暨南大学本科实验报告专用纸

课程名称 数值计算实验 成绩评定

实验项目名称 Computing Problems 指导老师 Liangda Fang

实验项目编号 03 实验项目类型 验证型 实验地点

学生姓名 学号

学院 国际学院 系 专业 计算机科学与技术

实验时间 2023 年 11 月 10 日上午 10:30 ～ 12:10

I. Problem

Given two inconsistent systems as follows:

$$\text{(a)} \begin{bmatrix} 3 & -1 & 2 \\ 4 & 1 & 0 \\ -3 & 2 & 1 \\ 1 & 1 & 5 \\ -2 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ -5 \\ 15 \\ 0 \end{bmatrix} \text{ and (b) } \begin{bmatrix} 4 & 2 & 3 & 0 \\ -2 & 3 & -1 & 1 \\ 1 & 3 & -4 & 2 \\ 1 & 0 & 1 & -1 \\ 3 & 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

1. Write a program that implements classical Gram-Schmidt to find the full QR factorization, and report the matrices **Q** and **R**.
2. Repeat the first question, but implement Householder reflections and report each Householder reflector $\mathbf{H_i}$ of every step, the matrices **Q** and **R**.
3. Report the least squares solution and 2-norm error.

II. Algorithm Summary

1. Least squares

What is the least square method? We do further analysis with the following functions:

$$\phi(x) = \sum(\text{ Observe } - \text{ Predict })^2$$

Our objective is to minimize the function $\Phi(x)$. Here, y_i represents observed values, such as those obtained from a temperature sensor, while $f(x_i)$ signifies the values we aim to forecast or predict. This principle underlines the essence of ”least squares,” which is frequently referred to as “Data Fitting” in our everyday conversations.

In mathematical modeling, data fitting stands out as an effective technique for enhancing data sets. Let’s consider the application of a linear function to fit these three points:

$$\text{Fitting function}\rightarrow f(x) = d_0 + d_1 x$$

$$\text{Data points}\rightarrow (1,2), (-1,1), (1,3)$$

Matrix representation:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

This is evidently an unsolved system of equations, meaning it is inconsistent. Nonetheless, our aim is to determine the closest possible solution. Upon revisiting our system of equations, it becomes apparent that the dimension of the space spanned by our matrix **A** is smaller than that of vector **b**.

Therefore, finding the nearest solution involves projecting the vector **b** into the geometric space where **A** is situated. The error in the solution obtained through this method coincides with the normal length of the vector **b** projected into space **A**. The following schematic diagram illustrates this process:

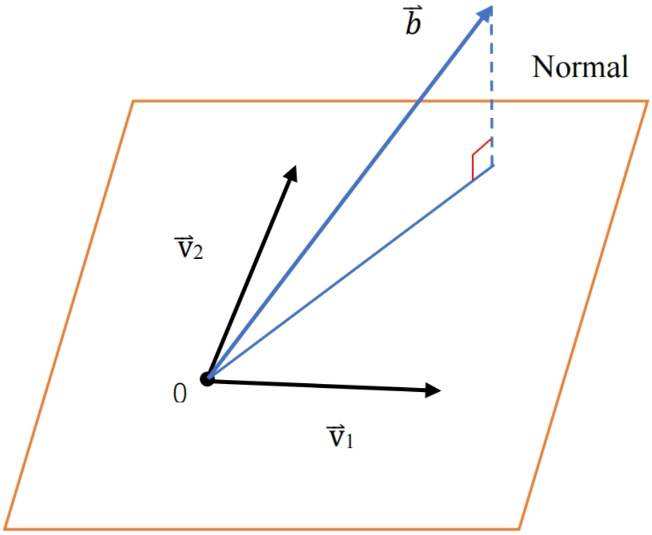

Figure 1: The schematic diagram

Formulated mathematically as:

$$\min(\phi(\mathbf{x})) = \min(||\mathbf{b} - \mathbf{Ax}||_2)$$

Noted that vector normal is perpendicular to space **A**, and:

$$\begin{aligned}
&\text{normal} = \mathbf{b} - \mathbf{A}\bar{\mathbf{x}} \\
&\text{normal} \perp \mathbf{Ax}, \mathbf{x} \in \mathbf{R}^n \\
&\therefore (\mathbf{Ax})^\top \times \text{normal} = 0 \\
&\therefore \mathbf{x}^\top \mathbf{A}^\top \times \text{normal} = 0 \\
&\because \mathbf{x} \neq \vec{0} \\
&\therefore \mathbf{A}^\top(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}) = 0 \\
&\therefore \mathbf{A}^\top\mathbf{A}\bar{\mathbf{x}} = \mathbf{A}^\top\mathbf{b}
\end{aligned}$$

$\mathbf{x}$ represents the closest solution we are seeking!

$$\therefore \mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \times \mathbf{b}$$

2. Classical Gram-Schmidt orthogonalization

We can resort to the normal equation to address the least squares problem; however, if the condition number of the matrix formed by the normal equation becomes excessively high, this approach becomes unreliable. Is it possible to circumvent the necessity of calculating the matrix $(\mathbf{A}^\top \mathbf{A})^{-1}$?

Let's introduce the concept of QR decomposition, which breaks down the matrix $\mathbf{A}$ into two matrices $\mathbf{Q}$ and $\mathbf{R}$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix. But how do we carry out this decomposition? It's essential to note that $\mathbf{Q}$ is an orthogonal matrix; thus, we initially use this matrix to generate a set of standard orthogonal bases. This generation process is accomplished through the Gram-Schmidt orthogonalization method, as described below:

$$\begin{aligned}
&\mathbf{A} \text{ is a } m \times n \text{ matrix , and } \operatorname{rank}(\mathbf{A}) = n \\
&\therefore \mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n] \\
&\text{Standard orthogonal bases } \langle q_1, q_2, \ldots, q_n \rangle \,. \\
&q_i \text{ is a } m\text{-dimension vector} \\
&\text{For } q_1, \text{ we just use} \mathbf{A}_1 \text{ replace it. But maybe } \|A_1\|_2 \neq 1 \\
&\therefore q_1 = \frac{\mathbf{A}_1}{\|\mathbf{A}_1\|_2}
\end{aligned}$$

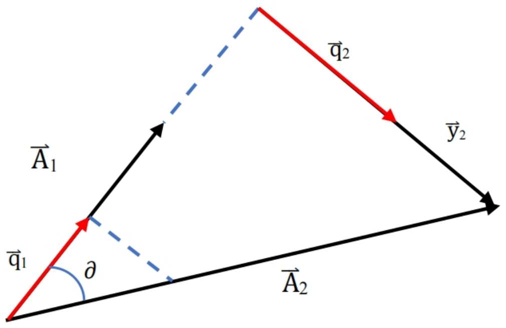

Figure 2: Decomposition diagram of matrix A

Now we want to use $\mathbf{A_2}$ to generate $\mathbf{q_2}$, noted that:

$$
\begin{aligned}
&\therefore \mathbf{q_1}^\top \mathbf{A_2} = |\mathbf{q_1}||\mathbf{A_2}|\cos\partial \\
&\therefore \mathbf{y_2} = \mathbf{A}_2 - \mathbf{q}_1 * \left(q_1^\top \mathbf{A}_2\right) \\
&\therefore \mathbf{q_2} = \frac{y_2}{\|\mathbf{y_2}\|_2}
\end{aligned}
$$

Now, maybe you've already guessed it, for q_n:

$$
\begin{aligned}
&\mathbf{y_n} = \mathbf{A_n} - \mathbf{q_1} \times \left(\mathbf{q_1}^\top \mathbf{A_n}\right) - \mathbf{q_2} \times \left(\mathbf{q_2}^\top \mathbf{A_n}\right) - \cdots - \mathbf{q_{n-1}} \times \left(\mathbf{q_{n-1}}^\top \mathbf{A_n}\right) \\
&\mathbf{q_n} = \frac{\mathbf{y_n}}{\|\mathbf{y_n}\|_2}
\end{aligned}
$$

It is also very simple to prove its orthogonality, as follows:

$$\text{For } i < j < n$$

Step 1 ($j = 2$):

$$\mathbf{q_1}^\top \mathbf{y_2} = \mathbf{q_1}^\top \mathbf{A_2} - \mathbf{q_1}^\top \times \mathbf{q_1} \times \left(\mathbf{q_1}^\top \mathbf{A_2}\right) = 0 \quad \text{Proof!}$$

Step 2 ($j < k$):

$$\text{Assume } \mathbf{q_i}^\top \mathbf{y_j} = 0$$

Step 3 ($j = k$):

$$
\begin{aligned}
\mathbf{q_i}^\top \times \mathbf{y_j} &= \mathbf{q_i}^\top \times \left(\mathbf{A_j} - \sum_{c=1}^{j-1} \mathbf{q_c} \times \left(\mathbf{q_c}^\top \mathbf{A_j}\right)\right) \\
&= \mathbf{q_i}^\top \times \mathbf{A_j} - \mathbf{q_i}^\top \times \mathbf{q_i} \times \left(\mathbf{q_i}^\top \mathbf{A_n}\right) = 0 \quad \text{Proof!}
\end{aligned}
$$

Now we have generated a set of **standard orthogonal bases $\mathbf{q_i}$**, but how should we construct the upper triangular matrix $\mathbf{R}$? If we look at the Gram-Schmidt orthogonalprocedure again, we can deform the above equation a little:

$$
\begin{aligned}
\mathbf{A_n} &= \mathbf{q_1}(\mathbf{q_1}^\top \mathbf{A_n}) + \cdots + \mathbf{q_{n-1}}(\mathbf{q_{n-1}}^\top \mathbf{A_n}) + \mathbf{y_n} \\
&= \mathbf{q_1}(\mathbf{q_1}^\top \mathbf{A_n}) + \cdots + \mathbf{q_{n-1}}(\mathbf{q_{n-1}}^\top \mathbf{A_n}) + \mathbf{q_n} \times \|\mathbf{y_n}\|_2
\end{aligned}
$$

That is to say:

$$
\begin{aligned}
&\mathbf{A} = \mathbf{Q} \times \mathbf{R} \\
&= [\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_n}] \times \begin{bmatrix} \|\mathbf{y_1}\|_2 & \mathbf{q_1}^\top \mathbf{A_2} & \ldots & \mathbf{q_1}^\top \mathbf{A_n} \\ 0 & \ldots & & \ldots \\ 0 & 0 & & \|\mathbf{y_n}\|_2 \end{bmatrix}
\end{aligned}
$$

We make it, isn't it? Basically, more detailed:

$$
\begin{aligned}
&\mathbf{r_{ij}} \text{ means } \mathbf{R}(\mathbf{i}, \mathbf{j}) \\
&\mathbf{r_{ii}} = \|\mathbf{y_i}\|_2 \\
&\mathbf{r_{ij}} = \mathbf{q_i}^\top \mathbf{A_j}
\end{aligned}
$$

Do you think it's complete? If $m < n$, then we can only obtain m orthogonal bases, which describe an n -dimensional space, not an m -dimensional space. Hence, it's termed as reduced QR decomposition. To overcome this limitation, we can introduce $n—m$ vectors, linearly independent from $\mathbf{q_m}$, denoted as $u = 1,\ 2,\ ...,\ n - m$ and then proceed with the aforementioned steps. This enables us to acquire n orthogonal bases, referred to as full QR decomposition.

Now, we can employ the Gram-Schmidt orthogonalization to decompose the matrix $\mathbf{A}$ into matrices $\mathbf{Q}$ and $\mathbf{R}$. Let's not forget that our primary objective is to find the most approximate solution to the system of inconsistent equations:

$$
\begin{aligned}
&\min \phi(\mathbf{x}) = \min(\mathbf{b} - \mathbf{A}\mathbf{x}) \\
&\because \mathbf{A} = \mathbf{QR} \\
&\therefore \mathbf{QRx} = \mathbf{b} \\
&\because \mathbf{Q}^\top \mathbf{Q} = \mathbf{I} \\
&\therefore \mathbf{Q}^\top = \mathbf{Q}^{-1} \\
&\therefore \mathbf{Rx} = \mathbf{Q}^\top \mathbf{b}
\end{aligned}
$$

Noted that $\mathbf{R}$ is $m \times n$, and $\mathbf{Q}$ is $m \times m$.

$$
\begin{aligned}
&\therefore \mathbf{e} = \mathbf{Rx} - \mathbf{Q}^\top \mathbf{b}, \quad \text{size}(\mathbf{e}) \text{ is } m \times 1 \\
&\therefore \min(\mathbf{b} - \mathbf{Ax}) = \min(\mathbf{e})
\end{aligned}
$$

$$
\because \mathbf{e} = \begin{bmatrix} \mathbf{e_1} \\ \vdots \\ \mathbf{e_n} \\ \mathbf{e_{n+1}} \\ \vdots \\ \mathbf{e_m} \end{bmatrix}, \quad \text{we can only minimize } (\mathbf{e_1}, \ldots, \mathbf{e_n}) \text{ to } (0, \ldots, 0) \text{ when } \hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{d}}.
$$

where $\mathbf{R} = \begin{bmatrix} \hat{\mathbf{R}} \\ 0 \end{bmatrix}$, and $\mathbf{d} = \mathbf{Q}^\top \mathbf{b}$ and $\mathbf{d} = \begin{bmatrix} \hat{\mathbf{d}} \\ d_{n+1} \\ \vdots \\ d_m \end{bmatrix}$.

Therefore, the closest solution $\mathbf{x}$:

$$
\begin{aligned}
\mathbf{x} &= \hat{\mathbf{R}}^{-1} \hat{\mathbf{d}} \\
\text{error } &= \sqrt{d_{n+1}^2 + \cdots + d_m^2}
\end{aligned}
$$

3. Householder reflections

We understand that employing QR decomposition helps evade the computation of large conditionally sensitive matrices like $\mathbf{A}^\top \mathbf{A}$. However, employing the standard Gram-Schmidt orthogonalization method for QR decomposition isn't consistently stable. This instability stems from the susceptibility to rounding errors in computers, potentially resulting in the generated orthogonal bases being non-orthogonal. Hence, we aim to introduce a novel concept known as the Householder reflection matrix:

$$\begin{aligned} \exists \mathbf{x}, \omega \text{ and } \|\mathbf{x}\|_2 &= \|\omega\|_2 \\ \text{let } \mathbf{u} &= \omega - \mathbf{x} \\ \text{let } \mathbf{v} &= \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \\ \mathbf{H} &= \mathbf{I} - 2\mathbf{v}\mathbf{v}^\top \end{aligned}$$

The **H** is the so-called Householder reflection matrix! It has these properties:

$$\begin{aligned} \mathbf{H}\mathbf{x} &= \omega \\ \mathbf{H}\omega &= \mathbf{x} \end{aligned}$$

Now, we will use the House Holder reflection to decompose **A** into QR decomposition:

$$\begin{aligned}
&\because \mathbf{R} \text{ is an upper triangular matrix} \\
&\therefore \text{let } \mathbf{A}_{12}, \ldots, \mathbf{A}_{1n} \text{ be } \mathbf{0} \\
&\therefore \mathbf{x} = \mathbf{A}_1, \omega = (\|\mathbf{x}\|_2, \mathbf{0}, \ldots, \mathbf{0}) \\
&\therefore \|\mathbf{x}\|_2 = \|\omega\|_2 \\
&\therefore \mathbf{u} = \omega - \mathbf{x}, \mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \\
&\therefore \widehat{\mathbf{H}_1} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^\top \\
&\therefore \mathbf{H}_1 = \widehat{\mathbf{H}_1} \text{ (we should keep } \mathbf{H}_i \text{ size as } m \times m) \\
&\mathbf{H}_1\mathbf{A} = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & & \\ \ldots & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & & \\ \mathbf{0} & \mathbf{x} & \mathbf{x} \end{bmatrix} \\
&\therefore \text{let } \mathbf{A}_{23}, \ldots, \mathbf{A}_{2n} \text{ be } \mathbf{0} \\
&\text{Now just } \mathbf{x} = (\mathbf{A}_{22}, \mathbf{A}_{23}, \ldots, \mathbf{A}_{2n}), \omega = (\|\mathbf{x}\|_2, \mathbf{0}, \ldots, \mathbf{0}) \\
&\text{And other procedure same as above.} \\
&\therefore \mathbf{H}_2 = \begin{bmatrix} 1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & & & \\ \ldots & & \widehat{\mathbf{H}_2} & \\ \mathbf{0} & & & \end{bmatrix} \text{(keep } \mathbf{H}_i \text{ size as } m \times m)
\end{aligned}$$

And $\mathbf{H}_n$ can be obtained by above procedures. Eventually we will get:

$$\begin{aligned} &\mathbf{H}_n\mathbf{H}_{n-1}\ldots\mathbf{H}_2\mathbf{H}_1\mathbf{A} = \mathbf{R} \\ &\because \mathbf{H}_i = \mathbf{H}_i^{-1} \\ &\therefore \mathbf{A} = \mathbf{H}_1\ldots\mathbf{H}_n\mathbf{R} \\ &\therefore \mathbf{Q} = \mathbf{H}_1\ldots\mathbf{H}_n \end{aligned}$$

That is the application of Householder in QR decomposition.

III. Experimental procedures

Step1: Define the inconsistent system of linear equations $\mathbf{A}_\mathrm{a}\mathbf{x} = \mathbf{b}_\mathrm{a}$ and $\mathbf{A}_\mathrm{b}\mathbf{x} = \mathbf{b}_\mathrm{b}$

Step2: Define the classic Gram-Schmidt method and the Householder reflections method to calculate the full QR factorization of the matrix $\mathbf{A}_\mathrm{a}$ and the matrix $\mathbf{A}_\mathrm{b}$.

Step3: Use the classic Gram-Schmidt method to calculate the QR decomposition of matrix $\mathbf{A}_\mathrm{a}$ and matrix $\mathbf{A}_\mathrm{b}$, report the matrices $\mathbf{Q}_\mathrm{a}$、$\mathbf{R}_\mathrm{a}$ and $\mathbf{Q}_\mathrm{b}$、$\mathbf{R}_\mathrm{b}$.

Step4: Use the Householder reflections method to calculate the QR decomposition of matrix $\mathbf{A}_\mathrm{a}$ and matrix $\mathbf{A}_\mathrm{b}$, report the matrices $\mathbf{Q}_\mathrm{a}$、$\mathbf{R}_\mathrm{a}$ and $\mathbf{Q}_\mathrm{b}$、$\mathbf{R}_\mathrm{b}$, also report each Householder reflector $\mathbf{H}_\mathrm{i}$ of every step.

Step5: Report the least squares solution $\mathbf{x}_\mathrm{a}$、$\mathbf{x}_\mathrm{b}$ and 2-norm error e_a、e_b.

IV. Result Analysis

1. QR factorization using Gram-Schmidt method

The **Q** matrix and **R** matrix obtained by performing full QR factorization using the Gram-Schmidt method on the matrix $\mathbf{A}_\mathrm{a}$ are as follows:

$$\mathbf{Q}_\mathrm{a} = \begin{bmatrix} 0.48038446 & -0.26969003 & 0.4057325 & 0.49712418 & -0.53361666 \\ 0.64051262 & 0.54936859 & -0.22364767 & 0.28585917 & 0.39522651 \\ -0.48038446 & 0.65924231 & -0.0310072 & 0.3830112 & -0.43240181 \\ 0.16012815 & 0.42950635 & 0.69139457 & -0.55475739 & -0.06403956 \\ -0.32025631 & -0.07990816 & 0.55351149 & 0.46550912 & 0.60661097 \end{bmatrix}$$

$$\mathbf{R}_\mathrm{a} = \begin{bmatrix} 6.244998 & -0.64051262 & 0.32025631 \\ 0. & 2.56704959 & 2.02766952 \\ 0. & 0. & 5.89796509 \\ 0. & 0. & 0. \\ 0. & 0. & 0. \end{bmatrix}$$

The **Q** matrix and **R** matrix obtained by performing full QR factorization using the Gram-Schmidt method on the matrix $\mathbf{A}_\mathrm{b}$ are as follows:

$$\mathbf{Q}_\mathrm{b} = \begin{bmatrix} 0.71842121 & 0.21150374 & 0.2258696 & 0.6085938 & 0.13316772 \\ -0.3592106 & 0.7684636 & 0.52281184 & -0.05163138 & 0.06658386 \\ 0.1796053 & 0.59926061 & -0.76877528 & -0.13204504 & -0.01331677 \\ 0.1796053 & -0.056401 & 0.05246724 & -0.40711782 & 0.8922237 \\ 0.53881591 & 0.04935087 & 0.28615111 & -0.66615837 & -0.42613669 \end{bmatrix}$$

$$\mathbf{R}_\mathrm{b} = \begin{bmatrix} 5.56776436 & 1.43684242 & 3.59210604 & -1.25723711 \\ 0. & 4.57553099 & -2.43934318 & 1.92468407 \\ 0. & 0. & 4.14081864 & -1.63950817 \\ 0. & 0. & 0. & 1.42371311 \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

It should be noted that the extra columns of the **Q** matrix obtained by using the Gram-Schmidt method for complete QR decomposition compared with the original **A** matrix are complementary columns. The complementary columns only need to satisfy the columns of the original matrix and themselves are orthogonal to each other, and the completion columns should be unit vectors.

This also means that the **Q** matrix obtained by QR decomposition is not unique. When the question1_solution.py code is run multiple times, the results of the **Q** matrix obtained are not the same.

2. QR factorization using Householder reflections method

The **Q** matrix and **R** matrix obtained by performing full QR factorization using the Householder reflections method on the matrix $\mathbf{A}_\mathrm{a}$ are as follows:

$$\mathbf{Q}_\mathrm{a} = \begin{bmatrix} 0.48038446 & -0.26969003 & 0.4057325 & 0.70977181 & -0.16764 \\ 0.64051262 & 0.54936859 & -0.22364767 & 0.0182042 & 0.48743007 \\ -0.48038446 & 0.65924231 & -0.0310072 & 0.55866251 & -0.14685061 \\ 0.16012815 & 0.42950635 & 0.69139457 & -0.42575741 & -0.36136885 \\ -0.32025631 & -0.07990816 & 0.55351149 & 0.05019364 & 0.76299162 \end{bmatrix}$$

$$\mathbf{R}_\mathrm{a} = \begin{bmatrix} 6.244998 & -0.64051262 & 0.32025631 \\ 0. & 2.56704959 & 2.02766952 \\ 0. & 0. & 5.89796509 \\ 0. & 0. & 0. \\ 0. & 0. & 0. \end{bmatrix}$$

The Householder reflector of every step are shown as follows:

$$\mathbf{H}_\mathrm{a0} = \begin{bmatrix} 0.48038446 & 0.64051262 & -0.48038446 & 0.16012815 & -0.32025631 \\ 0.64051262 & 0.21046162 & 0.59215378 & -0.19738459 & 0.39476919 \\ -0.48038446 & 0.59215378 & 0.55588466 & 0.14803845 & -0.29607689 \\ 0.16012815 & -0.19738459 & 0.14803845 & 0.95065385 & 0.0986923 \\ -0.32025631 & 0.39476919 & -0.29607689 & 0.0986923 & 0.80261541 \end{bmatrix}$$

$$\mathbf{H}_\mathrm{a1} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 0.21693074 & 0.90857069 & 0.34639689 & 0.08631076 \\ 0. & 0.90857069 & -0.05418607 & -0.40191344 & -0.10014367 \\ 0. & 0.34639689 & -0.40191344 & 0.84676859 & -0.03818025 \\ 0. & 0.08631076 & -0.10014367 & -0.03818025 & 0.99048673 \end{bmatrix}$$

$$\mathbf{H}_\mathrm{a2} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & -0.08530977 & 0.93873317 & 0.33391958 \\ 0. & 0. & 0.93873317 & 0.18804752 & -0.28882204 \\ 0. & 0. & 0.33391958 & -0.28882204 & 0.89726225 \end{bmatrix}$$

The **Q** matrix and **R** matrix obtained by performing full QR factorization using the Householder reflections method on the matrix $\mathbf{A}_\mathrm{b}$ are as follows:

$$\mathbf{Q}_\mathrm{b} = \begin{bmatrix} 0.71842121 & 0.21150374 & -0.2258696 & -0.6085938 & -0.13316772 \\ -0.3592106 & 0.7684636 & -0.52281184 & 0.05163138 & -0.06658386 \\ 0.1796053 & 0.59926061 & 0.76877528 & 0.13204504 & 0.01331677 \\ 0.1796053 & -0.056401 & -0.05246724 & 0.40711782 & -0.8922237 \\ 0.53881591 & 0.04935087 & -0.28615111 & 0.66615837 & 0.42613669 \end{bmatrix}$$

$$\mathbf{R}_\mathrm{b} = \begin{bmatrix} 5.56776436 & 1.43684242 & 3.59210604 & -1.25723711 \\ 0. & 4.57553099 & -2.43934318 & 1.92468407 \\ 0. & 0. & -4.14081864 & 1.63950817 \\ 0. & 0. & 0. & -1.42371311 \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

The Householder reflector of every step are shown as follows:

$$\mathbf{H}_{\mathrm{b0}} = \begin{bmatrix} 0.71842121 & -0.3592106 & 0.1796053 & 0.1796053 & 0.53881591 \\ -0.3592106 & 0.54175434 & 0.22912283 & 0.22912283 & 0.68736848 \\ 0.1796053 & 0.22912283 & 0.88543859 & -0.11456141 & -0.34368424 \\ 0.1796053 & 0.22912283 & -0.11456141 & 0.88543859 & -0.34368424 \\ 0.53881591 & 0.68736848 & -0.34368424 & -0.34368424 & -0.03105272 \end{bmatrix}$$

$$\mathbf{H}_{\mathrm{b1}} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 0.49864787 & 0.73416847 & 0.07850687 & 0.45407447 \\ 0. & 0.73416847 & -0.07509935 & -0.11496364 & -0.66493616 \\ 0. & 0.07850687 & -0.11496364 & 0.98770659 & -0.07110365 \\ 0. & 0.45407447 & -0.66493616 & -0.07110365 & 0.58874489 \end{bmatrix}$$

$$\mathbf{H}_{\mathrm{b2}} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 0.28105946 & -0.23328535 & -0.93090468 \\ 0. & 0. & -0.23328535 & 0.92430243 & -0.30206451 \\ 0. & 0. & -0.93090468 & -0.30206451 & -0.20536188 \end{bmatrix}$$

$$\mathbf{H}_{\mathrm{b3}} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & -0.16174595 & -0.98683243 \\ 0. & 0. & 0. & -0.98683243 & 0.16174595 \end{bmatrix}$$

It can be observed that the results obtained using the Householder reflections method for matrices $\mathbf{A}_{\mathrm{a}}$ and $\mathbf{A}_{\mathrm{b}}$ are very similar to the results obtained previously using the Gram-Schmidt method.

3. The least squares solution and 2-norm error

The least squares solution for inconsistent systems (a) using the QR decomposition from Gram-Schmidt is:

$$\begin{bmatrix} 2.5246085 & 0.66163311 & 2.09340045 \end{bmatrix}^{\top}$$

The 2-norm error of it is: 2.413492090641352.

The least squares solution for inconsistent systems (a) using the QR decomposition from Householder transformation is:

$$\begin{bmatrix} 2.5246085 & 0.66163311 & 2.09340045 \end{bmatrix}^{\top}$$

The 2-norm error of it is: 2.413492090641354.

The least squares solution for inconsistent systems (b) using the QR decomposition from Gram-Schmidt is:

$$\begin{bmatrix} 1.27389608 & 0.6885086 & 1.21244902 & 1.74968966 \end{bmatrix}^{\top}$$

The 2-norm error of it is: 0.8256398422677529.

The least squares solution for inconsistent systems (b) using the QR decomposition from Householder transformation is:

$$\begin{bmatrix}1.27389608 & 0.6885086 & 1.21244902 & 1.74968966\end{bmatrix}^\top$$

The 2-norm error of it is: 0.8256398422677516.

V. Experimental Summary

Through this experiment, we have gained insight into the principles of data fitting and the practical application of QR decomposition. In this study, we employed two distinct methods to compute the QR decomposition of matrix **A**: the classical Gram-Schmidt method and the Householder reflection method. In general, QR decomposition based on the Householder reflection matrix proves to be more stable and incurs less space-time overhead.

The experimental results indicate that the errors in the closest solutions obtained through both methods are relatively minor.

VI. Appendix: Source Code

1. define_matrices_and_methods.py

```python
import numpy as np

# Define the matrices from the given linear equations
A_a = np.array([
    [3, -1, 2],
    [4, 1, 0],
    [-3, 2, 1],
    [1, 1, 5],
    [-2, 0, 3]
], dtype=float)

b_a = np.array([10, 10, -5, 15, 0], dtype=float)

n_a = A_a.shape[1]

A_b = np.array([
    [4, 2, 3, 0],
    [-2, 3, -1, 1],
    [1, 3, -4, 2],
    [1, 0, 1, -1],
    [3, 1, 3, -2]
], dtype=float)

b_b = np.array([10, 0, 2, 0, 5], dtype=float)

n_b = A_b.shape[1]


# Function to perform the Gram-Schmidt process
def gram_schmidt(A):
    m, n = A.shape
    Q = np.zeros((m, m))  # Q is an m by m orthogonal matrix
    R = np.zeros((m, n))  # R is an m by n upper triangular matrix

    # Use the Gram-Schmidt process for the first n steps
    for j in range(n):
        # Start with the j-th column of A
        v = A[:, j]

        # Subtract the projection of v onto the previous vectors
        for i in range(j):
            q = Q[:, i]
            R[i, j] = np.dot(q, v)
            v = v - R[i, j] * q

        # Normalize v to get the j-th orthogonal vector
        R[j, j] = np.linalg.norm(v)
        Q[:, j] = v / R[j, j]

    # Extend Q to an m by m orthogonal matrix
    for k in range(n, m):
        # Start with a random vector
        v = np.random.rand(m)
        # Make v orthogonal to the previous vectors
        for i in range(k):
            v -= np.dot(Q[:, i], v) * Q[:, i]
        # Normalize v
        v /= np.linalg.norm(v)
        Q[:, k] = v

    return Q, R


# Function to perform QR decomposition using Householder reflections
def householder_reflections(A):
    m, n = A.shape
    R = A.copy()
    Q = np.eye(m)
    H_list = []

    for k in range(n):
        # Extract the vector to reflect on
        x = R[k:, k]
        e = np.zeros_like(x)
        e[0] = np.copysign(np.linalg.norm(x), -A[k, k])
        u = x + e
        u = u / np.linalg.norm(u)
        # Form the Householder reflector H
        H = np.eye(m)
        H[k:, k:] -= 2.0 * np.outer(u, u)
        # Apply the reflector to R
        R = np.dot(H, R)
        # Update Q as well
        Q = np.dot(Q, H.T)
        # Record every each Householder reflector
        H_list.append(H)

    # Ensure R is upper triangular
    R = np.triu(R)

    return Q, R, H_list
```

2. question1_solution.py

```python
from lab3_define_matrices_and_methods import A_a, A_b, gram_schmidt

# Perform QR decomposition using Gram-Schmidt for matrix A_a
Q_a_GS, R_a_GS = gram_schmidt(A_a)

print('(a).␣Q␣matrix␣using␣Gram-Schmidt:\n', Q_a_GS, '\n')
print('(a).␣R␣matrix␣using␣Gram-Schmidt:\n', R_a_GS, '\n')

# Perform QR decomposition using Gram-Schmidt for matrix A_b
Q_b_GS, R_b_GS = gram_schmidt(A_b)

print('(b).␣Q␣matrix␣using␣Gram-Schmidt:\n', Q_b_GS, '\n')
print('(b).␣R␣matrix␣using␣Gram-Schmidt:\n', R_b_GS, '\n')
```

3. question2_solution.py

```python
from lab3_define_matrices_and_methods import A_a, A_b, householder_reflections

# Perform QR decomposition using Householder reflections for matrix A_a
Q_a_H, R_a_H, H_a_list = householder_reflections(A_a)

print('(a).␣Q␣matrix␣using␣Householder␣reflections:\n', Q_a_H, '\n')
print('(a).␣R␣matrix␣using␣Householder␣reflections:\n', R_a_H, '\n')
print('(a).␣H_i␣matrix␣in␣every␣Householder␣reflections␣steps␣are:')
for i, H_a_i in enumerate(H_a_list):
    print('H_{0}␣=␣\n{1}\n'.format(i, H_a_i))

# Perform QR decomposition using Householder reflections for matrix A_b
Q_b_H, R_b_H, H_b_list = householder_reflections(A_b)

print('(b).␣Q␣matrix␣using␣Householder␣reflections:\n', Q_b_H, '\n')
print('(b).␣R␣matrix␣using␣Householder␣reflections:\n', R_b_H, '\n')
print('(b).␣H_i␣matrix␣in␣every␣Householder␣reflections␣steps␣are:')
for i, H_b_i in enumerate(H_b_list):
    print('H_{0}␣=␣\n{1}\n'.format(i, H_b_i))
```

4. question3_solution.py

```
from lab3_define_matrices_and_methods import *

# Solve the least squares problem (a) using the QR decomposition from Gram-Schmidt
Q_a_gs, R_a_gs = gram_schmidt(A_a)
x_a_ls_gs = np.linalg.solve(R_a_gs[:n_a, :n_a], Q_a_gs.T[:n_a, :] @ b_a)  # Only use the first n rows of Q and R
error_a_gs = np.linalg.norm(b_a - A_a @ x_a_ls_gs, 2)

# Solve the least squares problem (a) using the QR decomposition from Householder reflections
Q_a_h, R_a_h, _ = householder_reflections(A_a)
x_a_ls_h = np.linalg.solve(R_a_h[:n_a, :n_a], Q_a_h.T[:n_a, :] @ b_a)
error_a_h = np.linalg.norm(b_a - A_a @ x_a_ls_h, 2)

print('The␣least␣squares␣solution␣for␣problem␣(a)␣'
      'using␣the␣QR␣decomposition␣from␣Gram-Schmidt␣is:\n', x_a_ls_gs, '\n')
print('The␣2-norm␣error␣of␣it␣is:\n', error_a_gs, '\n')
print('The␣least␣squares␣solution␣for␣problem␣(a)␣'
      'using␣the␣QR␣decomposition␣from␣Householder␣transformation␣is:\n', x_a_ls_h, '\n')
print('The␣2-norm␣error␣of␣it␣is:\n', error_a_h, '\n')

# Solve the least squares problem (b) using the QR decomposition from Gram-Schmidt
Q_b_gs, R_b_gs = gram_schmidt(A_b)
x_b_ls_gs = np.linalg.solve(R_b_gs[:n_b, :n_b], Q_b_gs.T[:n_b, :] @ b_b)
error_b_gs = np.linalg.norm(b_b - A_b @ x_b_ls_gs, 2)

# Solve the least squares problem (b) using the QR decomposition from Householder reflections
Q_b_h, R_b_h, _ = householder_reflections(A_b)
x_b_ls_h = np.linalg.solve(R_b_h[:n_b, :n_b], Q_b_h.T[:n_b, :] @ b_b)
error_b_h = np.linalg.norm(b_b - A_b @ x_b_ls_h, 2)

print('The␣least␣squares␣solution␣for␣problem␣(b)␣'
      'using␣the␣QR␣decomposition␣from␣Gram-Schmidt␣is:\n', x_b_ls_gs, '\n')
print('The␣2-norm␣error␣of␣it␣is:\n', error_b_gs, '\n')
print('The␣least␣squares␣solution␣for␣problem␣(b)␣'
      'using␣the␣QR␣decomposition␣from␣Householder␣transformation␣is:\n', x_b_ls_h, '\n')
print('The␣2-norm␣error␣of␣it␣is:\n', error_b_h, '\n')
```