第五讲 线性最小二乘问题

QR 分解

目录

5-3 QR 分解

5.3 QR 分解

QR 分解

$$\boxed{A = QR}$$

- QR 分解是将一个矩阵分解一个正交矩阵 (酉矩阵) 和一个三角矩阵的乘积.
- QR 分解被广泛应用于线性最小二乘问题的求解和矩阵特征值的计算.

5-3-1 QR 分解的存在性与唯一性

定理 5.1 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$, 则存在一个单位列正交矩阵 $Q \in \mathbb{C}^{m\times n}$ (即 $Q^*Q = I_{n\times n}$) 和一个上三角矩阵 $R \in \mathbb{C}^{n\times n}$, 使得

$$\boxed{A = QR} \tag{5.1}$$

若 A 列满秩, 则存在一个具有正对角线元素的上三角矩阵 R 使得 (5.1) 成立, 且此时 QR 分解唯一, 即 Q 和 R 都唯一.

存在性 可通过构造法证明.

假定 A 列满秩, 记 $A = [a_1, a_2, \ldots, a_n] \in \mathbb{C}^{m\times n}$, 则 QR 分解就是对 A 的列向量组进行 Gram-Schmidt 正交化过程.

算法 Gram-Schmidt 过程

$r_{11} = \|a_1\|_2,\ q_1 = a_1/r_{11}$
for $j = 2$ to n **do**
    $q_j = a_j$
    for $i = 1$ to $j-1$ **do**
        $r_{ij} = q_i^* a_j$ % q_i^* 表示共轭转置
        $q_j = q_j - r_{ij} q_i$
    end for
    $r_{jj} = \|q_j\|_2$
    $q_j = q_j / r_{jj}$
end for

由 Gram-Schmidt 正交化过程可知

$$a_1 = r_{11} q_1, \quad a_j = r_{1j} q_1 + r_{2j} q_2 + \cdots + r_{jj} q_j = [q_1, q_2, \ldots, q_j] \begin{bmatrix} r_{1j} \\ r_{2j} \\ \vdots \\ r_{jj} \end{bmatrix}$$

记 $Q = [q_1, q_2, \ldots, q_n]$, $R = [r_{ij}]_{n \times n}$, 其中

$$r_{ij} = \begin{cases} q_i^* a_j, & \text{for } i \leq j \\ 0, & \text{for } i > j \end{cases} \tag{5.2}$$

则

$$[a_1, a_2, \ldots, a_n] = [q_1, q_2, \ldots, q_n] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & r_{n-1,n} \\ & & & r_{nn} \end{bmatrix}, \text{ 即 } A = QR$$

QR 分解

- A 不满秩情形: 见讲义, 留作自习.

- A 满秩时的 QR 分解的存在唯一性: 见讲义, 留作自习.

QR 分解的另一种形式

有时也将 QR 分解定义为: 存在酉矩阵 $Q \in \mathbb{C}^{m\times m}$ 使得

$$\boxed{A = QR},$$

其中 $R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix} \in \mathbb{C}^{m\times n}$ 是上三角矩阵.

A 不是满秩矩阵情形

- 设 $\mathrm{rank}(A) = r \leq n$, 则存在置换矩阵 P, 使得 AP 的前 r 列是线性无关.

因此我们可以对 AP 进行 QR 分解.

A 不是满秩矩阵情形

设 $\mathrm{rank}(A)=r\le n$, 则存在置换矩阵 P, 使得 AP 的前 r 列是线性无关.

因此我们可以对 AP 进行 QR 分解.

推论 5.2 设 $A\in\mathbb{C}^{m\times n}$ $(m\ge n)$, 且秩为 r $(0\le r\le n)$, 则存在一个置换矩阵 P, 使得

$$AP=Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}_{n\times n},$$

其中 $Q\in\mathbb{C}^{m\times n}$ 单位列正交, $R_{11}\in\mathbb{C}^{r\times r}$ 是非奇异上三角矩阵.

A 不是满秩矩阵情形

设 $\mathrm{rank}(A) = r \le n$, 则存在置换矩阵 P, 使得 AP 的前 r 列是线性无关.

因此我们可以对 AP 进行 QR 分解.

推论 5.2 设 $A \in \mathbb{C}^{m\times n}$ $(m \ge n)$, 且秩为 r $(0 \le r \le n)$, 则存在一个置换矩阵 P, 使得

$$AP = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}_{n\times n},$$

其中 $Q \in \mathbb{C}^{m\times n}$ 单位列正交, $R_{11} \in \mathbb{C}^{r\times r}$ 是非奇异上三角矩阵.

✍ 上述结论也可简化为

$$AP = \tilde{Q} \begin{bmatrix} R_{11} & R_{12} \end{bmatrix}, \quad \text{其中} \quad \tilde{Q} \in \mathbb{C}^{m\times r}$$

几点注记

- ✍ 若 A 是实矩阵, 则 Q 和 R 都可以是实矩阵.
- ✍ 如果 A 是非奇异的方阵, 则 QR 分解也可以用来求解线性方程组 $Ax = b$.
- ✍ 基于 GS 正交化的 QR 分解算法的运算量大约为 $2mn^2$.

怎么计算 QR 分解

三种方案

MGS,　Householder,　Givens

5-3-2 基于 MGS 的 QR 分解

为了数值稳定性, 实际计算中不能直接用 GS 过程, 需要用**修正的 GS 过程** (MGS).

- Gram-Schmidt 过程的第 j 步:

  (1) $\boxed{\text{计算 } r_{ij} = q_i^* a_j}$, $\quad i = 1, 2, \ldots, j-1$;

  (2) $\boxed{\text{计算 } \tilde{q}_j = a_j - r_{1j}q_1 - r_{2j}q_2 - \cdots - r_{j-1,j}q_{j-1};}$

  (3) 计算 $r_{jj} = \|\tilde{q}_j\|, \quad q_j = \tilde{q}_j / r_{jj}$;

- Modified Gram-Schmidt 过程的第 j 步:

  (1) 令 $\tilde{q}_j = a_j$;

  (2) $\boxed{\text{计算 } r_{ij} = q_i^* \tilde{q}_j, \quad \tilde{q}_j = \tilde{q}_j - r_{ij}q_i}$, $\quad i = 1, 2, \ldots, j-1$;

  (3) 计算 $r_{jj} = \|\tilde{q}_j\|, \quad q_j = \tilde{q}_j / r_{jj}$;

算法 基于 MGS 的 QR 分解

Set $R = [r_{ik}] = 0_{n\times n}$ (the $n \times n$ zero matrix)
if $a_1 = 0$ **then**
    $q_1 = 0$
else
    $r_{11} = \|a_1\|_2, \quad q_1 = a_1/\|a_1\|_2$
end if
for $k = 2$ to n **do**
    $q_k = a_k$
    for $i = 1$ to $k-1$ **do** % MGS 过程, 注意与 GS 的区别
        $r_{ik} = q_i^{\intercal} q_k, \quad q_k = q_k - r_{ik} q_i$
    end for
    if $q_k \neq 0$ **then**
        $r_{kk} = \|q_k\|_2, \quad q_k = q_k/r_{kk}$
    end if
end for

由 MGS 得到的 QR 分解中, $Q \in \mathbb{R}^{m\times n}$, $R \in \mathbb{R}^{n\times n}$, 运算量大约为 $2mn^2$.

5-3-3 基于 Householder 变换的 QR 分解

Householder 变换的化零功能: 将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\pm\|x\|_2 e_1$

5-3-3 基于 Householder 变换的 QR 分解

Householder 变换的化零功能: 将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\pm\|x\|_2 e_1$

不失一般性, 考虑 $m = n$ 时的情形, 即 $A \in \mathbb{R}^{n\times n}$.

5-3-3 基于 Householder 变换的 QR 分解

Householder 变换的化零功能: 将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\pm\|x\|_2 e_1$

不失一般性, 考虑 $m = n$ 时的情形, 即 $A \in \mathbb{R}^{n\times n}$.

构造 Householder 变换 $H_1 \in \mathbb{R}^{n\times n}$ 使得

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

5-3-3 基于 Householder 变换的 QR 分解

Householder 变换的化零功能: 将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\pm\|x\|_2 e_1$

不失一般性, 考虑 $m = n$ 时的情形, 即 $A \in \mathbb{R}^{n\times n}$.

构造 Householder 变换 $H_1 \in \mathbb{R}^{n\times n}$ 使得

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \Longrightarrow \quad H_1 A = \left[\begin{array}{c|ccc} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_2 & \\ 0 & & & \end{array}\right].$$

第二步

构造 Householder 变换 $\tilde{H}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$, 对 $\tilde{A}_2$ 的第一列化零:

$$\tilde{H}_2 \tilde{A}_2 = \left[\begin{array}{c|ccc} r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_3 & \\ 0 & & & \end{array}\right].$$

第二步

构造 Householder 变换 $\tilde{H}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$, 对 $\tilde{A}_2$ 的第一列化零:

$$\tilde{H}_2 \tilde{A}_2 = \left[\begin{array}{c|ccc} r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_3 & \\ 0 & & & \end{array}\right].$$

令 $H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_2 \end{bmatrix} \quad \Longrightarrow \quad H_2 H_1 A = \left[\begin{array}{cc|ccc} r_1 & \tilde{a}_{12} & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & \tilde{A}_3 & \\ 0 & 0 & & & \end{array}\right].$

第 k 步

不断重复上述过程, 得到一系列矩阵 $H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{H}_k \end{bmatrix}$, $\quad k = 1, 2, 3, 4, \ldots, n-1$, 使得

$$H_{n-1} \cdots H_2 H_1 A = \begin{bmatrix} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \cdots & \tilde{a}_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix} \triangleq R.$$

第 k 步

不断重复上述过程, 得到一系列矩阵 $H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{H}_k \end{bmatrix}, \quad k = 1, 2, 3, 4, \ldots, n-1$, 使得

$$H_{n-1} \cdots H_2 H_1 A = \begin{bmatrix} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \cdots & \tilde{a}_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix} \triangleq R.$$

易知 $H_1, H_2, \ldots, H_{n-1}$ 都是正交矩阵. 令

$$Q = (H_{n-1} \cdots H_2 H_1)^{-1} = H_1^{-1} H_2^{-1} \cdots H_{n-1}^{-1} = H_1 H_2 \cdots H_{n-1},$$

则 Q 也是正交矩阵, 且 $\boxed{A = (H_{n-1} \cdots H_2 H_1)^{-1} R = QR}$

矩阵 $Q = H_1 H_2 \cdots H_{n-1}$ 的计算

方法一: 直接计算

```
Set Q = I_n
for k = 1 to n − 1 do
    Q = Q H_k
end for
```

方法二: 向后累积法

```
Set Q = I_n
for k = n − 1 to 1 do
    Q = H_k Q
end for
```

优点: 一开始 Q 会比较稀疏, 随着迭代的进行, Q 才会慢慢变满, 实现效率较高.

几点注记

✍ 如果 $m > n$, 我们仍然可以通过上面的过程进行 QR 分解, 只是最后我们得到一个正交矩阵 $Q \in \mathbb{R}^{m\times m}$ 和一个上三角矩阵 $R \in \mathbb{R}^{m\times n}$, 使得 $A = QR$.

✍ 如果不需要生成 Q, 则 Householder-QR 的总运算量大约为 $2mn^2 - 2n^3/3$.

✍ 如果需要计算 Q, 则总运算量大约为 $4m^2n - 2mn^2 + \frac{2}{3}n^3$.

✍ 如果只需计算 Q 的前 n 列 (与 MGS-QR 一样), 则总运算量为 $4mn^2 - 4n^3/3$.

算法 基于 Householder 变换的 QR 分解 (MATLAB)

% Given $A \in \mathbb{R}^{m\times n}$, compute Q and R such that $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$

% The upper triangular part of R is stored in the upper triangular part of A

Set $Q = I_{m\times m}$

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

  $x = A(k:m, k)$

  $[\beta, v_k] =$ **House**(x)

  $A(k:m, k:n) = (I_{m-k+1} - \beta v_k v_k^\mathsf{T}) A(k:m, k:n)$

  $\qquad\qquad = A(k:m, k:n) - \beta v_k \big(v_k^\mathsf{T} A(k:m, k:n)\big)$

  $Q(:, k:m) = Q(:, k:m)(I_{m-k+1} - \beta v_k v_k^\mathsf{T})$

  $\qquad\qquad = Q(:, k:m) - \beta \big(Q(:, k:m) v_k\big) v_k^\mathsf{T}$

end for

✍ 上面只是关于 Householder-QR 的一个简单描述, 并没有考虑运算的优化问题.

5-3-4 基于 Givens 变换的 QR 分解

(1) 构造 Givens 变换 G_{21}, 作用在 A 的最前面的两行上, 使得 $G_{21}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$.

由于 G_{21} 只改变矩阵的第 1 行和第 2 行的值, 所以其它行保存不变.

5-3-4 基于 Givens 变换的 QR 分解

(1) 构造 Givens 变换 G_{21}, 作用在 A 的最前面的两行上, 使得 $G_{21}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$.

由于 G_{21} 只改变矩阵的第 1 行和第 2 行的值, 所以其它行保存不变.

(2) 构造 Givens 变换 G_{31}, 作用在 $G_{21}A$ 的第 1 行和第 3 行, 将第一列的第 3 个元素化为零.

由于 G_{31} 只改变矩阵的第 1 行和第 3 行的值, 所以第二行的零元素维持不变.

5-3-4 基于 Givens 变换的 QR 分解

(1) 构造 Givens 变换 G_{21}, 作用在 A 的最前面的两行上, 使得 $G_{21}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$.

由于 G_{21} 只改变矩阵的第 1 行和第 2 行的值, 所以其它行保存不变.

(2) 构造 Givens 变换 G_{31}, 作用在 $G_{21}A$ 的第 1 行和第 3 行, 将第一列的第 3 个元素化为零.

由于 G_{31} 只改变矩阵的第 1 行和第 3 行的值, 所以第二行的零元素维持不变.

以此类推, 我们可以构造一系列的 Givens 变换 $G_{41}, G_{51}, \ldots, G_{n1}$, 使得 $G_{n1} \cdots G_{21}A$ 的第一列中除第一个元素外, 其它元素都化为零.

5-3-4 基于 Givens 变换的 QR 分解

(1) 构造 Givens 变换 G_{21}, 作用在 A 的最前面的两行上, 使得 $G_{21}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$.

由于 G_{21} 只改变矩阵的第 1 行和第 2 行的值, 所以其它行保存不变.

(2) 构造 Givens 变换 G_{31}, 作用在 $G_{21}A$ 的第 1 行和第 3 行, 将第一列的第 3 个元素化为零.

由于 G_{31} 只改变矩阵的第 1 行和第 3 行的值, 所以第二行的零元素维持不变.

- 以此类推, 我们可以构造一系列的 Givens 变换 $G_{41}, G_{51}, \ldots, G_{n1}$, 使得 $G_{n1}\cdots G_{21}A$ 的第一列中除第一个元素外, 其它元素都化为零.
- 依此类推, 处理其他列, 直至化为上三角矩阵, 即可得 QR 分解.

Givens-QR

- ✍ 与 Householder 变换一样, 在进行 Givens 变换时, 我们不需要显式地写出 Givens 矩阵.
- ✍ 对于稠密矩阵而言, Givens-QR 的运算量比 Householder-QR 要多很多.
- ✍ 但如果 A 的非零下三角元素相对较少 (如上 Hessenberg 矩阵), 则可以采用 Givens-QR 算法.

5-3-5 QR 分解的稳定性

MGS-QR

基于 MGS 的 QR 分解也是向后稳定的:

$$Q^{\mathsf{T}}Q = I + E_{MGS} \quad 其中 \quad \|E_{MGS}\|_2 \approx \varepsilon_u \kappa_2(A).$$

Householder-QR, Givens-QR

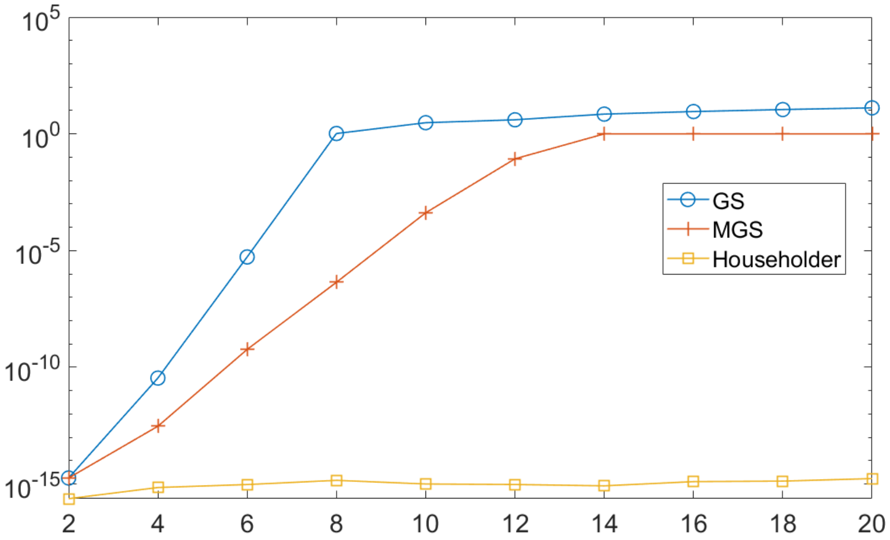
基于 Householder 变换和 Givens 变换的 QR 分解都具有很好的数值稳定性:

$$Q^{\mathsf{T}}Q = I + E_H \quad 其中 \quad \|E_H\|_2 \approx \varepsilon_u.$$

如果正交性至关重要, 则当 A 的列向量接近线性相关时, 建议使用 Householder-QR.

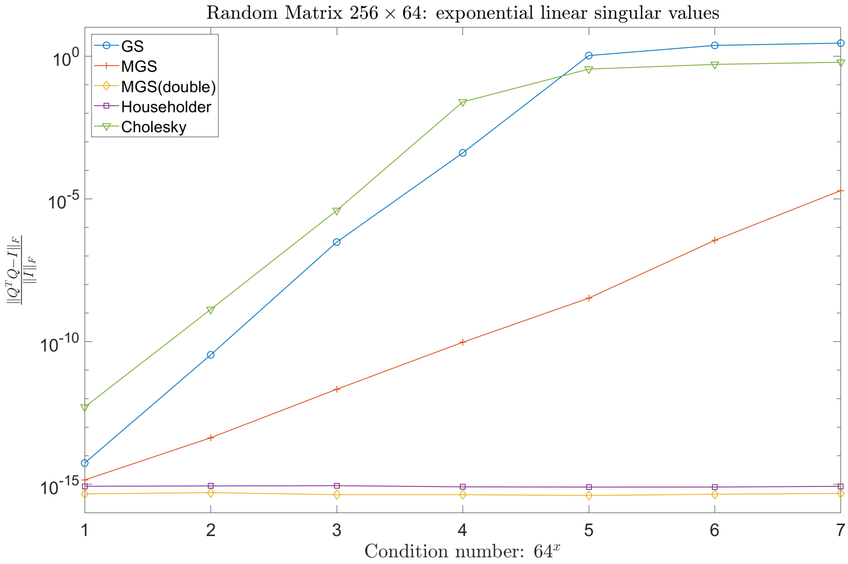
例 5.1 编写程序, 分别用 GS, MGS 和 Householder 变换计算 n 阶 Hilbert 矩阵 H 的 QR 分解, 并比较三种算法的稳定性, 即观察 $\|\tilde{Q}\tilde{R}-H\|_2$ 和 $\|\tilde{Q}^{\mathsf{T}}\tilde{Q}-I\|_2$ 的值, 其中 $\tilde{Q}$ 和 $\tilde{R}$ 是计算出来的 QR 分解矩阵因子. (LS_QR_stability.m)

	GS		MGS		Householder	
n	$\|\tilde{Q}\tilde{R}-H\|_2$	$\|\tilde{Q}^{\mathsf{T}}\tilde{Q}-I\|_2$	$\|\tilde{Q}\tilde{R}-H\|_2$	$\|\tilde{Q}^{\mathsf{T}}\tilde{Q}-I\|_2$	$\|\tilde{Q}\tilde{R}-H\|_2$	$\|\tilde{Q}^{\mathsf{T}}\tilde{Q}-I\|_2$
2	0.00e+00	1.81e-15	0.00e+00	1.81e-15	1.24e-16	2.36e-16
4	3.93e-17	3.45e-11	5.55e-17	2.98e-13	2.46e-16	7.08e-16
6	6.21e-17	5.33e-06	2.78e-17	5.81e-10	1.49e-16	9.49e-16
8	7.48e-17	1.03e+00	6.59e-17	4.38e-07	2.57e-16	1.44e-15
10	6.52e-17	3.00e+00	7.49e-17	4.16e-04	6.36e-16	1.00e-15
12	7.87e-17	4.00e+00	7.55e-17	8.52e-02	4.68e-16	9.52e-16
14	6.54e-17	7.00e+00	8.54e-17	9.96e-01	5.71e-16	8.35e-16



10^5
10^0
10^-5
10^-10
10^-15
2
4
6
8
10
12
14
16
18
20
GS
MGS
Householder

✍ 也可以通过重正交来提升 MGS 的稳定性, 即进行两次 MGS. (QR_3methods_256_64.m)



谢谢

THANK YOU