$，
且有 $A = \widetilde{L}D\widetilde{L}^{\mathrm{H}} = (\widetilde{L}D^{\frac{1}{2}})(\widetilde{L}D^{\frac{1}{2}})^{\mathrm{H}} = LL^{\mathrm{H}}$ 成立.

唯一性证明:
假设存在两个对角元均为正实数的下三角阵 L_1, L_2 使得正定阵 $A = L_1L_1^{\mathrm{H}} = L_2L_2^{\mathrm{H}}$，
则我们有 $L_2^{-1}L_1 = L_2^{\mathrm{H}}L_1^{-\mathrm{H}}$.
注意到等式左端是两个下三角阵的乘积，因此仍是下三角阵.
而等式右端是两个上三角阵的乘积，因此仍是上三角阵.
对比可知矩阵乘积一定是一个对角阵，因此只考虑对角元即可:

$$\begin{aligned} \frac{L_1(i,i)}{L_2(i,i)} &= \frac{L_2(i,i)}{L_1(i,i)} \quad (i = 1, \dots, n) \\ &\Updownarrow \\ L_1(i,i) &= L_2(i,i) \quad (i = 1, \dots, n) \end{aligned}$$

这表明 L_1 和 L_2 的对角元完全相同，且 $L_2^{-1}L_1$ 的对角元均为 1.
注意到:

$$L_2^{-1}L_1L_1^{\mathrm{H}}L_2^{-\mathrm{H}} = L_2^{-1}L_1(L_2^{-1}L_1)^{\mathrm{H}} = I_n$$

故 $L_2^{-1}L_1$ 为酉矩阵.
考虑到 $L_2^{-1}L_1$ 同时还是单位下三角阵，因此其非对角元均为零，即 $L_2^{-1}L_1 = I_n$.
(事实上，若酉矩阵 U 是下三角的，则它必为对角矩阵，且所有对角元的模长都为 1)
这说明 $L_1 = L_2$.

若方阵 $A \in \mathbb{C}^{n\times n}$ Hermite 正定，则我们可以这样求解线性方程组 $Ax = b$:

- 计算 A 的 Cholesky 分解 $A = LL^{\mathrm{H}}$
- 前代法求解 $Ly = b$ 得到 y
- 回代法求解 $L^{\mathrm{H}}x = y$ 得到 x

1.4.2 平方根法

关于 Cholesky 分解，更简单实用的方法是逐元素比较 $A = LL^{\mathrm{T}}$ 来计算 L.
设 $L = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}$
比较 $A = LL^{\mathrm{T}}$ 两边对应元素，得到 $a_{ij} = \sum\limits_{p=1}^{\min(i,j)} l_{ip}l_{jp} \ (1 \le i, j \le n)$

- 首先由 $a_{11} = l_{11}^2$ 得到 $l_{11} = \sqrt{a_{11}}$
  再由 $a_{i1} = l_{11}l_{i1} \ (1 \le i \le n)$ 得到 $l_{i1} = \frac{1}{l_{11}}a_{i1} \ (1 \le i \le n)$
  这样便得到矩阵 L 的第 1 列元素.
- 假设已经计算出 L 的前 $k-1$ 列元素.
  由 $a_{kk} = \sum\limits_{p=1}^{k} l_{kp}^2$ 得到 $l_{kk} = (a_{kk} - \sum\limits_{p=1}^{k-1} l_{kp}^2)^{\frac{1}{2}}$
  再由 $a_{ik} = \sum\limits_{p=1}^{k} l_{ip}l_{kp} = \sum\limits_{p=1}^{k-1} l_{ip}l_{kp} + l_{ik}l_{kk} \ \ (i = k+1, \dots, n)$
  得到 $l_{ik} = \frac{1}{l_{kk}}(a_{ik} - \sum\limits_{p=1}^{k-1} l_{ip}l_{kp}) \ \ (i = k+1, \dots, n)$
  这样便得到矩阵 L 的第 k 列元素.

上述次序可以调整为按行计算.
由于 A 的元素 a_{ij} 被用来计算 l_{ij} 后就不再使用，故我们可将 L 的元素存储在 A 的对应位置上.
(平方根法, 数值线性代数, 算法 $1.3.1$)

Given symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$

function: $[L] = \text{Cholesky}(A)$

  for $k = 1 : n$

    $A(k,k) = \sqrt{A(k,k)}$

    $A(k+1:n,k) = A(k+1:n,k)/A(k,k)$

    for $j = k+1 : n$

      $A(j:n,j) = A(j:n,j) - A(j:n,k)A(j,k)$

    end

  end

  $L = A \odot$ (lower triangular matrix with all ones) ($\odot$ stands for Hadamand product)

end

总浮点运算数为:

$$\begin{aligned}\sum_{k=1}^{n}\left[1+(n-k-1)+\sum_{j=k+1}^{n}2(n-j+1)\right] &= \sum_{k=1}^{n}[1+(n-k-1)+2\cdot\frac{1}{2}(n-k)(n-k-1)] \\ &= \sum_{i=1}^{n}[1+i+(i+1)i] \\ &= \sum_{i=1}^{n}(1+2i+i^2) \\ &= n+2\cdot\frac{1}{2}n(n+1)+\frac{1}{6}n(n+1)(2n+1) \\ &= O(\frac{1}{3}n^3)\end{aligned}$$

仅是不选主元 Gauss 消去法运算量的一半.

此外，平方根法计算 Cholesky 分解的计算过程是稳定的.
因为 $a_{ii} = \sum_{p=1}^{i} l_{ip}^2$，所以 $|l_{ij}| \le \sqrt{a_{ii}}\ \ (j = 1, \dots, i-1)$
这表明 Cholesky 分解中的量 l_{ij} 能够得以控制，因此其计算过程是稳定的.

上述算法拓展到复数域上的结果参见 Homework 4 的 Problem 3 (1)

那么非奇异线性方程组 $Ax = b$ 能否化为 $A^{\mathrm{H}}Ax = A^{\mathrm{H}}b$ 求解呢?
一般来说是不提倡这样做的，原因如下:

- ① 计算 $A^{\mathrm{H}}A$ 的代价是 $O(n^3)$ (只用计算下三角部分的元素)，比较贵.
- ② 条件数 $\kappa_2(A^{\mathrm{H}}A) = (\kappa_2(A))^2$，只有在某些特殊情况下可以这样做 (例如增长因子失控的情况).

我们来估计按 Gauss 消去法计算的 Cholesky 分解的增长因子 $\frac{\|L\|_\infty\|L^{\mathrm{H}}\|_\infty}{\|A\|_\infty}$
根据范数的等价性，我们可转而考虑 $\frac{\|L\|_2\|L^{\mathrm{H}}\|_2}{\|A\|_2}$，如果它能控制住，那么增长因子也就能控制住:

$$\frac{\|L\|_2\|L^{\mathrm{H}}\|_2}{\|A\|_2} = \frac{\|L\|_2\|L\|_2}{\|LL^{\mathrm{H}}\|_2} = \frac{\|L\|_2^2}{\|L\|_2^2} = 1$$

这表明按 Gauss 消去法计算 Cholesky 分解是数值稳定的，因此解得准不准完全取决于问题的条件数.

1.4.3 推广

为避免开方运算 (实际上大多数计算机开方运算是比较安全的，不能尽信书)，我们分解正定阵 A 为 $A = LDL^{\mathrm{T}}$
其中 L 是单位下三角阵，D 是正定对角阵.
它是 Cholesky 分解的变形，称为 **Cholesky-Banachiewicz 分解**.

其原理如下 (以 5 阶方阵为例):

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ * & 1 & & & \\ * & & 1 & & \\ * & & & 1 & \\ * & & & & 1 \end{bmatrix} \begin{bmatrix} * & & & & \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \end{bmatrix} \begin{bmatrix} 1 & * & * & * & * \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

即每次行 Gauss 消元，都进行对称的列 Gauss 消元，以保证系数矩阵的对称性不被破坏.

设 $L=\begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}$ 和 $D=\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \succ 0$

我们有 $LD=\begin{bmatrix} l_{11}d_1 & & & \\ l_{21}d_1 & l_{22}d_2 & & \\ \vdots & \vdots & \ddots & \\ l_{n1}d_1 & l_{n2}d_2 & \dots & l_{nn}d_n \end{bmatrix}$

比较 $A=LDL^{\mathrm{T}}$ 两侧的对应元素可知:

$a_{ij}=\sum\limits_{k=1}^{j}(l_{ik}d_k)l_{jk}=\sum\limits_{k=1}^{j-1}l_{ik}d_kl_{jk}+l_{ij}d_jl_{jj}=\sum\limits_{k=1}^{j-1}l_{ik}d_kl_{jk}+l_{ij}d_j\ \ (1\le j\le i\le n)$ (注意到 $l_{jj}=1$)

由此可以确定 l_{ij} 和 d_j 的计算公式 $(j=1,\dots,n)$:

$$\begin{cases} v_k=d_kl_{jk} & (k=1,\dots,j-1) \\ d_j=a_{jj}-\sum\limits_{k=1}^{j-1}l_{jk}v_k & (i=j\text{ 的情况}) \\ l_{ij}=\frac{1}{d_j}(a_{ij}-\sum\limits_{k=1}^{j-1}l_{ik}v_k) & (i=j+1,\dots,n) \end{cases}$$

上述确定 $A=LDL^{\mathrm{T}}$ 分解的方法称为**改进的平方根法**

实际计算时，我们将 L 的严格下三角元素存储在 A 的对应位置上，

同时将 D 的对角元存储在 A 的对应位置上.

(改进的平方根法, 数值线性代数, 算法 1.3.2)

$$\begin{aligned} &\text{for } j=1:n \\ &\quad \text{for } k=1:j-1 \\ &\quad\quad v(k)=A(k,k)A(j,k) \\ &\quad \text{end} \\ &\quad A(j,j)=A(j,j)-A(j,1:j-1)v(1:j-1) \\ &\quad A(j+1:n,j)=[A(j+1:n,j)-A(j+1:n,1:j-1)v(1:j-1)]/A(j,j) \\ &\text{end} \end{aligned}$$

总浮点运算数大约为:

$$\begin{aligned} &\sum_{j=1}^{n}[(j-1)+(j-1)+2(n-j-1)(j-1)] \\ &=2\sum_{j=1}^{n}(n-j)(j-1) \\ &=2[(n+1)\sum_{j=1}^{n}j-\sum_{j=1}^{n}j^2-\sum_{j=1}^{n}n] \\ &=2[(n+1)\cdot\frac{1}{2}n(n+1)-\frac{1}{6}n(n+1)(2n+1)-n^2] \\ &=O(\frac{1}{3}n^3) \end{aligned}$$

改进的平方根法计算复杂度也是 $O(\frac{1}{3}n^3)$，而且还不需要开方运算，因此比平方根法更加实用.

(但大多数现代计算机都已经硬件实现了开方运算，因此平方根法也是可以直接用的. 尽信书不如无书)

若方阵 $A\in\mathbb{R}^{n\times n}$ 对称正定，则我们可以这样求解线性方程组 $Ax=b$:

- 改进的平方根法计算 A 的 Cholesky-Banachiewicz 分解 $A=LDL^{\mathrm{T}}$
- 前代法求解 $Ly=b$ 得到 y
- 回代法求解 $DL^{\mathrm{T}}x=y$ (即 $L^{\mathrm{T}}x=D^{-1}x$) 得到 x

---

实际上 Cholesky-Banachiewicz 分解可以用于求解对称不定线性方程组.

其原理如下 (以 5 阶方阵为例):

- ① 若对角元是非零元，则我们可以进行对称行列 Gauss 消元:

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ * & 1 & & & \\ * & & 1 & & \\ * & & & 1 & \\ * & & & & 1 \end{bmatrix} \begin{bmatrix} * & & & & \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \end{bmatrix} \begin{bmatrix} 1 & * & * & * & * \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

- ② 对于第一个对角元为 0 的情况:
  - 若其他对角元比较大，则我们可使用对称行列置换将当前的 0 对角元替换为模最大的对角元
    然后进行对称行列 Gauss 消元.
  - 若其他对角元均为 0，则我们可以在该列寻找模最大的非零元素，
    使用对称行列置换将其放到紧贴当前的 0 对角元的位置.
    然后取该 2×2 分块进行对称行列 Gauss 消元:
    (简单起见，假设 $(2,1)$ 位置的元素为第 1 列模最大的非零元素，这样不需进行对称行列置换)

$$\begin{bmatrix} 0 & * & * & * & * \\ * & 0 & * & * & * \\ * & * & 0 & * & * \\ * & * & * & 0 & * \\ * & * & * & * & 0 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ * & * & 1 & & \\ * & * & & 1 & \\ * & * & & & 1 \end{bmatrix} \begin{bmatrix} 0 & * & & & \\ * & 0 & & & \\ & & * & * & * \\ & & * & * & * \\ & & * & * & * \end{bmatrix} \begin{bmatrix} 1 & & * & * & * \\ & 1 & * & * & * \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

最终得到 $A = LDL^{\mathrm{T}}$，其中 L 为单位下三角阵，而 D 为分块对角阵 (对角块要么是 1×1 的，要么是 2×2，因此其逆是好求的)
于是我们可以这样求解线性方程组 $Ax = b$:

- 计算 A 的 Cholesky-Banachiewicz 分解 $A = LDL^{\mathrm{T}}$
- 前代法求解 $Ly = b$ 得到 y
- 计算 D^{-1}，并使用回代法求解 $L^{\mathrm{T}}x = D^{-1}y$ 得到 x

The End