第二讲　线性方程组直接解法

2 特殊方程组的求解

2.1 对称正定线性方程组

我们首先给出对称正定矩阵的几个基本性质.

- 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

$$\max_{i\neq j}\{|a_{ij}|\} < \max_{i}\{a_{ii}\},$$

即绝对值最大的元素出现在对角线上.

Cholesky 分解

定理 (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**.

(板书)

Cholesky 分解的实现

设 $A = LL^\intercal$, 即

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

直接比较等式两边的元素可得

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

根据上面的计算公式, 可得下面的算法:

算法 2.1 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**

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj}$$

end for

end for

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 $\frac{1}{3}n^3 + O(n^2)$, 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
- 基于 Cholesky 分解的求解方法称为 **平方根法**

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

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

基于该分解的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 2.2 改进的平方根法

for $j = 1$ to n **do** % 先计算分解

  $d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k$

  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

$y_1 = b_1$ % 解方程组: $Ly = b$ 和 $DL^\intercal x = y$

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 对称不定线性方程组 *

$A \to$ 非奇异, **对称** 但 **不定**

若 A 存在 LU 分解, 即 $A = LU$, 则可写成 $\boxed{A = LDL^{\mathsf{T}}}$

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.

* 本小节内容只需了解。

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

$$PAP^{\mathsf{T}} = LDL^{\mathsf{T}}. \tag{2.4}$$

通常称 (2.4) 为对称矩阵的 $\mathrm{LDL}^{\mathrm{T}}$ 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 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 $\mathrm{LDL}^{\mathrm{T}}$ 分解.

方法一：Aasen 算法

1971 年, Aasen 提出了下面的分解

$$\boxed{PAP^\intercal = LTL^\intercal},$$

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**.

该分解本质上与部分选主元 LU 分解是一样的,

思考：该分解怎么实现?

方法二：块 LDL$^\mathrm{T}$ 分解

设 A 对称非奇异, 则存在置换矩阵 P 使得

$$PAP^\intercal = \begin{bmatrix} B & E^\intercal \\ E & C \end{bmatrix}, \qquad B \in \mathbb{R} \text{ 或 } B \in \mathbb{R}^{2\times 2}, \text{且非奇异.}$$

思考：如何实现?

于是可以对 $PAP^\intercal$ 进行块对角化, 即

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

其中 $C - EB^{-1}E^\intercal$ 是 Schur 补, 对称非奇异.

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

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

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

选主元块 $\mathrm{LDL}^{\mathsf{T}}$ 分解

实际使用时，需要考虑选主元策略, 目前常用的策略有:

- 全主元策略: 由 Bunch 和 Parlett 于 1971 年提出, 并证明了其稳定性. 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 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.
- Rook 策略: 由 Ashcraft, Grimes 和 Lewis 于 1998 年提出, 整体上与部分选主元类似, 但在选主元时加了一层迭代, 精度更高.

✍ **目前大部分软件都采用部分选主元块 $\mathrm{LDL}^{\mathsf{T}}$ 分解算法.**

2.3 三对角线性方程组

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix}.$$

我们假定

$$|b_1| > |c_1| > 0, \quad |b_n| > |a_{n-1}| > 0, \tag{2.5}$$

$$|b_i| \geq |a_{i-1}| + |c_i|, \quad a_i c_i \neq 0, \quad i = 1, \dots, n-1. \tag{2.6}$$

即 A 是 **不可约弱对角占优**

思考：如果 A 可约, 怎么处理? (什么情况下，三对角矩阵可约?)

此时, 我们可以得到下面的三角分解

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

得递推公式:

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

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

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

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 (2.5) 和 (2.6). 则 A 非奇异, 且

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

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

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

(板书)

算法 2.3 追赶法

$\beta_1 = c_1/b_1$

$y_1 = f_1/b_1$

for $i = 2$ to $n-1$ **do**

  $\alpha_i = b_i - a_{i-1}\beta_{i-1}$

  $\beta_i = c_i/\alpha_i$

  $y_i = (f_i - a_{i-1}y_{i-1})/\alpha_i$

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

✍ **追赶法** (也称为 **Thomas 算法**) 的运算量大约为 $8n$.

✍ 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

$$A=\begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix}=\begin{bmatrix} 1 & & & \\ \gamma_1 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_{n-1} & 1 \end{bmatrix}\begin{bmatrix} \alpha_1 & c_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & \alpha_n \end{bmatrix}. \tag{2.11}$$

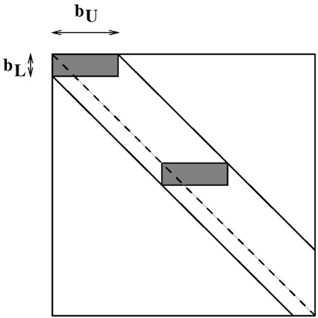
但此时 $|\gamma_i|$ 可能大于 1. 比如当 $|b_1|<|a_1|$ 时, $|\gamma_1|=|a_1/b_1|>1$. 所以在回代求解时, 误差可能得不到有效控制. 同时, 计算 γ_i 时也可能会产生较大误差 (大数除小数). **如果 A 列对角占优, 则可以保证 $|\gamma_i|<1$.**

✍ 所以，如果 A 是 (行) 对角占优, 则采用前面的分解; 如果 A 是**列对角占优**, 则采用分解 (2.11).

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

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



* 本小节内容只需了解。

对于带状矩阵, 其 LU 分解有如下性质:

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.

练习: 统计求解带状矩阵 $Ax = b$ 的运算量.

若采用部分选主元的 LU 分解, 则有

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.

3 扰动分析 *

3.1 δx 与 $\hat{x}$ 的关系

3.2 δx 与 x_* 的关系

3.3 δx 与残量的关系

* 本节内容只需了解。

扰动方程

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.

向后误差分析 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.

3.1 δx 与 $\hat{x}$ 的关系

定理 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

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

当 $\delta b = 0$ 时, 有

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

3.2 δx 与 x_* 的关系

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

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

如果 $\|\delta A\| = 0$, 则

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

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

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

✍ 上述定理中的结论对所有 p-范数都成立.

3.3 δx 与残量的关系

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

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

实际计算中 r 是可以计算的, 因此比较实用.

4 解的改进 *

* 本节内容只需了解。

4.1 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.

4.2 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.

✍ 缩放技术在工程计算时经常会用到.

例 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 法求解, 计算过程保留 8 位有效数字, 求得数值解

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

与精确解 $x = [1.9273\cdots, -0.698496\cdots, 0.9004233\cdots]^{\mathsf{T}}$ 相差较大.

考虑矩阵元素缩放, 即同乘对角阵 $D = \mathrm{diag}(0.00005, 1, 1)$, 得新方程组

$$DADy = Db.$$

最后令 $\tilde{x} = Dy$, 即可求得比较精确的数值解.

思考: 试比较 A 和 DAD 的条件数.

4.3 迭代改进法

设近似解 $\hat{x}$, 如果 $\hat{x}$ 没达到精度要求, 可以考虑对其进行修正

$$\tilde{x} = \hat{x} + z,$$

即加上一个向量 z, 得到新的近似解 $\tilde{x}$.

怎么选取 z

为了使得 $\tilde{x}$ 尽可能地接近真解，我们希望 $\tilde{x}$ 能满足 $A\tilde{x} = b$，即

$$\boxed{A(\hat{x}+z) = b \Longleftrightarrow Az = r} \quad \rightarrow \text{ 残量方程}$$

其中 $\boxed{r = b - A\hat{x}}$ 是残量.

在实际计算中, 我们只能得到近似解 $\hat{z}$, 但 $\|r - A\hat{z}\|$ 会比较小, 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 $\tilde{x} \triangleq \hat{x} + \hat{z}$ 还不满足精度要求, 则可重复以上过程.

算法 4.1 通过迭代改进解的精度

设 $PA = LU$, $\hat{x}$ 是 $Ax = b$ 的近似解
while 近似解 $\hat{x}$ 不满足精度要求, **do**
    计算 $r = b - A\hat{x}$
    求解 $Ly = Pr$, 即 $y = L^{-1}Pr$
    求解 $Uz = y$, 即 $z = U^{-1}y$
    令 $\hat{x} = \hat{x} + z$
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 $O(n^2)$. 所以相对来讲还是比较经济的.

✍ 为提高计算精度, 在计算残量 r 时最好使用原始数据 A, 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. 这里 ε_u 表示机器精度.

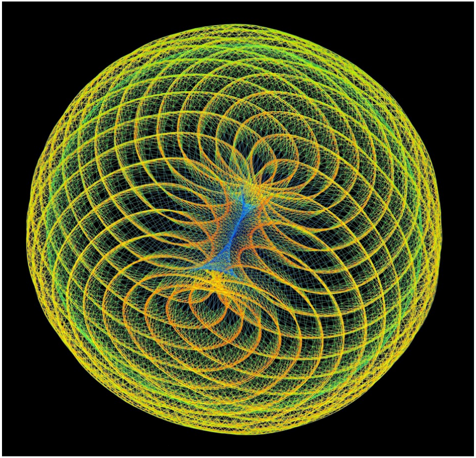
但 $\varepsilon_u \kappa_\infty(A) \geq 1$ 时, 效果可能不佳，得不到任何改善.

大规模稀疏线性方程组的直接解法

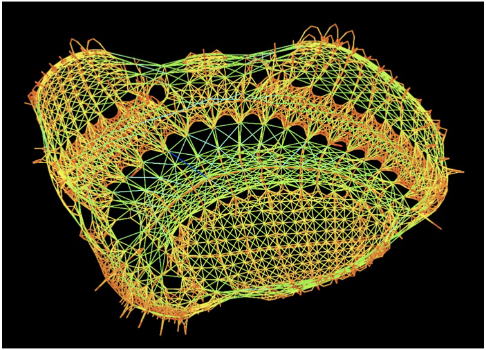
存在的困难

- 规模大, 百万量级以上或以亿为单位.
- LU/Cholesky 分解无法保持充分的稀疏性, 可能出现大量非零元, 导致存在困难, 即存储爆炸, 同时也无法有效降低运算量.

Example sparse matrices from THE SUITESPARSE MATRIX COLLECTION



A Barrier Hessian Matrix from Convex Quadratic Programming



The structural loading of an exhibition hall in Beijing

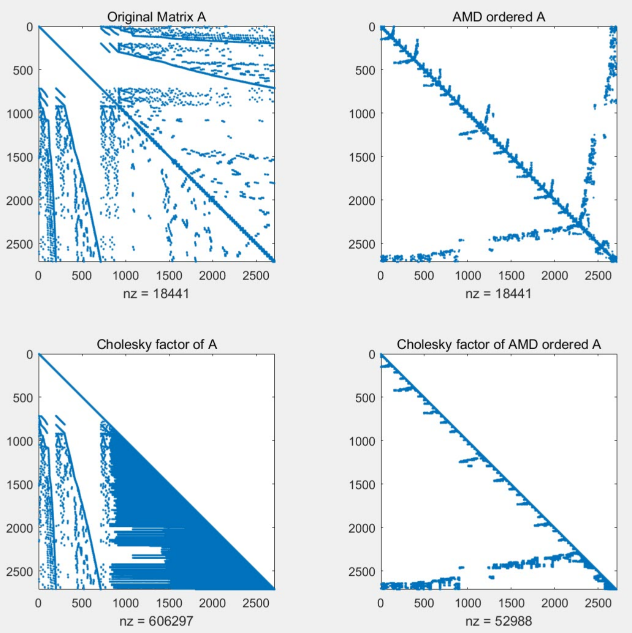
目前的策略

(1) 对矩阵进行重排序, 使得新矩阵的 LU/Cholesky 分解具有更好的稀疏性.

(2) 利用稀疏矩阵的重复结构: Supernodal, Frontal, Multifrontal methods

(3) 并行计算, GPU 加速, 低秩逼近, update/downdate methods

Davis, Rajamanickam and Sid-Lakhdar, A survey of direct methods for sparse linear systems, *Acta Numerica*, 2016.


Original Matrix A
nz = 18441
AMD ordered A
nz = 18441
Cholesky factor of A
nz = 606297
Cholesky factor of AMD ordered A
nz = 52988