SI231 - Matrix Computations, Fall 2020-21

Solution of Homework Set #3

Prof. Yue Qiu and Prof. Ziping Zhao

I. Understanding projection

Problem 1. (5 points × 3) This problem is graded by Xinyue Zhang (**zhangxy11@**) & Yijia Chang (**changyj@**).

Suppose that $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a projector onto a subspace $\mathcal{U}$ along its orthogonal complement $\mathcal{U}^\perp$, then it is called the **orthogonal projector** onto $\mathcal{U}$.

1) Prove that an orthogonal projector must be singular if it is not an identity matrix.
2) What is the orthogonal projector onto $\mathcal{U}^\perp$ along the subspace $\mathcal{U}$?
3) Let $\mathcal{U}$ and $\mathcal{W}$ be two subspaces of a vector space $\mathcal{V}$, and denote $\mathbf{P}_\mathcal{U}$ and $\mathbf{P}_\mathcal{W}$ as the corresponding orthogonal projectors, respectively. Prove that $\mathbf{P}_\mathcal{U}\mathbf{P}_\mathcal{W} = 0$ if and only if $\mathcal{U} \perp \mathcal{W}$.

Solution.

1) Given an orthogonal projector $\mathbf{P} \in \mathbb{R}^{n\times n}$, we have learnt that $\mathbf{P}$ can be given by $\mathbf{P} = \mathbf{A}(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}^T$, where $\mathbf{A}$ is an $n \times m$ matrix and its column vectors form a basis of $\mathcal{R}(\mathbf{P})$. Since the column vectors of $\mathbf{A}$ are a basis of $\mathcal{R}(\mathbf{P})$, they must be linearly independent, which further implies that $m \le n$ and accordingly $\mathsf{rank}(\mathbf{A}) \le m$. Recall that $\mathbf{P} = \mathbf{A}(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}^T$, which implies $\mathsf{rank}(\mathbf{P}) \le \mathsf{rank}(\mathbf{A}) \le m \le n$.
   If $\mathsf{rank}(\mathbf{P}) < n$, we can directly know that $\mathbf{P}$ is singular. If $\mathsf{rank}(\mathbf{P}) = n$, we have that $\mathsf{rank}(\mathbf{A}) = n = m$ and accordingly $\mathbf{A}$ is non-singular. Hence, $\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{A}\mathbf{A}^{-1}(\mathbf{A}^T)^{-1}\mathbf{A}^T = \mathbf{I}$, which concludes this proof.
2) For any vector $\mathbf{x} \in \mathbb{R}^n$, since $(\mathbf{I} - \mathbf{P})\mathbf{x} = \mathbf{x} - \mathbf{P}\mathbf{x} \in \mathcal{S}^\perp$, we have that $\mathbf{I} - \mathbf{P}$ is the orthogonal projector onto subspace $\mathcal{S}^\perp$ along subspace $\mathcal{S}$.
3) For all $\mathbf{v} \in \mathcal{V}$, $\mathbf{P}_\mathcal{U}\mathbf{P}_\mathcal{W}\mathbf{v} = \mathbf{0}$ and $\mathbf{P}_\mathcal{W}\mathbf{v} \in \mathcal{W} \subseteq \mathcal{V}$. Then we can get $\mathbf{P}_\mathcal{W}\mathbf{v} \in \mathcal{N}(\mathbf{P}_\mathcal{U})$ for all $\mathbf{v} \in \mathcal{V}$, $\mathcal{R}(\mathbf{P}_\mathcal{W}) \subseteq \mathcal{N}(\mathbf{P}_\mathcal{U})$. Conversely, it still true. Hence,

$$\mathbf{P}_\mathcal{U}\mathbf{P}_\mathcal{W} = \mathbf{0} \iff \mathcal{R}(\mathbf{P}_\mathcal{W}) \subseteq \mathcal{N}(\mathbf{P}_\mathcal{U}) \iff \mathcal{W} \subseteq \mathcal{U}^\perp \iff \mathcal{W} \perp \mathcal{U}.$$

Grading policy: 2.5 points for the prove of sufficient condition and 2.5 points for the prove of necessary condition.

Remark: from $\mathbf{P}_\mathcal{U}\mathbf{P}_\mathcal{W} = 0$ we can only have $\mathcal{W} \subseteq \mathcal{U}^\perp$, that is $\mathcal{W}$ is not necessary to be the complement subspace of $\mathcal{U}$. So you can not use the assumption $\mathbf{P}_\mathcal{W} = I - \mathbf{P}_\mathcal{U}$ to prove this problem.

II. Least Square (LS) programming.

Problem 2. (8 points + 8 points + 4 points) This problem is graded by Chenguang Zhang (**zhangchg@**) & Bing Jiang (**jiangbing@**).

Write programs to solve the least square problem with specified methods, any programming language is suitable.

$$\mathbf{x} = \arg\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad f(\mathbf{x}) = ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a matrix representing the predefined data set with m data samples of n dimensions (m=1000, n=210), and $\mathbf{y} \in \mathbb{R}^m$ represents the labels. The data samples are provided in the "data.txt" file, and the labels are provided in the "label.txt" file, you are supposed to load the data before solving the problem.

1) Solve the LS with gradient decent method.

   The gradient descent method for solving problem updates $\mathbf{x}$ as

   $$\mathbf{x} = \mathbf{x} - \gamma \cdot \nabla_{\mathbf{x}} f(\mathbf{x}),$$

   where γ is the step size of the gradient decent methods. We suggest that you can set $\gamma = 1e - 5$.
2) Solve the LS by the method of normal equation with Cholesky decomposition and forward/backward substitution.
3) Compare two methods above.
   (a) Basing on the true running results from the program, count the number of "flops"*;
   (b) Compare gradient norm and loss $f(\mathbf{x})$ for results $\mathbf{x} = \mathbf{x}_{\mathbf{LS}}$ of above two algorithms.

Notation*: "flop": one flop means one floating point operation, i.e., one addition, subtraction, multiplication, or division of two floating-point numbers, in this problem each floating points operation $+, -, \times, \div, \sqrt{\cdot}$ counts as one "flop".

Hint for gradient decent programming:

1) **Step size selection**: to ensure the convergence of the method, γ is supposed to be selected properly (large step size may accelerate the convergence rate but also may lead to instability, A sufficiently small compensation always ensures that the algorithm converges).
2) **Terminal condition**: the gradient decent is an iteration algorithm that need a terminal condition. In this problem, the algorithm can stop when the gradient of the loss function $f(\mathbf{x})$ at current $\mathbf{x}$ is small enough.

Remarks:

- The solution of the two methods should be printed in files named "sol1.txt" and "sol2.txt" and submitted in gradescope. The format should be same as the input file (210 rows plain text, each rows is a dimension of the final solution).
- Make sure that your codes are executable and are consistent with your solutions.

Solution.

This part is graded using both auto-grader and the manual grade. To make sure your HW can be correctly graded next time, ".zip" format package rather than ".rar, .7zip" should be uploaded, solution of the results should be named according to requirements.

1) The norm of gradient should be a value near zero. The loss should be small enough and the results should be close to the reference results.
   - This an optimization problem with a convex target function that means one can get an optimal solution. It is supposed to find a stationary point(here is also a global optimal), i.e. gradient of which is close to zero!!
   - To conform the gradient is close to zero, you should make sure that the norm $(L_1, L_2$ or L_∞ etc.) is small enough, which is supposed as the terminal condition in your code. Heuristic stop condition such as termination when loss is small enough, termination after fixed times iterations is not allowed here. Some point is deducted with these conditions.
   - Error tolerance of the norm setting should not be too large!
2) Solve the normal equation $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}$ with required methods. The loss should be small enough and the results should be close to the reference results (1.e The L1 norm between our computed reference result and ground truth is 5.71, results that lower than 10 are OK.).
3) Grading system: following points or relative points should be discussed:
   - (1). "flops" of gradient decent method
   - (1). "flops" of gradient normal equation
   - (2). list some differences or similarities about two methods: i.e. compare gradient norm (optional) and compare loss.
   - (2). some analysis

   Example for counting "flops" :

   - normal equation: 91971390
     $\frac{n^3}{3} + (n^2 + n^2 - n) + (n^2 + n^2 - n) + 2mn^2 + 2mn + 2n^2 + n = 91971390$
   - In gradient decent algorithm, "flops" a iteration consumes is listed below:

operation	+	-	*	/	$\sqrt{\cdot}$
weight update		N	N		
loss vector	M*(N-1)	M	M*N		
loss	M-1		M		
gradient	N*(M-1)		N*M		
gradient norm	N-1		N		

     If the algorithm stops after K times iteration, total number of "flops" is $841838 \times K$. for example with K= 142

III. Understanding the QR Factorization

Problem 3 [Understanding the Gram-Schmidt algorithm.]. (5 points + 7 points + 6 points + 7 points) This problem is graded by Zhihang Xu (**xuzhh@**) & Zhicheng Wang (**wangzhch1@**).

1) Consider the subspace $\mathcal{S}$ spaned by $\{\mathbf{a}_1, \dots, \mathbf{a}_4\}$, where

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \quad \mathbf{a}_4 = \begin{bmatrix} 3 \\ 5 \\ 7 \\ 11 \end{bmatrix}.$$

Use the **classical** Gram-Schimidt algorithm (See Algorithm 1), find a set of orthonormal basis $\{\mathbf{q}_i\}$ for $\mathcal{S}$ by hand (derivation is expected). Do not use decimals in your answers, fraction and n-th roots of numbers are accepted. Verify the orthonormality of the found basis.

Algorithm 1: Classical Gram-Schmidt algorithm

```
Input  : A collection of linearly independent vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$.
Initilization: $\widetilde{\mathbf{q}}_1 = \mathbf{a}_1, \mathbf{q}_1 = \widetilde{\mathbf{q}}_1 / \|\widetilde{\mathbf{q}}_1\|_2$
for $i = 2, \dots, n$ do
    $\widetilde{\mathbf{q}}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} (\mathbf{q}_j^T \mathbf{a}_i) \mathbf{q}_j$
    $\mathbf{q}_i = \widetilde{\mathbf{q}}_i / \|\widetilde{\mathbf{q}}_i\|_2$
end
Output: $\mathbf{q}_1, \dots, \mathbf{q}_n$
```

2) Orthogonal projection of vector $\mathbf{a}$ onto a nonzero vector $\mathbf{b}$ is defined as

$$\text{proj}_{\mathbf{b}}(\mathbf{a}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{b}, \mathbf{b} \rangle} \mathbf{b},$$

where $\langle , \rangle$ denotes the inner product of vectors. And for subspace $\mathcal{M}$ with orthonormal basis $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$, the orthogonal projector onto subspace $\mathcal{M}$ is given by

$$\mathbf{P} = \mathbf{U}\mathbf{U}^T, \quad \mathbf{U} = [\mathbf{u}_1 | \cdots | \mathbf{u}_k].$$

In the context of **projection of vector** and **projection onto subspace** respectively, can you give another two understandings of the classical Gram-Schmidt algorithm?

3) Consider the subspace $\mathcal{S}$ spaned by $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$,

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ \epsilon \\ \epsilon \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} 1 \\ \epsilon \\ 0 \end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix} 1 \\ 0 \\ \epsilon \end{bmatrix},$$

where ϵ is a small real number such that $1 + k\epsilon^2 = 1$ ($k \in \mathbb{N}^+$). First complete the pseudo algorithm in Algorithm 2. Then use the **classical** Gram-Schimidt algorithm and the **modified** Gram-Schimidt algorithm respectively, find two sets of basis for $\mathcal{S}$ by hand (derivation is expected). Are the two sets of basis the same? If not, which one is the desired orthonormal basis? Report what you have found.

```
Algorithm 2: Modified Gram-Schmidt algorithm
Input  : A collection of linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$.
Initilization:
Complete your algorithm here...
Output: $\mathbf{q}_1, \ldots, \mathbf{q}_n$
```

4) **Programming part:** In this part, you are required to code both the **classical Gram-Schmidt** and **the modified Gram-Schmidt** algorithms. For $\epsilon = 1\mathrm{e}{-4}$ and $\epsilon = 1\mathrm{e}{-9}$ in sub-problem 3), give the outputs of two algorithms and calculate $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_{\mathrm{F}}$, where $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]$.

Remarks:

- Coding languages are not restricted, but do not use any bulit-in function such as `qr`.
- When handing in your homework in gradescope, package all your codes into your_student_id+hw3_code.zip and upload. In the package, you also need to include a file named README.txt/md to clearly identify the function of each file.
- Make sure that your codes can run and are consistent with your solutions.

Solution.

1) Note that $\{\mathbf{a}_1, \ldots, \mathbf{a}_4\}$ are not linearly independent vectors since $\mathbf{a}_3 = 2\mathbf{a}_2 - \mathbf{a}_1$. Delete $\mathbf{a}_3$ such that $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}$ is a basis of $\mathcal{S}$. For the simplicity of notations, we rewrite $\mathbf{b}_1 = \mathbf{a}_1, \mathbf{b}_2 = \mathbf{a}_2$ and $\mathbf{b}_3 = \mathbf{a}_4$. Then, following the steps in Algorithm 1, we have,

   - For $i = 1$, $\widetilde{\mathbf{q}}_1 = \mathbf{b}_1$ and $\|\widetilde{\mathbf{q}}_1\|_2 = \sqrt{30}$, therefore $\mathbf{q}_1 = \frac{1}{\sqrt{30}} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$.
   - For $i = 2$, $\widetilde{\mathbf{q}}_2 = \mathbf{b}_2 - (\mathbf{q}_1^T\mathbf{b}_2)\mathbf{q}_1 = \frac{1}{3} \begin{bmatrix} 2 \\ 1 \\ 0 \\ -1 \end{bmatrix}$, therefore $\mathbf{q}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 2 \\ 1 \\ 0 \\ -1 \end{bmatrix}$.
   - For $i = 3$, $\widetilde{\mathbf{q}}_3 = \mathbf{b}_3 - (\mathbf{q}_1^T\mathbf{b}_3)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{b}_3)\mathbf{q}_2 = \frac{1}{5} \begin{bmatrix} 2 \\ -1 \\ -4 \\ 3 \end{bmatrix}$, therefore $\mathbf{q}_3 = \frac{1}{\sqrt{30}} \begin{bmatrix} 2 \\ -1 \\ -4 \\ 3 \end{bmatrix}$.

Therefore,

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{6}} & \frac{2}{\sqrt{30}} \\ \frac{2}{\sqrt{30}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{30}} \\ \frac{3}{\sqrt{30}} & 0 & -\frac{4}{\sqrt{30}} \\ \frac{4}{\sqrt{30}} & -\frac{1}{\sqrt{6}} & \frac{3}{\sqrt{30}} \end{bmatrix}, \text{ (4 points)}$$

we can verify the orthnormality of the basis via

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{I}\,. \text{ (1 points)}$$

2) In the context of **projection of vectors** (3.5 points), we can understand the classical Gram-Schmidt algorithm as follows, for a set of linearly independent vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$,

- First, normalize $\mathbf{a}_1$ to obtain the first basis $\mathbf{q}_1$.
- Then, to obtain a set of orthonormal basis, we can take $\widetilde{\mathbf{q}}_2$ to be $\mathbf{a}_2$ minus its projection of $\mathbf{q}_1$, in this way, the obtained $\widetilde{\mathbf{q}}_2$ is orthogonal with $\mathbf{q}_1$, and then normalize $\widetilde{\mathbf{q}}_2$ to obtain $\mathbf{q}_2$. So far, we have $\{\mathbf{q}_1, \mathbf{q}_2\}$ which are a set of orthonormal basis for the subspace spaned by $\{\mathbf{a}_1, \mathbf{a}_2\}$.
- Repeat the previous step: each time we take $\widetilde{\mathbf{q}}_i$ to be the $\mathbf{a}_i$ minus the projection its projection of $\mathbf{q}_1, \ldots, \mathbf{q}_{i-1}$. By doing so, at the end, we can obtain a set of orthonormal basis for the subspace spaned by $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$.

Rewrite Algorithm 1 in the context of **projection of vectors**:

$$\begin{aligned} &\widetilde{\mathbf{q}}_1 = \mathbf{a}_1\,, \quad \mathbf{q}_1 = \widetilde{\mathbf{q}}_1/\|\widetilde{\mathbf{q}}_1\|_2\,, \\ &\widetilde{\mathbf{q}}_2 = \mathbf{a}_2 - \mathrm{proj}_{\mathbf{q}_1}(\mathbf{a}_2)\,, \quad \mathbf{q}_2 = \widetilde{\mathbf{q}}_2/\|\widetilde{\mathbf{q}}_2\|_2\,, \\ &\cdots \\ &\widetilde{\mathbf{q}}_n = \mathbf{a}_n - \sum_{i=1}^{n-1} \mathrm{proj}_{\mathbf{q}_i}(\mathbf{a}_n)\,, \quad \mathbf{q}_n = \widetilde{\mathbf{q}}_n/\|\widetilde{\mathbf{q}}_n\|_2\,. \end{aligned}$$

In the context of **projection onto subspace** (3.5 points), we can understand the Gram-Schmidt algorithm as follows, for a set of linearly independent vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$,

- First, normalize $\mathbf{a}_1$ to obtain the first basis $\mathbf{q}_1$.
- Then, to obtain a set of orthonormal basis, we can take $\widetilde{\mathbf{q}}_2$ to be $\mathbf{a}_2$ minus its projection onto span$(\mathbf{q}_1)$, in this way, the obtained $\widetilde{\mathbf{q}}_2$ is orthogonal with $\mathbf{q}_1$, and then normalize $\widetilde{\mathbf{q}}_2$ to obtain $\mathbf{q}_2$. So far, we have $\{\mathbf{q}_1, \mathbf{q}_2\}$ which are a set of orthonormal basis for the subspace spaned by $\{\mathbf{a}_1, \mathbf{a}_2\}$.
- Repeat the previous step: each time we take $\widetilde{\mathbf{q}}_i$ to be the $\mathbf{a}_i$ minus the projection onto the span$\{\mathbf{q}_1, \ldots, \mathbf{q}_{i-1}\}$. By doing so, at the end, we can obtain a set of orthonormal basis for the subspace spaned by $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$.

Rewrite Algorithm 1 in the context of **projection onto subspaces**:

$$
\begin{aligned}
&\widetilde{\mathbf{q}}_1 = \mathbf{a}_1\,, \quad \mathbf{q}_1 = \widetilde{\mathbf{q}}_1/\|\widetilde{\mathbf{q}}_1\|_2\,,\\
&\widetilde{\mathbf{q}}_2 = (\mathbf{I} - \mathbf{Q}_1\mathbf{Q}_1^T)\mathbf{a}_1\,, \quad \mathbf{q}_2 = \widetilde{\mathbf{q}}_2/\|\widetilde{\mathbf{q}}_2\|_2\,, \mathbf{Q}_1 = [\mathbf{q}_1]\\
&\cdots\\
&\widetilde{\mathbf{q}}_n = (\mathbf{I} - \mathbf{Q}_n\mathbf{Q}_n^T)\mathbf{a}_n\,, \quad \mathbf{q}_n = \widetilde{\mathbf{q}}_n/\|\widetilde{\mathbf{q}}_n\|_2\,, \mathbf{Q}_n = [\mathbf{q}_1|\mathbf{q}_2|\cdots|\mathbf{q}_{n-1}]\,.
\end{aligned}
$$

3) The complete modified gram-schimidt algorithm is shown in Algorithm 3. For linearly independent vectors $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$, following the steps in Algorithm 1, we have,

- For $i = 1$, $\widetilde{\mathbf{q}}_1 = \mathbf{a}_1$ and $\|\widetilde{\mathbf{q}}_1\|_2 = \sqrt{1 + 2\epsilon^2} = 1$, therefore $\mathbf{q}_1 = \mathbf{a}_1 = \begin{bmatrix} 1 \\ \epsilon \\ \epsilon \end{bmatrix}$.
- For $i = 2$, $\widetilde{\mathbf{q}}_2 = \mathbf{a}_2 - (\mathbf{q}_1^T\mathbf{a}_2)\mathbf{q}_1 = \begin{bmatrix} 0 \\ 0 \\ -\epsilon \end{bmatrix}$, therefore $\mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$.
- For $i = 3$, $\widetilde{\mathbf{q}}_3 = \mathbf{a}_3 - (\mathbf{q}_1^T\mathbf{a}_3)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{a}_3)\mathbf{q}_2 = \begin{bmatrix} 0 \\ -\epsilon \\ -\epsilon \end{bmatrix}$, therefore $\mathbf{q}_3 = \begin{bmatrix} 0 \\ -1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$ (2 points).

Following the steps in Algorithm 3, we have,

- Initilization: $\widetilde{\mathbf{q}}_1 = \mathbf{a}_1, \widetilde{\mathbf{q}}_2 = \mathbf{a}_2, \widetilde{\mathbf{q}}_3 = \mathbf{a}_3$.
- For $i = 1$, $\mathbf{q}_1 = \widetilde{\mathbf{q}}_1/\|\widetilde{\mathbf{q}}_1\|_2 = \begin{bmatrix} 1 \\ \epsilon \\ \epsilon \end{bmatrix}$. For $k = 2$, $\widetilde{\mathbf{q}}_2 = \widetilde{\mathbf{q}}_2 - (\mathbf{q}_1^T\widetilde{\mathbf{q}}_2)\mathbf{q}_1 = \begin{bmatrix} 0 \\ 0 \\ \epsilon \end{bmatrix}$. For $k = 3$, $\widetilde{\mathbf{q}}_3 = \widetilde{\mathbf{q}}_3 - (\mathbf{q}_1^T\widetilde{\mathbf{q}}_3)\mathbf{q}_1 = \begin{bmatrix} 0 \\ -\epsilon \\ 0 \end{bmatrix}$.
- For $i = 2$, $\mathbf{q}_2 = \widetilde{\mathbf{q}}_2/\|\widetilde{\mathbf{q}}_2\|_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$. For $k = 3$, $\widetilde{\mathbf{q}}_3 = \widetilde{\mathbf{q}}_3 - (\mathbf{q}_2^T\widetilde{\mathbf{q}}_3)\mathbf{q}_2 = \begin{bmatrix} 0 \\ -\epsilon \\ 0 \end{bmatrix}$.
- For $i = 3$, $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3/\|\widetilde{\mathbf{q}}_3\|_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$ (2 points).

Therefore, to sum up, classical Gram-Schmidt returns

$$
\mathbf{q}_1 = \begin{bmatrix} 1 \\ \epsilon \\ \epsilon \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad \mathbf{q}_3 = \begin{bmatrix} 0 \\ -1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix},
$$

which is not a set of orthonormal basis since $\mathbf{q}_2$ and $\mathbf{q}_3$ are not orthogonal. And the modified Gram-Schmidt algorithm returns,

$$\mathbf{q}_1 = \begin{bmatrix} 1 \\ \epsilon \\ \epsilon \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad \mathbf{q}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix},$$

which is expected (1 point).

4) **Reference codes can be found in Appendix part (VI-0a).** For $\epsilon = 1\text{e}-4$, $\mathbf{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ (CGS and MGS), $\mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ (CGS and MGS), $\mathbf{q}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$ (CGS and MGS). $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_{\text{F}}$ <1e-7(CGS), $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_{\text{F}}$ <1e-10(MGS).

For $\epsilon = 1\text{e}-9$, $\mathbf{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ (CGS), $\mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ (CGS), $\mathbf{q}_3 = \frac{\sqrt{2}}{2}\begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$ (CGS), $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_{\text{F}}$=1(CGS), $\mathbf{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ (MGS), $\mathbf{q}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ (MGS), $\mathbf{q}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$ (MGS), $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_{\text{F}}$ <1e-7(MGS).

Algorithm 3: Modified Gram-Schmidt algorithm (1 point)

```
Input : A collection of linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$.
Initilization: $\widetilde{\mathbf{q}}_i = \mathbf{a}_i$, for $i = 1, \ldots, n$.
for $i = 1, \ldots, n$ do
    $\mathbf{q}_i = \widetilde{\mathbf{q}}_i / \|\widetilde{\mathbf{q}}_i\|_2$.
    for $k = i + 1, \ldots, n$ do
        $\widetilde{\mathbf{q}}_k = \widetilde{\mathbf{q}}_k - (\mathbf{q}_i^T \widetilde{\mathbf{q}}_k)\mathbf{q}_i$.
    end
end
Output: $\mathbf{q}_1, \ldots, \mathbf{q}_n$
```

Remarks: This problem contains 4 sub-problems.

1) In sub-problem 1), you are required to find a set of orthonormal basis $\{\mathbf{q_i}\}$ and also verify the orthonormality of the found basis. The found basis takes 4 points and the verification of orthonormality takes 1 point. And the verification of orthogonality takes 0.5 point, and the verification of normality takes 0.5 point.
2) In sub-problem 2), you are required to give two understandings of Gram-Schmidt algorithm in two different contexts. Each understanding takes 3.5 points, and the verbal description or the mathematical representation are both accepted.

3) In sub-problem 3), you are required to give two results of classical Gram-Schmidt algorithm and modified Gram-Schmidt algorithm, conclusion of results and pseudo of modified Gram-Schmidt algorithm. Each result takes 2 points, conclusion takes 1 point and pseudo takes 1 point.
4) In sub-problem 4), you are required to give two results of classical Gram-Schmidt algorithm and modified Gram-Schmidt algorithm, including orthogonal basis and $\|\mathbf{Q}^T\mathbf{Q}-\mathbf{I}\|_{\mathrm{F}}$. Each orthogonal basis takes 1 point, each result of $\|\mathbf{Q}^T\mathbf{Q}-\mathbf{I}\|_{\mathrm{F}}$ takes 1 point, CGS code takes 1 point and MGS code takes 2 points.

Grading policy:

1) In sub-problem 1), if you give the correct basis $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$ you will get 4 points, and in this case if you also give the verification of orthonormality, you will get 5 points.
2) In sub-problem 1), if you give the wrong basis $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$ for some mistakes in computation (In this case, we specifically mean that you give the right deviation and also notice that $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ is linearly dependent but there exist some computation mistakes), you will get 3 points due to deviation process, and in this case the verification of orthonormality takes no points.
3) In sub-problem 1), if you give a set of basis contains zero vector and you give the derivation of $\mathbf{q}_1$ and $\mathbf{q}_2$, you will get 1 point for **partially** right derivation. But if you directly give a set of basis containning zero vector without any derivation, you will get 0 point. Because a set of basis cannot contain zero vector, and we strongly suggest a review of previous basic concepts.
4) In sub-problem 1), after give the correct basis $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$, you are also required to verify the orthonormality of the found basis, then you should check $\mathbf{q}_i^T\mathbf{q}_i = 1$, for $i = 1, 2, 3$ (normality) and $\mathbf{q}_i^T\mathbf{q}_j = 1$, for $i \neq j$ (orthogonality) at the same time. Write $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]$ and directly show that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ is also accepted. Many students lose 0.5 point because you simply show that $\mathbf{q}_i^T\mathbf{q}_j = 0 (i \neq j)$, and note that this **cannot** imply the orthonormality but only orthogonality. You may argue that the normality explains by itself, because normalization part is done during the algorithm, but since the problem clearly requires *verify*, and the key of verifying lies in re-checking something we may already knew, so you still need to state your verification.
5) In sub-problem 2), you are required to give two understandings in two different contexts, basically, if your answer contains keywords/key-sentences like $\widetilde{\mathbf{q}}_i$ *is obtained via* a_i *minus the projection with respect to the previous basis* and $\widetilde{\mathbf{q}}_i$ *is obtained via* a_i *minus the projection onto the subspace spaned by the previous basis*, you will get full points. Some minor mistakes in statement and completeness are pointed out in your gradescope system, but basically you will not lose points if there is no major misunderstanding in the algorithm.
6) In sub-problem 3), if you give result of classical or modified Gram-Schmidt algorithm which is not normalized, you will loss 1 point.
7) In sub-problem 3), if you give result of Gram-Schmidt algorithm which not use $1 + k\epsilon^2 = 1$, you will loss 1 point.
8) In sub-problem 4), if you give result of Frobenius norm using norm(A) rather than norm(A,'fro') in Matlab, you will loss 1 point.

IV. SOLVING LS VIA QR FACTORIZATION AND NORMAL EQUATION

Problem 4 [Understanding the influence of the condition number to the solution.]. (4 points + 5 points + 4 points + 4 points +3 points) This problem is graded by Lin Zhu (**zhulin@**).

Consider such two LS problems:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \tag{1}$$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-(\mathbf{b}+\delta\mathbf{b})\|_2^2 \tag{2}$$

with $\mathbf{A}\in\mathbb{R}^{m\times n}$. For $\mathbf{b}=\begin{bmatrix}1 & 3/2 & 3 & 6\end{bmatrix}^T$ and $\delta\mathbf{b}=\begin{bmatrix}1/10 & 0 & 0 & 0\end{bmatrix}^T$,

1) Give the QR decomposition of $\mathbf{A}$ with $\mathbf{Q}$ being square, and then compute the solution to problem (1) via QR decomposition when

$$\mathbf{A}=\begin{bmatrix}1 & 2 & 3\\ 2 & 3 & 5\\ 3 & 4 & 7\\ 4 & 5 & 11\end{bmatrix}.$$

2) For a full-rank matrix $\mathbf{A}$, consider the equation $\mathbf{A}\mathbf{x}=\mathbf{b}$, after adding some noise $\delta\mathbf{b}$ to $\mathbf{b}$, we have $\mathbf{A}(\mathbf{x}+\delta\mathbf{x})=\mathbf{b}+\delta\mathbf{b}$ve, and then prove

$$\frac{1}{\|\mathbf{A}\|\|\mathbf{A}^\dagger\|}\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}\leq\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}\leq\|\mathbf{A}\|\|\mathbf{A}^\dagger\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|},$$

and give it a plain interpretation.

3) Computing the solutions to the two LS problems via the normal equation $\mathbf{A}^T\mathbf{A}\mathbf{x}_{LS}=\mathbf{A}^T\mathbf{b}$ when

$$\mathbf{A}=\begin{bmatrix}1 & 2 & 2\\ 2 & 2 & 2\\ 3 & 3 & 3\\ 1 & 1 & 0\end{bmatrix}.$$

4) Computing the solutions to the two LS problems via the normal equation $\mathbf{A}^T\mathbf{A}\mathbf{x}_{LS}=\mathbf{A}^T\mathbf{b}$ when

$$\mathbf{A}=\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 4\\ 1 & 3 & 9\\ 1 & 4 & 16\end{bmatrix}.$$

5) Compare the 2-norm condition number $\|\mathbf{A}\|\|\mathbf{A}^\dagger\|$ for $\mathbf{A}$ in 3) and 4) and the influence on the solution to problem (1) resulted by the additional noise $\delta\mathbf{b}$.

   Hint: Show the influence on the solution by $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}$.

Remarks: You can use MATLAB for some matrix computations (deviation is expected) in 3), 4), 5). Do not use decimals in your answers, fraction and n-th roots of numbers are accepted.

Solution.

1) Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$, where $\mathbf{a}_1 = [1,2,3,4]^T$, $\mathbf{a}_2 = [2,3,4,5]^T$ and $\mathbf{a}_3 = [3,5,7,11]^T$. If $\mathbf{A}$ has such a decomposition that $\mathbf{A} = \mathbf{QR}$,

- for $i = 1$, $\widetilde{\mathbf{q}}_1 = \mathbf{a}_1$ and $\|\widetilde{\mathbf{q}}_1\|_2 = \sqrt{30}$, therefore $\mathbf{q}_1 = \frac{1}{\sqrt{30}}\begin{bmatrix}1\\2\\3\\4\end{bmatrix}$ and $\mathbf{R}_{11} = \|\widetilde{\mathbf{q}}_1\|_2 = \sqrt{30}$.
- for $i = 2$, $\widetilde{\mathbf{q}}_2 = \mathbf{a}_2 - (\mathbf{q}_1^T\mathbf{b}_2)\mathbf{q}_1 = \frac{1}{3}\begin{bmatrix}2\\1\\0\\-1\end{bmatrix}$, therefore $\mathbf{q}_2 = \frac{1}{\sqrt{6}}\begin{bmatrix}2\\1\\0\\-1\end{bmatrix}$, $\mathbf{R}_{22} = \|\widetilde{\mathbf{q}}_2\|_2 = \frac{2}{\sqrt{6}}$ and $\mathbf{R}_{12} = \mathbf{q}_2^T\mathbf{a}_1 = \frac{40}{\sqrt{30}}$.
- for $i = 3$, $\widetilde{\mathbf{q}}_3 = \mathbf{a}_3 - (\mathbf{q}_1^T\mathbf{b}_3)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{b}_3)\mathbf{q}_2 = \frac{1}{5}\begin{bmatrix}2\\-1\\-4\\3\end{bmatrix}$, therefore $\mathbf{q}_3 = \frac{1}{\sqrt{30}}\begin{bmatrix}2\\-1\\-4\\3\end{bmatrix}$, $\mathbf{R}_{33} = \|\widetilde{\mathbf{q}}_3\|_2 = \frac{6}{\sqrt{30}}$, $\mathbf{R}_{32} = \mathbf{q}_3^T\mathbf{a}_2 = 0$ and $\mathbf{R}_{31} = \mathbf{q}_3^T\mathbf{a}_1 = \frac{78}{\sqrt{30}}$. (2 points for the procedures)

Therefore, $\mathbf{A} = \mathbf{QR}$ with (1 point for the Q and R matrices)

$$\mathbf{Q} = \begin{bmatrix}\frac{1}{\sqrt{30}} & \frac{2}{\sqrt{6}} & \frac{2}{\sqrt{30}}\\ \frac{2}{\sqrt{30}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{30}}\\ \frac{3}{\sqrt{30}} & 0 & -\frac{4}{\sqrt{30}}\\ \frac{4}{\sqrt{30}} & -\frac{1}{\sqrt{6}} & \frac{3}{\sqrt{30}}\end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix}\sqrt{30} & \frac{40}{\sqrt{30}} & \frac{78}{\sqrt{30}}\\ 0 & \frac{2}{\sqrt{6}} & 0\\ 0 & 0 & \frac{6}{\sqrt{30}}\end{bmatrix}.$$

We can get $\mathbf{QR}\cdot\mathbf{x}_{\text{LS}} = \mathbf{b}$, i.e., $\mathbf{R}\cdot\mathbf{x}_{\text{LS}} = \mathbf{Q}^T\mathbf{b}$ (1 point for solving the equation)

$$\begin{bmatrix}\sqrt{30} & \frac{40}{\sqrt{30}} & \frac{78}{\sqrt{30}}\\ 0 & \frac{2}{\sqrt{6}} & 0\\ 0 & 0 & \frac{6}{\sqrt{30}}\end{bmatrix}\cdot\mathbf{x}_{\text{LS}} = \begin{bmatrix}\frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{3}{\sqrt{30}} & \frac{4}{\sqrt{30}}\\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{6}}\\ \frac{2}{\sqrt{30}} & -\frac{1}{\sqrt{30}} & -\frac{4}{\sqrt{30}} & \frac{3}{\sqrt{30}}\end{bmatrix}\cdot\begin{bmatrix}1\\3/2\\3\\6\end{bmatrix} = \begin{bmatrix}\frac{37}{\sqrt{30}}\\ -\frac{5}{2\sqrt{6}}\\ \frac{13}{2\sqrt{30}}\end{bmatrix},$$

then $\mathbf{x}_{\text{LS}} = [1/12, -5/4, 13/12]^T$.

2) We have $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{A}\delta\mathbf{x} = \delta\mathbf{b}$, then $\|\mathbf{b}\| \le \|\mathbf{A}\|\|\mathbf{x}\|$ and $\|\delta\mathbf{b}\| \le \|\mathbf{A}\|\|\delta\mathbf{x}\|$.

① Since $\mathbf{A}^\dagger\mathbf{b}$ is the LS solution to (1), i.e., $x = \mathbf{A}^\dagger\mathbf{b}$ so we get $\|\mathbf{x}\| \le \|\mathbf{A}^\dagger\|\|\mathbf{b}\|$. With $\|\delta\mathbf{b}\| \le \|\mathbf{A}\|\|\delta\mathbf{x}\|$, we then compute that

$$\|\mathbf{x}\|\cdot\|\delta\mathbf{b}\| \le \|\mathbf{A}^\dagger\|\|\mathbf{b}\|\|\mathbf{A}\|\|\delta\mathbf{x}\|, \quad \text{i.e., } \frac{1}{\|\mathbf{A}\|\|\mathbf{A}^\dagger\|}\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \le \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}. \text{ (2 points)}$$

② Since $\|\delta\mathbf{x}\| \le \|\mathbf{A}^\dagger\|\|\delta\mathbf{b}\|$, with $\|\mathbf{b}\| \le \|\mathbf{A}\|\|\mathbf{x}\|$, we can get

$$\|\mathbf{b}\|\cdot\|\delta\mathbf{x}\| \le \|\mathbf{A}^\dagger\|\|\mathbf{x}\|\|\mathbf{A}\|\|\delta\mathbf{b}\|, \quad \text{i.e., } \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \|\mathbf{A}\|\|\mathbf{A}^\dagger\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}. \text{ (2 points)}$$

Therefore, the claim is proved. A plain interpretation: in the LS problem (1), the bigger the $\|\mathbf{A}\|\|\mathbf{A}^\dagger\|$ is, the more disturbance resulted from the noise $\delta\mathbf{b}$ to the solution may be. (it stands if it makes sense) (1 point)

3) By the normal equation, $\mathbf{x}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ (the derivation of $\mathbf{x}_{\text{LS}}$), i.e.,

$$\mathbf{x}_{\text{LS}} = \left(\begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 2 & 3 & 1 \\ 2 & 2 & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 1 & 1 & 0 \end{bmatrix}\right)^{-1} \cdot \begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 2 & 3 & 1 \\ 2 & 2 & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} 17/13 & -17/13 & 2/13 \\ -17/13 & 30/13 & -15/13 \\ 2/13 & -15/13 & 14/13 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 2 & 3 & 1 \\ 2 & 2 & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 4/13 & 6/13 & 0 \\ 1 & -4/13 & -6/13 & 1 \\ 0 & 2/13 & 3/13 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 11/13 \\ 67/13 \\ -66/13 \end{bmatrix}. \text{ (2 points)}$$

In the similar way, we can get $\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b} + \delta\mathbf{b})$ (the derivation of $\hat{\mathbf{x}}_{\text{LS}}$) and, with the help of the computing results above,

$$\hat{\mathbf{x}}_{\text{LS}} = \begin{bmatrix} -1 & 4/13 & 6/13 & 0 \\ 1 & -4/13 & -6/13 & 1 \\ 0 & 2/13 & 3/13 & -1 \end{bmatrix} \cdot \begin{bmatrix} 11/10 \\ 3/2 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 97/130 \\ 683/130 \\ -66/13 \end{bmatrix}. \text{ (2 points)}$$

4) By the normal equation, $\mathbf{x}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ (the derivation of $\mathbf{x}_{\text{LS}}$), i.e.,

$$\mathbf{x}_{\text{LS}} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}\right)^{-1} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} 31/4 & -27/4 & 5/4 \\ -27/4 & 129/20 & -5/4 \\ 5/4 & -5/4 & 1/4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} 9/4 & -3/4 & -5/4 & 3/4 \\ -31/20 & 23/20 & 27/20 & -19/20 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3/2 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 15/8 \\ -59/40 \\ 5/8 \end{bmatrix}. \text{ (2 points)}$$

In the similar way, we can get $\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b}+\delta\mathbf{b})$ (the derivation of $\hat{\mathbf{x}}_{\text{LS}}$) and, with the help of the computing results above, $\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b}+\delta\mathbf{b})$,

$$\hat{\mathbf{x}}_{LS} = \begin{bmatrix} 9/4 & -3/4 & -5/4 & 3/4 \\ -31/20 & 23/20 & 27/20 & -19/20 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{bmatrix} \cdot \begin{bmatrix} 11/10 \\ 3/2 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 21/10 \\ -163/100 \\ 13/20 \end{bmatrix}. \text{ (2 points)}$$

5) In 3) we have $r_1 = \frac{\|\hat{\mathbf{x}}_{LS}-\mathbf{x}_{LS}\|_2}{\|\mathbf{x}_{LS}\|_2} = \frac{\sqrt{2}}{10} \cdot \frac{13}{\sqrt{8966}} \approx 0.0194$ and $k_1 = \|\mathbf{A}\|_2\|\mathbf{A}^\dagger\|_2 \approx 13.3254$ (by MATLAB) (1 point); in 4) we have $r_2 = \frac{\|\hat{\mathbf{x}}_{LS}-\mathbf{x}_{LS}\|_2}{\|\mathbf{x}_{LS}\|_2} = \frac{\sqrt{3011}}{200} \cdot \frac{40}{\sqrt{9731}} \approx 0.1113$ and $k_2 = \|\mathbf{A}\|_2\|\mathbf{A}^\dagger\|_2 \approx 73.6944$ (by MATLAB) (1 point).

As $r_1 < r_2$ and $k_1 < k_2$, $\delta\mathbf{b}$ has a greater influence on solution in 4) than that of 3) and the 2-norm condition number of the coefficient matrix in 4) is also bigger than that of 3); (**or:** When b is changed, solution of 4) is more stable.) (**or:** the bigger the $\|\mathbf{A}\|\|\mathbf{A}^\dagger\|$ is, the greater disturbance to the solution to LS problem (1) resulted from the small additional noise $\delta\mathbf{b}$ is.) (it stands if it makes sense) (1 point)

Grading policy:

1) In sub-problem 1), the procedures of **QR** decomposition are necessary, if you do not get the 2 points but you have already gave the right procedures in Problem 3, you can argue on GS in time. As for the **QR** decomposition results, the **Q** can also be written with the additional 4-th column $[\frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}, 0]^T$ and **R** with the additional 4-th row $[0, 0, 0]$.

2) In sub-problem 2), for a full-rank matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, since the problem we have is an over-determined system, i.e., $m > n$, so actually it is full-column-rank with $\text{rank}(\mathbf{A}) = n$. For the under-determined system, $m < n$, you may have infinity solutions, however, LS does not suit such a system.

   Some students use $\mathbf{A}^{-1} = \mathbf{A}^\dagger$, it is not reasonable, because the evidence only stands when matrix $\mathbf{A}$ is invertible. Some use $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}$, it is wrong here, since sometimes $\mathbf{A}\mathbf{A}^\dagger \neq \mathbf{I}$ even $\mathbf{A}$ has full rank with $\mathbf{A}^\dagger\mathbf{A} = \mathbf{I}$. Here comes such an example $\mathbf{A} = [1, 0]^T$.

   The plain interpretation does not have a standard edition, the key point is how the $\|\mathbf{A}\|\|\mathbf{A}^\dagger\|$ influence the relative error of the solution, which can be understood as the relationship between the coefficient matrix $\mathbf{A}$ and the stability of the solution to the LS problem (1).

3) In sub-problem 3) and 4), you are expected to give the derivations but not required. If you give wrong computing results without the right derivation, you will lose all the points. But if you show the right derivation and computing procedures, you can get some partial grades even your final computations are wrong.

4) In sub-problem 5), after computing, you should also give your final compare and propose your conclusion about the influence according to the compare. Since the computing result of $\|\mathbf{A}\|_2\|\mathbf{A}^\dagger\|_2$ is an approximate value by MATLAB, it is acceptable to use decimals here.

   Some compare the two $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}$ by $\frac{1}{\|\mathbf{A}\|\|\mathbf{A}^\dagger\|}\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \le \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \|\mathbf{A}\|\|\mathbf{A}^\dagger\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$ with both the left and the right sides being computing results, it is not reasonable, because here both the two $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}$ have been specific values.

V. Underdetermined System

Problem 5 [Solving Underdetermined System by QR]. (15 points) This problem is graded by Sihang Xu (**xush**@).

Consider the following underdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $m < n$. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 2 & 0 \\ 0 & -2 & 2 & 1 \\ 2 & 5 & 6 & 1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

1) Use Householder reflection to give the full QR decomposition of tall $\mathbf{A}^T$, i.e., $\mathbf{A}^T = \mathbf{Q}\mathbf{R}$ with $\mathbf{Q}$ being a square matrix with orthonormal columns.
2) Give one possible solution via QR decomposition of $\mathbf{A}^T$, write down your solution using $\mathbf{b}$.

Solution.

1) Following the steps in Househoulder QR, rewrite

$$\mathbf{A}^{(0)} = \mathbf{A}^T = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -2 & 5 \\ 2 & 2 & 6 \\ 0 & 1 & 1 \end{bmatrix} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3].$$

- First, perform Householder reflection to the first column of $\mathbf{A}^{(0)}$,

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_1 = \mathbf{a}_1 - \|\mathbf{a}_1\|_2 \mathbf{e}_1 = \begin{bmatrix} -2 \\ 2 \\ 2 \\ 0 \end{bmatrix},$$

$$\mathbf{H}_1 = \mathbf{I} - \frac{2}{\|\mathbf{v}_1\|_2^2}\mathbf{v}_1\mathbf{v}_1^T = \frac{1}{3}\begin{bmatrix} 1 & 2 & 2 & 0 \\ 2 & 1 & -2 & 0 \\ 2 & -2 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}, \quad \mathbf{A}^{(1)} = \mathbf{H}_1\mathbf{A} = \begin{bmatrix} 3 & 0 & 8 \\ 0 & -2 & -1 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

.

- Next, perform Householder reflection to $\mathbf{A}^{(1)}_{2:4,2}$ (the block we colored red).

$$\widetilde{\mathbf{a}}_2 = \begin{bmatrix} -2 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \widetilde{\mathbf{a}}_2 - \|\widetilde{\mathbf{a}}_2\|\mathbf{e}_2 = \begin{bmatrix} -5 \\ 2 \\ 1 \end{bmatrix}$$

$$\widetilde{\mathbf{H}}_2 = \mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}_2\mathbf{v}_2^T = \begin{bmatrix} -\frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{11}{15} & -\frac{2}{15} \\ \frac{1}{3} & -\frac{2}{15} & \frac{14}{15} \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{11}{15} & -\frac{2}{15} \\ 0 & \frac{1}{3} & -\frac{2}{15} & \frac{14}{15} \end{bmatrix},$$

$$\mathbf{A}^{(2)} = \mathbf{H}_2\mathbf{A}^{(1)} = \begin{bmatrix} 3 & 0 & 8 \\ 0 & 3 & 1 \\ 0 & 0 & -\frac{4}{5} \\ 0 & 0 & \frac{3}{5} \end{bmatrix}.$$

- Perform Householder reflection to $\mathbf{A}^{(2)}_{3:4,3}$ (the block we colored red):

$$\widetilde{\mathbf{a}}_3 = \begin{bmatrix} -\frac{4}{5} \\ \frac{3}{5} \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \widetilde{\mathbf{a}}_3 - \|\widetilde{\mathbf{a}}_3\|_2\mathbf{e}_3 = \begin{bmatrix} -\frac{9}{5} \\ \frac{3}{5} \end{bmatrix},$$

$$\widetilde{\mathbf{H}}_3 = \mathbf{I} - \frac{2}{\|\mathbf{v}_3\|_2^2}\mathbf{v}_3\mathbf{v}_3^T = \begin{bmatrix} -\frac{4}{5} & \frac{3}{5} \\ \frac{3}{5} & \frac{4}{5} \end{bmatrix}, \quad \mathbf{H}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{4}{5} & \frac{3}{5} \\ 0 & 0 & \frac{3}{5} & \frac{4}{5} \end{bmatrix},$$

$$\mathbf{A}^{(3)} = \mathbf{H}_3\mathbf{A}^{(2)} = \begin{bmatrix} 3 & 0 & 8 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

- And letting $\mathbf{R} = \mathbf{A}^{(3)}$ and

$$\mathbf{Q} = \mathbf{H}_1\mathbf{H}_2\mathbf{H}_3 = \begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -2 & 1 & 0 \\ 2 & 2 & 0 & -1 \\ 0 & 1 & 2 & 2 \end{bmatrix},$$

  we obtain the full QR for $\mathbf{A}^T$.

2) We can obtain the thin QR decomposition for tall $\mathbf{A}$,

$$\mathbf{A}^T = \mathbf{Q}\mathbf{R} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1 + \mathbf{Q}_2\mathbf{0},$$

with

$$\mathbf{Q}_1 = \frac{1}{3}\begin{bmatrix} 1 & 0 & -2 \\ 2 & -2 & 1 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{R}_1 = \begin{bmatrix} 3 & 0 & 8 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that

$$\mathbf{A}\mathbf{x} = \mathbf{R}_1^T \mathbf{Q}_1^T \mathbf{x} + \mathbf{0}^T \mathbf{Q}_2^T \mathbf{x} = \mathbf{b}\,,$$

and $\mathbf{Q}_2^T \mathbf{x}$ can be anything. To get the minimum norm solution, we set $\mathbf{Q}_2^T \mathbf{x} = 0$. Therefore, one possible solution is given by

$$\mathbf{x} = \mathbf{Q}\begin{bmatrix} \mathbf{R}_1^{-T}\mathbf{b} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \frac{17}{9} & \frac{2}{9} & -\frac{2}{3} \\ -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\ \frac{2}{9} & \frac{2}{9} & 0 \\ -\frac{16}{9} & -\frac{1}{9} & \frac{2}{3} \end{bmatrix} \mathbf{b}\,.$$

VI. Solving LS via Projection

Problem 6. (Bonus question: 6 points + 4 points) This problem is graded by Song Mao (**maosong@**).

Consider the Least Square (LS) problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 \tag{3}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$ $(m>n)$ may not be full rank. Denote

$$X_{\text{LS}} = \left\{\mathbf{x}\in\mathbb{R}^n | \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}\right\}$$

as the set of all solutions to (3), and

$$\mathbf{x}_{\text{LS}} = \mathbf{A}^\dagger\mathbf{y}$$

where $\mathbf{A}^\dagger \in \mathbb{R}^{n\times m}$ is the *pseudo inverse of* $\mathbf{A}$ satisfies the following properties:

1) $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$.
2) $\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger = \mathbf{A}^\dagger$.
3) $(\mathbf{A}\mathbf{A}^\dagger)^T = \mathbf{A}\mathbf{A}^\dagger$.
4) $(\mathbf{A}^\dagger\mathbf{A})^T = \mathbf{A}^\dagger\mathbf{A}$.

Answer the following questions:

1) Prove that $\mathbf{x}_{\text{LS}}$ is a solution to (3) and is of minimum 2-norm in X_{LS}, that is

$$\mathbf{x}_{\text{LS}} = \arg\min_{\mathbf{x}\in X_{\text{LS}}} \|\mathbf{x}\|_2$$

Hint. Notice that the orthogonal projection onto $\mathcal{N}(A)$ is given by

$$\mathbf{\Pi}_{\mathcal{N}(A)} = \mathbf{I} - \mathbf{A}^\dagger\mathbf{A}$$

2) Prove that $X_{\text{LS}} = \{\mathbf{x}_{\text{LS}}\}$ if and only if $\text{rank}(\mathbf{A}) = n$.

Solution.

1) Since

$$X_{\text{LS}} = \{\mathbf{x}|\ \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}\}$$

By the definition of $\mathbf{A}^\dagger$, we have

$$\begin{aligned}
\mathbf{A}^T\mathbf{A}\mathbf{x}_{\text{LS}} &= \mathbf{A}^T\mathbf{A}(\mathbf{A}^\dagger\mathbf{y}) \\
&= \mathbf{A}^T(\mathbf{A}\mathbf{A}^\dagger)\mathbf{y} \\
&= \mathbf{A}^T(\mathbf{A}\mathbf{A}^\dagger)^T\mathbf{y} \\
&= \mathbf{A}^T(\mathbf{A}^\dagger)^T\mathbf{A}^T\mathbf{y} \\
&= (\mathbf{A}\mathbf{A}^\dagger\mathbf{A})^T\mathbf{y} \\
&= \mathbf{A}^T\mathbf{y} \\
&\Rightarrow \mathbf{A}^\dagger\mathbf{y} \in X_{\text{LS}}
\end{aligned}$$

Any solution to the LS problem (3) can be written as

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y} + \tilde{\mathbf{z}} = \mathbf{A}^{\dagger}\mathbf{y} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{z}, \quad \mathbf{z} \in \mathbb{R}^n$$

where $\tilde{\mathbf{z}} \in \mathcal{N}(\mathbf{A})$ is the projection of $\mathbf{z}$ onto $\mathcal{N}(\mathbf{A})$. Note that

$$\left[(\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{z}\right]^T \left(\mathbf{A}^{\dagger}\mathbf{y}\right) = 0$$

For any $\mathbf{x} \in X_{\text{LS}}$, we have

$$\begin{aligned} \|\mathbf{x}\|_2^2 &= \left\|\mathbf{A}^{\dagger}\mathbf{y} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{z}\right\| \\ &= \left\|\mathbf{A}^{\dagger}\mathbf{y}\right\|_2^2 + \left\|(\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{z}\right\|_2^2 \\ &\geq \left\|\mathbf{A}^{\dagger}\mathbf{y}\right\|_2^2 \end{aligned}$$

with equality holds if and only if $\mathbf{z} = \mathbf{0}$.

2) If $X_{\text{LS}} = \{\mathbf{x}_{\text{LS}}\}$, then $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$ by the previous proof, which means $\mathcal{R}(\mathbf{A}) = \mathbb{R}^n$ and $\text{rank}(\mathbf{A}) = \dim(\mathcal{R}(\mathbf{A})) = n$.

If $\text{rank}(\mathbf{A}) = n$, then $\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^T\mathbf{A}) = \{\mathbf{0}\}$, which implies that $X_{\text{LS}} = \{\mathbf{x}_{\text{LS}}\}$.

a) Remark:

1) Please be careful about the difference between a point and a set, for example, X_{LS}, $\mathcal{N}(\mathbf{A})$ are both sets (the latter is also a subspace), and x_{LS} is point, so $X_{\text{LS}} = x_{\text{LS}}$ is not right.
2) Please note that $(\mathcal{N}(\mathbf{A}))^{\perp} \neq \mathcal{R}(\mathbf{A})$, you may refer to slides TBD for details.
3) Some of you use optimization techniques to prove that x_{LS} is a unique solution, it's necessary that you prove the objective function is convex (which is obvious in this problem), your result will collpse if the objective function is concave.
4) How to prove that $\mathcal{N}(\mathbf{A}^T\mathbf{A}) = \mathcal{N}(\mathbf{A})$:
5) Some students are confused about how to prove problem 2), the point is to prove the linear system $\mathbf{A}^T\mathbf{A}x = \mathbf{A}^T y$ has a unique solution, it is not enough to prove that $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}$ when $\mathbf{A}$ is full rank, since it may not be the only one (and we actually don't need this conclusion).

Appendix

Codes of **Problem 3** for reference:

- Matlab version

```
function [loss, Q]= Classical_GS(A)
    Q = A
    Q(:,1) = A(:,1)/norm(A(:,1))
    for i = 1:size(A,2)
        for j = 1:i-1
            Q(:,i) = Q(:,i) - Q(:,j)’*A(:,i)*Q(:,j)
        end
        Q(:,i) = Q(:,i)/norm(Q(:,i));
    end
    loss = norm(Q’*Q-eye(size(A,2)), ’fro’)
end

function [loss, Q]= Modified_GS(A)
    Q = A
    Q(:,1) = A(:,1)/norm(A(:,1))
    for i = 1:size(A,2)
        for j = 1:i-1
            Q(:,i) = Q(:,i) - Q(:,j)’*Q(:,i)*Q(:,j)
        end
        Q(:,i) = Q(:,i)/norm(Q(:,i));
    end
    loss = norm(Q’*Q-eye(size(A,2)), ’fro’)
end
```

- Python version

```
import numpy as np

def Classical_GS(A):
    V = np.zeros_like(A)
    Q = np.zeros_like(A)
    _,n = A.shape
    V[:,0] = A[:,0]
    Q[:,0] = V[:,0] / np.sqrt(np.sum(V[:,0] ** 2))

    for j in range(1,n):
        V[:,j] = A[:,j]
        for i in range(0,j):
            V[:,j] = V[:,j] - Q[:,i].T.dot(A[:,j]) * Q[:,i]
        Q[:,j] = V[:,j] / np.sqrt(np.sum(V[:,j] ** 2))

    return Q

def Modified_GS(A):
    V = np.zeros_like(A)
    Q = np.zeros_like(A)
    _,n = A.shape
    for j in range(0,n):
        V[:,j] = A[:,j]
    for i in range(0,n):
        Q[:,i] = V[:,i] / np.sqrt(np.sum(V[:,i] ** 2))
        for j in range(i+1,n):
            V[:,j] = V[:,j] - Q[:,i].T.dot(V[:,j])*Q[:,i]

    return Q

def F_norm(A):
    return np.sqrt(np.sum(np.abs(A) ** 2))
```