

数值分析第一次上机作业

宋歌 2015080086 数 52

10/21/2018

设 $\mathbf{H}_n = [h_{ij}] \in \mathbb{R}^{n \times n}$ 是 Hilbert 矩阵, 即

$$h_{ij} = \frac{1}{i+j-1}$$

对 $n = 2, 3, 4, \dots$

1 Gauss 消元法与 Cholesky 分解

用 Gauss 消元法和 Cholesky 分解法分别求解 $\mathbf{H}_n \mathbf{y} = \mathbf{b}_n$, 并计算误差。

1.1 实验原理

用 Gauss 消元法求解 $\mathbf{Ax} = \mathbf{b}$ 的基本思想就是用第一个方程消去后面 $n-1$ 个方程的第一个未知数, 再利用第二个方程消去后面 $n-2$ 个方程中的第二个未知数, 以此类推, 直到把系数矩阵 \mathbf{A} 变成上三角阵的形式。变成上三角阵后, 从最后一个方程开始, 解出 x_n , 代入前一个方程解出 x_{n-1} ……最终解出 x_1, \dots, x_n 。

用 Cholesky 分解法求解 $\mathbf{Ax} = \mathbf{b}$ 是利用了定理: 若 \mathbf{A} 为对称正定阵, 则 \mathbf{A} 可唯一地分解为 $\mathbf{A} = \mathbf{LL}^T$, 其中 \mathbf{L} 为对角元为正的下三角阵。

1.2 实验设计

根据以上实验原理, 我们可以得到如图 2 和图 1 的算法设计。以此构造基于两个不同方法求解方程 $\mathbf{Ax} = \mathbf{b}$ 的两个函数 Gauss(\mathbf{A}, \mathbf{b}) 和 Chol(\mathbf{A}, \mathbf{b})。为了得到两种方法之下的误差, 我们对每一个 $\mathbf{H}_n, n = 1, 2, \dots, 20$ 都设定好人工的精确解 $\mathbf{x}_n = (1, 1, \dots, 1) \in \mathbb{R}^n$, 并得到相应的 $\mathbf{b}_n = \mathbf{H}_n \mathbf{x}_n$: 之后我们对每一个 $n = 1, 2, \dots, 20$ 依次用两个函数求解 $\mathbf{H}_n \mathbf{y}_n = \mathbf{b}_n$, 最终可以得到误差 $\|\mathbf{x}_n - \mathbf{y}_n\|$ 和每一次求解方程的时间。实验主要部分的代码如图 3。

算法 2.4 (平方根法)

① (*Cholesky* 分解) 对 $j = 1, \dots, n$:

$$\begin{cases} l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{1/2}, \\ l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}, \quad i = j+1, \dots, n \end{cases}$$

② 先后求解 $\mathbf{Ly} = \mathbf{b}$, 及 $\mathbf{L}^T \mathbf{x} = \mathbf{y}$:

$$\begin{aligned} y_i &= (b_i - \sum_{k=1}^{i-1} l_{ik} y_k) / l_{ii}, \quad i = 1, 2, \dots, n, \\ x_i &= (y_i - \sum_{k=i+1}^n l_{ki} x_k) / l_{ii}, \quad i = n, n-1, \dots, 1. \end{aligned}$$

Figure 1: Cholesky 算法设计

算法 2.1 (Gauss顺序消元算法)

① 对 $k = 1, \dots, n-1$ 做以下消元步骤

$$l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}, \quad i = k+1, \dots, n.$$

$$\begin{cases} a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & i = 1, \dots, k; \quad j = 1, \dots, n \\ 0, & i = k+1, \dots, n; \quad j = 1, \dots, k, \\ a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}, & i = k+1, \dots, n; \quad j = k+1, \dots, n; \end{cases} \\ b_i^{(k+1)} = \begin{cases} b_i^{(k)}, & i = 1, \dots, k; \\ b_i^{(k)} - l_{ik}b_k^{(k)}, & i = k+1, \dots, n. \end{cases} \end{cases}$$

② 回代:
$$\begin{cases} x_n = b_n^{(n)} / a_{nn}^{(n)}, \\ x_i = \frac{1}{a_{ii}^{(i)}} \left(b_i^{(i)} - \sum_{j=i+1}^n a_{ij}^{(i)} x_j \right), \quad i = n-1, n-2, \dots, 1. \end{cases}$$

Figure 2: Gauss 算法设计

```
% initialization
M = 20;
er_G = zeros(M, 1);
er_C = zeros(M, 1);
t_G = zeros(M, 1);
t_C = zeros(M, 1);

for n = 1:M
    % initialization
    H = hilb(n); % generate Hilbert matrix
    x = ones(n, 1);
    b = H * x;

    % Gauss elimination
    [x_G, t_G(n)] = Gauss(H, b);
    er_G(n) = norm(x_G - x, 2);

    % Cholesky decomposition
    [x_C, t_C(n)] = Chol(H, b);
    er_C(n) = norm(x_C - x, 2);

    % condition number
    cond2(n) = cond(H, 2);
end
```

Figure 3: 第一题实验主要部分代码

1.3 实验结果

按照以上实验设计，我们得到如图 4 的实验结果，主要反映了误差、时间随着 Hilbert 矩阵维数的变化。我们看到误差和计算时间都随着矩阵维数的增大而增大，同时，Cholesky 分解法总体上误差比 Gauss 消元法小，时间也更少。

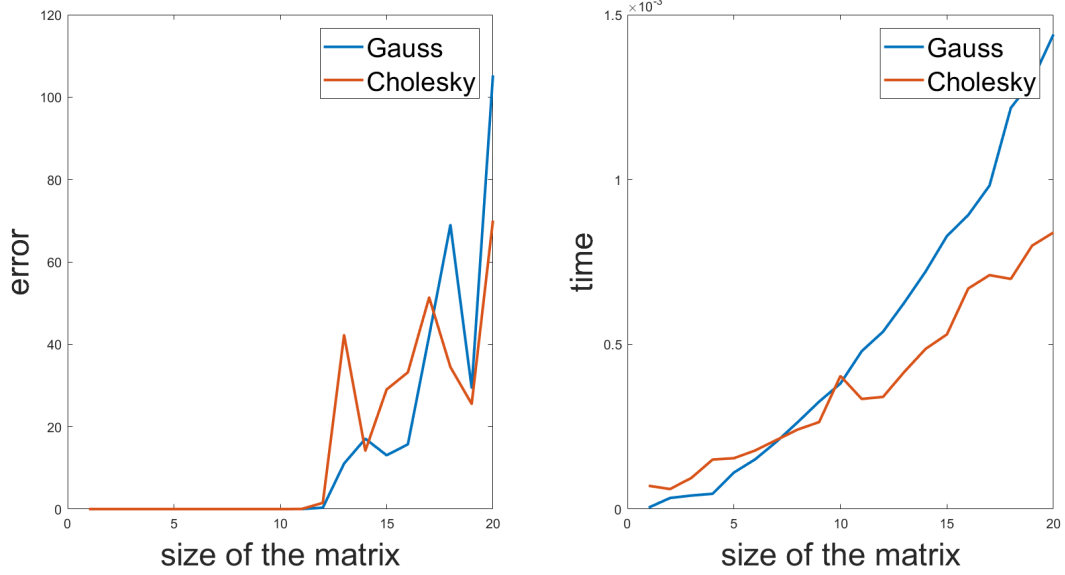


Figure 4: 以上左图反映了误差随着矩阵维数的变化，右图反映了解方程时间随着矩阵维数的变化。图中蓝色折线代表 Gauss 消元法，红色折线代表 Cholesky 分解法。

2 矩阵条件数

计算 Hilbert 矩阵的条件数，结果如下：

$\text{cond}(\mathbf{H}_1)_2 = 1$	$\text{cond}(\mathbf{H}_{11})_2 = 522020733204515$
$\text{cond}(\mathbf{H}_2)_2 = 19.2814700679040$	$\text{cond}(\mathbf{H}_{12})_2 = 1.62116390474750e + 16$
$\text{cond}(\mathbf{H}_3)_2 = 524.056777586063$	$\text{cond}(\mathbf{H}_{13})_2 = 4.78639230237801e + 17$
$\text{cond}(\mathbf{H}_4)_2 = 15513.7387389290$	$\text{cond}(\mathbf{H}_{14})_2 = 2.55149884837821e + 17$
$\text{cond}(\mathbf{H}_5)_2 = 476607.250241000$	$\text{cond}(\mathbf{H}_{15})_2 = 2.49595175000979e + 17$
$\text{cond}(\mathbf{H}_6)_2 = 14951058.6414534$	$\text{cond}(\mathbf{H}_{16})_2 = 4.88749497963028e + 17$
$\text{cond}(\mathbf{H}_7)_2 = 475367356.911439$	$\text{cond}(\mathbf{H}_{17})_2 = 4.51443968392446e + 17$
$\text{cond}(\mathbf{H}_8)_2 = 15257575566.6280$	$\text{cond}(\mathbf{H}_{18})_2 = 1.35002449311858e + 18$
$\text{cond}(\mathbf{H}_9)_2 = 493153404551.012$	$\text{cond}(\mathbf{H}_{19})_2 = 1.25785827806332e + 18$
$\text{cond}(\mathbf{H}_{10})_2 = 16025028168113.2$	$\text{cond}(\mathbf{H}_{20})_2 = 2.10653055807044e + 18$

3 吉洪诺夫正则化方法

3.1 实验原理

设 μ_1, \dots, μ_n 是矩阵 $\mathbf{A} \in \mathbb{C}^{m \times n}$ 的奇异值, 则有如图 3.1 的定理, 这表明, 求解 $\mathbf{Ax} = \mathbf{b}$ 可以用 $(\alpha \mathbf{I} + \mathbf{A}^* \mathbf{A})\mathbf{x}_\alpha = \mathbf{A}^* \mathbf{b}$ 来近似。 α 充分小的时候, \mathbf{x}_α 会是 \mathbf{x} 的一个好的近似。 同时由于 $\text{cond}_2(\alpha \mathbf{I} + \mathbf{A}^* \mathbf{A}) = (\alpha + \mu_1^2)/(\alpha + \mu_n^2)$, $\text{cond}_2(\mathbf{A}) = \mu_1/\mu_n$, 我们希望 $\alpha > \mu_1 \mu_n$

定理 2.13

设 $A \in \mathbb{C}^{m \times n}$ 秩为 r , $\forall \alpha > 0, \forall \mathbf{b} \in \mathbb{C}^m$, 方程组

$$(\alpha I + A^* A)\mathbf{x}_\alpha = A^* \mathbf{b} \text{ 存在唯一解: } \mathbf{x}_\alpha = \sum_{j=1}^r \frac{\mu_j}{\alpha + \mu_j^2} (\mathbf{b}, \mathbf{v}_j) \mathbf{u}_j \in \mathbb{C}^n.$$

3.2 实验设计

在第一题的基础上, 我们在做 Gauss 消元法和 Cholesky 分解法之前, 先对矩阵 \mathbf{H}_n 进行正则化处理, 即把求解 $\mathbf{H}_n \mathbf{x} = \mathbf{b}$ 换为求解 $(\alpha \mathbf{I}_n + \mathbf{H}_n^* \mathbf{H}_n)\mathbf{x}_\alpha = \mathbf{H}_n^* \mathbf{b}_n$ 。 实验主要部分代码如图 5

```
for n = 1:M
    % initialization
    H = hilb(n); % generate Hilbert matrix
    x = ones(n,1);
    b = H * x;
    %s = svd(H);
    alpha = 0.001;
    A = alpha * eye(n) + H' * H;

    % condition number
    cond2(n) = cond(A, 2);

    % Gauss elimination
    [x_G, t_G(n)] = Gauss(A, H' * b);
    er_G(n) = norm(x_G - x, 2);

    % Cholesky decomposition
    [x_C, t_C(n)] = Chol(A, H' * b);
    er_C(n) = norm(x_C - x, 2);
end
```

Figure 5: 第三题实验主要部分代码

3.3 实验结果

按照以上实验设计，我们得到如图 6 的实验结果，主要反映了误差、时间随着 Hilbert 矩阵维数的变化。除了和图 4 相似的趋势以外，我们看到，做了吉洪诺夫正则化之后，误差和时间都显著下降。

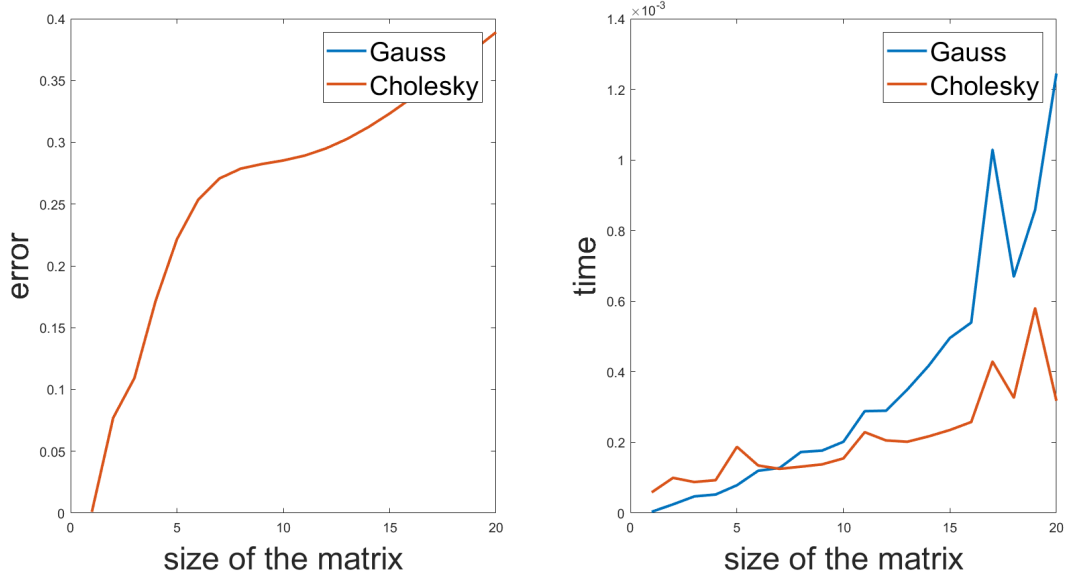


Figure 6: 以上左图反映了误差随着矩阵维数的变化，右图反映了解方程时间随着矩阵维数的变化。图中蓝色折线代表 Gauss 消元法，红色折线代表 Cholesky 分解法。左图中，两种方法的误差曲线基本重合。

4 SOR, 共轭梯度法, GMRES

4.1 实验原理

SOR 迭代法：设矩阵 \mathbf{A} 有分解 $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ ，其中 \mathbf{D} 为对角元部分， $-\mathbf{L}$ 为下三角部分， $-\mathbf{U}$ 为上三角部分；对于对称正定阵来说， \mathbf{D} 可逆，因此我们得到如下关系式：

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

进一步有分量形式

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)} \right)$$

我们引入一个松弛参数即得到 SOR 迭代法的迭代公式：

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)} \right)$$

共轭梯度法：已知最速下降法是在迭代步骤 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ 中，把搜索方向 $\mathbf{p}^{(k)}$ 取负梯度方向作 $\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ 。但随着迭代次数增多，由于舍入误差的影响，每次搜索方向与实际残差方向有很大的差别，而共轭梯度法做出的改进是：使得每一次搜索之后找到的都是前面所有搜索方向组成的线性空间中的最小值点。即保证了残差向量 $\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ 都是相互正交的，且搜索方向 $\mathbf{p}^{(k)}$ 都是 \mathbf{A} 共轭的。

GMRES：将问题 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 转化为求解 $\mathbf{A}\mathbf{z} = \mathbf{r}_0$ ，其中 $0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \mathbf{x} = \mathbf{x}_0 + \mathbf{z}$ 。根据 Galerkin 原理在 m 维子空间中找到 $\mathbf{A}\mathbf{z} = \mathbf{r}_0$ 的近似解。

4.2 实验设计

根据以上实验原理，我们可以得到如图 7和图 8和图 9的算法设计。以此构造基于三个不同迭代方法求解方程 $\mathbf{Ax} = \mathbf{b}$ 的三个函数 $\text{SOR}(\mathbf{A}, \mathbf{b}, \omega)$ 和 $\text{Conj}(\mathbf{A}, \mathbf{b})$ 和 $\text{GMRES}(\mathbf{A}, \mathbf{b}, m)$ 。将用吉洪诺夫正则化方法改进后的 Gauss 消元法和 Cholesky 分解法加入进行对比，与题一类似，可以得到如图 10的实验主要部分的代码。

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^n a_{ij} x_j^{(k)} \right),$$

称之为 **SOR (Successive Over Relaxation, 逐次超松弛)方法**。

Figure 7: SOR 迭代法算法设计

算法 3.3 (共轭梯度法)

取 $\mathbf{x}^{(0)} \in \mathbb{R}^n$, 计算残差 $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$. 令 $k = 0$, $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$.

- ① 计算 $\alpha_k = \frac{\|\mathbf{r}^{(k)}\|_2^2}{(A\mathbf{p}^{(k)}, \mathbf{p}^{(k)})}$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$.

若 $\|\alpha_k \mathbf{p}^{(k)}\| < \varepsilon$ 则停止迭代; 否则继续.

- ② 计算新残差 $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)}$, 计算 $\beta_k = \frac{\|\mathbf{r}^{(k+1)}\|_2^2}{\|\mathbf{r}^{(k)}\|_2^2}$,
[利用了 $A\mathbf{p}^{(j)} = \frac{1}{\alpha_j}(\mathbf{r}^{(j)} - \mathbf{r}^{(j+1)})$] $\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k \mathbf{p}^{(k)}$.

- ③ $\mathbf{r}^{(k+1)}$ 或 $\mathbf{p}^{(k+1)} = 0$ 或 $k = n - 2$ 停止; 否则 $k = k + 1$, 转1

$$\alpha_{n-1} = \frac{\|\mathbf{r}^{(n-1)}\|_2^2}{(A\mathbf{p}^{(n-1)}, \mathbf{p}^{(n-1)})}, \mathbf{x}^{(n)} = \mathbf{x}^{(n-1)} + \alpha_{n-1} \mathbf{p}^{(n-1)}.$$

Figure 8: 共轭梯度法算法设计

算法 3.8 (GMRES(m) 算法)

- ① 选择 $\mathbf{x}_0 \in \mathbb{R}^n$ 并计算 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, 记 $\|\mathbf{r}_0\| = \beta$, $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$.
选择适当大小的 m .

- ② 用 Arnoldi 过程求出 $\{\mathbf{v}_i\}_{i=1}^m$ 和 \tilde{H}_m .

- ③ 求解最小二乘问题 $\min_{\mathbf{y}_m \in \mathbb{R}^m} \|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}_m\|$ 得到 \mathbf{y}_m .

- ④ 计算 $\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m$, $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$, $\mathbf{r}_0 = \mathbf{r}_m$.

- ⑤ 如果 $\beta = \|\mathbf{r}_m\| < \varepsilon$, 则停止迭代;
否则 $\mathbf{x}_0 = \mathbf{x}_m$, $\mathbf{v}_1 = \mathbf{r}_m / \|\mathbf{r}_m\|$, 转第 2 步.

Figure 9: GMRES 算法设计

```

M = 20; omega = 1;
er_G = zeros(M,1);er_C = zeros(M,1);er_sor = zeros(M,1);
er_cj = zeros(M,1);er_gm = zeros(M,1);
t_G = zeros(M,1);t_C = zeros(M,1);t_sor = zeros(M,1);
t_cj = zeros(M,1);t_gm = zeros(M,1);

for n = 1:M
    H = hilb(n); x = ones(n,1); b = H * x;
    alpha = 0.001; A = alpha * eye(n) + H'*H;

    [x_G, t_G(n)] = Gauss(A, H' * b); er_G(n) = norm(x_G - x, 2);
    [x_C, t_C(n)] = Chol(A, H' * b); er_C(n) = norm(x_C - x, 2);
    [x_sor, t_sor(n)] = SOR(H,b,omega); er_sor(n) = norm(x_sor - x, 2);
    [x_cj, t_cj(n)] = Conj(H,b); er_cj(n) = norm(x_cj - x, 2);

    if n <= 2
        m = n;
    else
        m = n - 2;
    end
    [x_gm, t_gm(n)] = GMRES(H,b,m); er_gm(n) = norm(x_gm - x, 2);
end

```

Figure 10: 第四题实验主要部分代码

4.3 实验结果

按照以上实验设计，我们得到如图 11 的实验结果。从误差来看，GMRES 最优，共轭梯度法最差；从运行时间来看，共轭梯度法最快，GMRES 最差。

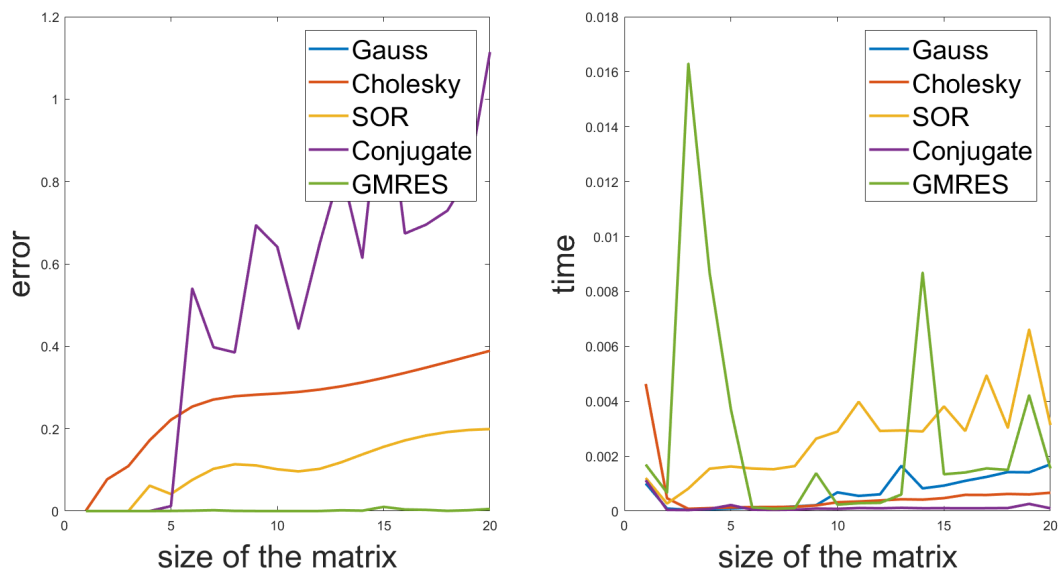


Figure 11: 以上左图反映了误差随着矩阵维数的变化，右图反映了解方程时间随着矩阵维数的变化。不同颜色代表不同的解方程的方法。左图中，Guass 和 Cholesky 的误差曲线基本重合。