Suhas Kamath
25945

Numerical Linear Algebra - Assignment 4

2 a)

Theory: The algorithm for computing the columns of $\underset{\sim}{Q}$ is as follows:

for $j = 1 \rightarrow n$:
{
$\qquad \underset{\sim}{v_j} = \underset{\sim}{a_j}$
$\qquad$ for $i = 1 \rightarrow (j-1)$:
$\qquad$ {
$\qquad\qquad r_{ij} = \underset{\sim}{v_j}^T \underset{\sim}{a_i}$
$\qquad\qquad \underset{\sim}{v_j} = \underset{\sim}{v_j} - r_{ij} \underset{\sim}{q_i}$
$\qquad$ }
$\qquad \underset{\sim}{q_j} = \dfrac{\underset{\sim}{v_j}}{\|\underset{\sim}{v_j}\|_2}$
}

Hence, let us now calculate $q_0(x) \cdots q_3(x)$ for $\underset{\sim}{A}$:

i) Calculating $q_0(x)$:

$$v_0(x) = a_0(x) = 1$$

$$\|v_0(x)\|_2 = \sqrt{(a_0(x), a_0(x))} = \sqrt{\int_{-1}^{1} 1^2 dx} = \sqrt{[x]_{-1}^{1}} = \sqrt{2}$$

$$q_0(x) = \frac{v_0(x)}{\|v_0(x)\|_2} = \frac{1}{\sqrt{2}}$$

ii) Calculating $q_1(x)$

$$v_1(x) = a_1(x) - (a_1(x), q_0(x))\, q_0(x)$$

$$(a_1(x), q_0(x)) = \int_{-1}^{1} x \cdot \frac{1}{\sqrt{2}} dx = \frac{1}{\sqrt{2}} \cdot \frac{1}{2} [x^2]_{-1}^{1} = 0$$

$$\therefore v_1(x) = a_1(x) - 0 \cdot q_0(x) = a_1(x) = x$$

$$\|v_1\|_2 = \sqrt{\int_{-1}^{1} x^2\,dx} = \sqrt{\left[\frac{x^3}{3}\right]_{-1}^{1}} = \sqrt{\frac{2}{3}}$$

$$\therefore q_1(x) = \frac{x}{\sqrt{\frac{2}{3}}} = \sqrt{\frac{3}{2}}\,x$$

iii) Calculating $q_2(x)$

$$q_2(x) = a_2(x) - (a_2(x), q_0(x))\,q_0(x) - (a_2(x), q_1(x))\,q_1(x)$$

$$(a_2(x), q_0(x)) = \int_{-1}^{1} x^2 \cdot \frac{1}{\sqrt{2}}\,dx = \frac{1}{\sqrt{2}}\left[x^3\right]_{-1}^{1}$$

$$= \frac{1}{\sqrt{2}} \cdot \frac{1}{3} \cdot 2$$

$$= \frac{\sqrt{2}}{3}$$

$$(a_2(x), q_1(x)) = \int_{-1}^{1} x^2 \cdot \sqrt{\frac{3}{2}}\,x = \sqrt{\frac{3}{2}} \int_{-1}^{1} x^3\,dx$$

$$= \sqrt{\frac{3}{2}} \cdot \frac{1}{4}\left[x^4\right]_{-1}^{1}$$

$$= \sqrt{\frac{3}{2}} \cdot \frac{1}{4} \cdot 0$$

$$= 0$$

$$\therefore v_2(x) = x^2 - \frac{\cancel{\sqrt{2}}}{3}\left(\frac{1}{\cancel{\sqrt{2}}}\right) - \cancel{0 \cdot q_1(x)}\ 0$$

$$= x^2 - 1/3$$

$$\|v_2(x)\|_2 = \sqrt{\int_{-1}^{1} (x^2 - 1/3)^2\,dx}$$

$$= \sqrt{\int_{-1}^{1} \left(x^4 - \frac{2}{3}x^2 + \frac{1}{9}\right) dx}$$

$$= \left[\frac{x^5}{5} - \frac{2x^3}{9} + \frac{1}{9}x\right]_{-1}^{1}$$

$$= \sqrt{\left(\frac{4}{45}\right) - \left(-\frac{4}{45}\right)}$$

$$= \sqrt{\frac{8}{45}}$$

$$= \frac{2}{3}\sqrt{\frac{2}{5}}$$

$$\therefore q_2(x) = \frac{x^2 - 1/3}{\frac{2}{3}\sqrt{\frac{2}{5}}} = \frac{3}{2}\sqrt{\frac{5}{2}}\left(x^2 - \frac{1}{3}\right)$$

$$= \sqrt{\frac{5}{8}}\left(3x^2 - 1\right)$$

iv) Calculating $q_3(x)$:

$$v_3(x) = a_3(x) - (a_3(x), q_0(x))\, q_0(x)$$
$$- (a_3(x), q_1(x))\, q_1(x) - (a_3(x), q_2(x))\, q_2(x)$$

$$(a_3(x), q_0(x)) = \int_{-1}^{1} x^3 \cdot \frac{1}{\sqrt{2}}\, dx = \frac{1}{4\sqrt{2}}\left[x^4\right]_{-1}^{1} = 0$$

$$(a_3(x), q_1(x)) = \int_{-1}^{1} x^3 \cdot \sqrt{\frac{3}{2}}\, x\, dx = \sqrt{\frac{3}{2}} \int_{-1}^{1} x^4\, dx$$

$$= \sqrt{\frac{3}{2}} \cdot \frac{1}{5} \cdot [x^5]_{-1}^{1}$$

$$= \frac{\sqrt{3}}{\cancel{\sqrt{2}}_{1}} \cdot \frac{1}{5} \cdot \cancel{2}^{\sqrt{2}}$$

$$= \frac{\sqrt{3} \cdot \sqrt{2}}{5}$$

$$= \frac{\sqrt{6}}{5}$$

$$(a_3(x), q_2(x)) = \int_{-1}^{1} x^3 \cdot \sqrt{\frac{5}{8}} (3x^2 - 1)\, dx$$

$$= \sqrt{\frac{5}{8}} \int_{-1}^{1} (3x^5 - x^3)\, dx$$

$$= \sqrt{\frac{5}{8}} \left[\frac{3x^6}{6} - \frac{x^4}{4} \right]_{-1}^{1}$$

$$= 0$$

$$\therefore v_3(x) = a_3(x) - \cancel{0 \cdot q_0(x)}^{0} - \frac{\sqrt{6}}{5} q_1(x) - \cancel{0 \cdot q_2(x)}^{0}$$

$$= x^3 - \frac{\sqrt{6}}{5} \cdot \frac{\sqrt{3}}{\sqrt{2}} x$$

$$= x^3 \; \frac{\sqrt{3} \cdot \cancel{\sqrt{2}}}{5} \cdot \frac{\sqrt{3}}{\cancel{\sqrt{2}}} x$$

$$= x^3 - 3/5\, x$$

$$\|v_3(x)\|_2 = \sqrt{\int_{-1}^{1} \left(x^3 - \tfrac{3}{5}x\right)^2 dx}$$

$$= \int_{-1}^{1} \left(x^6 - \frac{6}{5}x^4 + \frac{9}{25}x^2\right) dx$$

Using property of even function:

$$\int_{-a}^{a} f(x)\,dx = 2\int_{0}^{a} f(x)\,dx,$$

if $f(-x) = f(x)$

$$= \sqrt{2\int_{0}^{1} \left(x^6 - \frac{6}{5}x^4 + \frac{9}{25}x^2\right) dx}$$

$$= \sqrt{2\left[\frac{x^7}{7} - \frac{6x^5}{25} + \frac{3x^3}{25}\right]_0^1}$$

$$= \sqrt{2\left(\frac{1}{7} - \frac{6}{25} + \frac{3}{25}\right)}$$

$$= \sqrt{\frac{8}{175}}$$

$$= \frac{2}{5}\sqrt{\frac{2}{7}}$$

$$q_3 = \frac{x^3 - \tfrac{3}{5}x}{\tfrac{2}{5}\sqrt{\tfrac{2}{7}}} = \frac{5}{2}\sqrt{\frac{7}{2}}\left(x^3 - \tfrac{3}{5}x\right)$$

$$= \sqrt{\frac{7}{2}} \cdot \frac{1}{2}\left(5x^3 - 3x\right)$$

$$= \sqrt{\frac{7}{8}}\left(5x^3 - 3x\right)$$

Hence, the expressions for $q_0(x) \dots q_3(x)$ are:

$$q_0(x) = 1/\sqrt{2}$$

$$q_1(x) = \sqrt{\frac{3}{2}}\, x$$

$$q_2(x) = \sqrt{\frac{5}{8}}\,(3x^2 - 1)$$

$$q_3(x) = \sqrt{\frac{7}{8}}\,(5x^3 - 3x)$$

2) b)

The given integral can be expressed as the inner product:

$$\int_{-1}^{1} q_{n-1}(x)\, dx = \int_{-1}^{1} q_{n-1}(x) \cdot 1 \, dx$$

$$= (q_{n-1}(x), 1)$$

Because we know from part (a) that $1 = q_0(x) \cdot \|1\|_2$,

$$\int_{-1}^{1} q_{n-1}(x)\, dx = (q_{n-1}(x), q_0(x) \cdot \|1\|_2)$$

$$= \|1\|_2 \, (q_{n-1}(x), q_0(x))$$

Because we know that $(q_i, q_j) = 0 \;\forall\, i \neq j$, and $n > 2$, we can say that:

$$\int_{-1}^{1} q_{n-1}(x)\, dx = \|1\|_2 \cdot 0 = 0 \quad \text{(proven)}$$

Name:	Suhas Kamath	**Assignment No:**	Assignment 4
SR No:	06-18-01-10-51-25-1-25945	**Course Code:**	DS284
Email ID:	suhaskamath@iisc.ac.in	**Course Name:**	Numerical Linear Algebra
Date:	September 29, 2025	**Term:**	AUG 2025

Question 5

Theory

A least squares problem is basically when we have to approximate a function using only given data points (co-ordinates). Usually, we approximate the function to a polynomial, as follows:

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

We can actually create an interpolating polynomial. However, an interpolating polynomial may result in upto polynomial degree 99 for 100 data points. This shall become cumbersome to handle. Furthermore, Runge phenomenon may occur for such a polynomial. The Runge phenomenon is when the polynomial occilates widly between two consecutive data points, as a result of overfitting the data.
Another method is to calculate a best fit curve. This can be done by minimizing the square error between given data points and the best fit curve. In mathematical terms:

$$\begin{gathered}\mathbf{Ax} = \mathbf{b}, \text{where} \\ \mathbf{A} \in \mathbb{R}^{(m \times n)}, m > n, \\ \mathbf{x} \in \mathbb{R}^n, \\ \mathbf{b} \in \mathbb{R}^m\end{gathered}$$

We can clearly see that this is an overdetermined system. In this question, we have 100 data points and we have been asked to construct a least square curve of degree 14. This means that $m = 100$, and $n = 14$.

The idea of least squares is to find an $\mathbf{x}$ such that the 2-norm of the residual $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ is minimised.

We know that $||\mathbf{r}||_2$ shall be minimised if $\mathbf{r} \perp \mathrm{range}(\mathbf{A})$. This happens when:

$$\begin{gathered}\mathbf{Ax} = \mathbf{Pb}, \text{where} \\ \mathbf{P} = \mathbf{A}(\mathbf{A^T A})^{-1}\mathbf{A^T}\end{gathered}$$

Hence,

$$\begin{gathered}\mathbf{Ax} = \mathbf{A}(\mathbf{A^T A})^{-1}\mathbf{A^T b} \\ \mathbf{x} = (\mathbf{A^T A})^{-1}\mathbf{A^T b}\end{gathered}$$

$$\mathbf{A^T A \hat{x}} = \mathbf{A^T b}$$

Because we get a slightly different value of $\mathbf{x}$ when solving this normal equation, we have replaced it with $\hat{\mathbf{x}}$.

Such an equation can be solved in an easier way by using decomposition.

QR Decomposition

Because $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$, and $\mathbf{Ax} = \mathbf{Pb}$,

$$\begin{gathered}\hat{\mathbf{Q}}\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}\widehat{\mathbf{Q^T}}\mathbf{b} \\ \hat{\mathbf{R}}\mathbf{x} = \widehat{\mathbf{Q^T}}\mathbf{b}\end{gathered}$$

Because $\hat{\mathbf{R}}$ is a diagonal matrix,

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

equation solving shall be easier.

Using Single Value Decomposition

Because $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V^T}$, and $\mathbf{Ax} = \mathbf{Pb}$,

$$\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V^T}\mathbf{x} = \hat{\mathbf{U}}\widehat{\mathbf{U^T}}\mathbf{b}$$
$$\hat{\mathbf{\Sigma}}\mathbf{V^T}\mathbf{x} = \widehat{\mathbf{U^T}}\mathbf{b}$$

Let $\mathbf{y} = \mathbf{V^T}\mathbf{x}$. Then,

$$\hat{\mathbf{\Sigma}}\mathbf{y} = \widehat{\mathbf{U^T}}\mathbf{b}$$

Once we obtain $\mathbf{y}$, we can calculate $\mathbf{x}$:

$$\mathbf{V^T}\mathbf{x} = \mathbf{y}$$
$$\mathbf{x} = (\mathbf{V^T})^{-\mathbf{1}}\mathbf{x}$$
$$\mathbf{x} = \mathbf{V}\mathbf{x}$$

[$\mathbf{V}$ is a orthogonal matrix, which means that $(\mathbf{V^T})^{-\mathbf{1}} = \mathbf{V}$]

Part (a)

For this part, we have been asked to use the Modified Gram-Schmidt method to calculate the least squares polynomial. The algorithm for Modified Gram-Schmidt is as follows:

```
for j = 1 → n
```

$$\boldsymbol{v_j^{(1)}} = \boldsymbol{a_j}$$

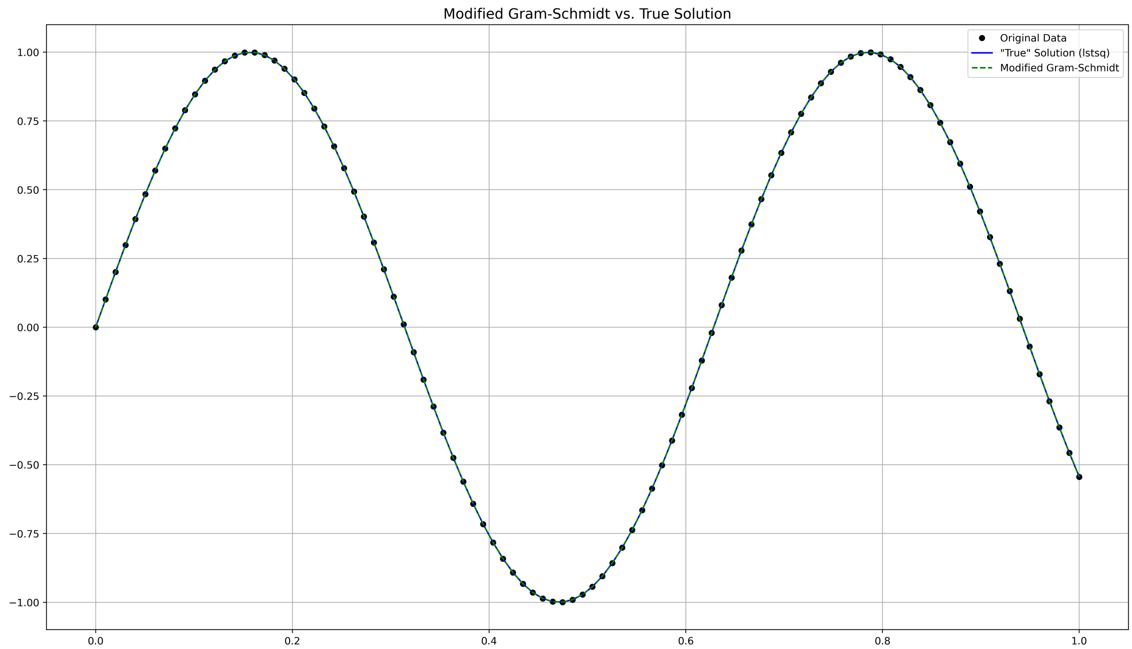
$$\boldsymbol{v_j^{(2)}} = \boldsymbol{P_{\perp q_1} a_j} = \boldsymbol{v_j^{(1)}} - \boldsymbol{q_1 q_1^T v_j^{(1)}}$$

$$\boldsymbol{v_j^{(3)}} = \boldsymbol{P_{\perp q_2} a_j} = \boldsymbol{v_j^{(2)}} - \boldsymbol{q_2 q_2^T v_j^{(2)}}$$

$$\vdots$$

$$\boldsymbol{v_j^{(j)}} = \boldsymbol{v_j^{(j-1)}} - \boldsymbol{q_{j-1} q_{j-1}^T v_j^{(j-1)}}$$

The graph obtained is as follows:



Part (b)

For this part, we have been asked to use the Householder Triangularization method to calculate the least squares polynomial. The algorithm for Householder Triangularization is as follows:

for k = 1 → n

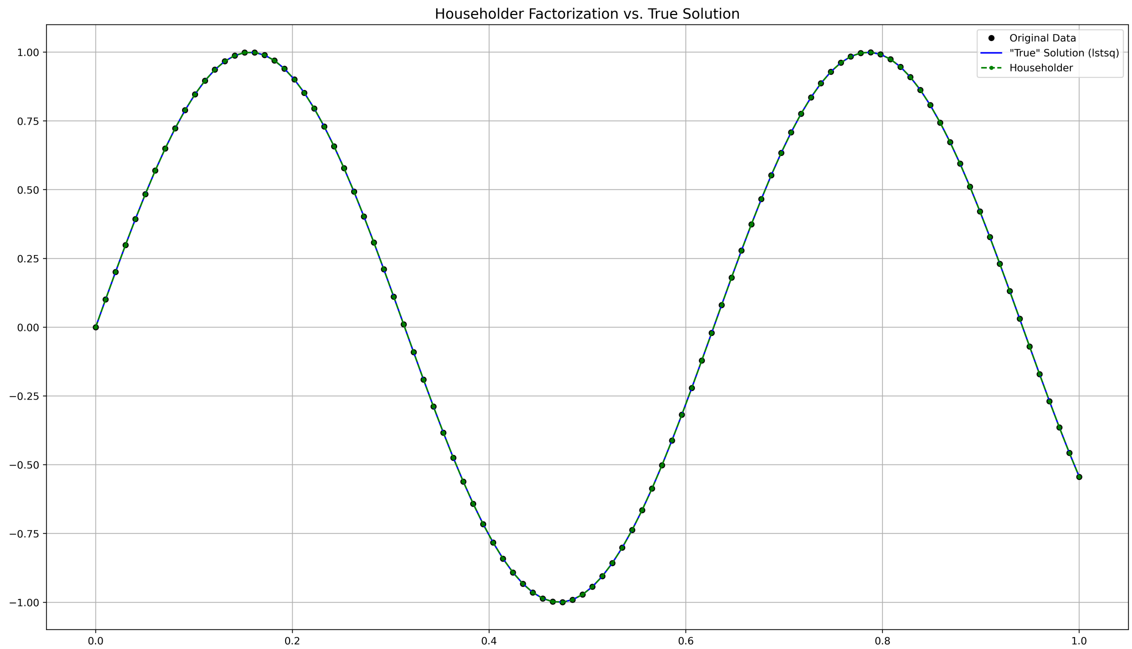
$$x = A(k{:}m, k)$$

$$v_k = sgn(x_1) \times \left||x|\right| \times e_1 + x$$

$$v_k = \frac{v_k}{\left||v_k|\right|_2} \qquad \text{//normalisation of } V_k$$

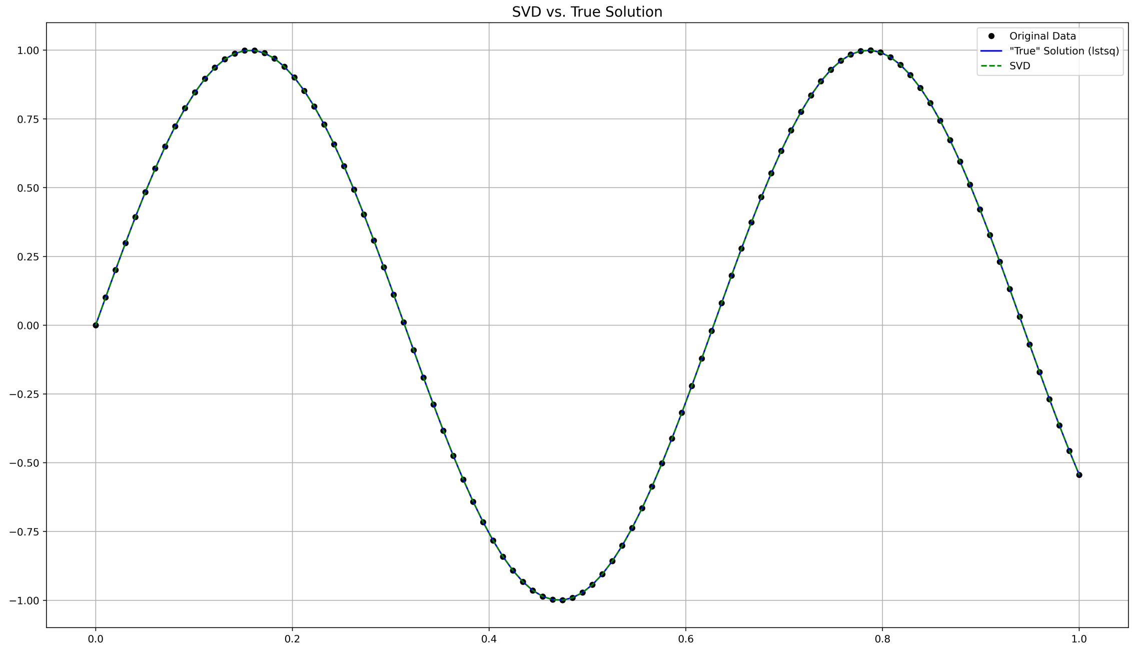
$$A(k{:}m, k{:}n) = A(k{:}m, k{:}n) - 2v_k v_k^T A(k{:}m, k{:}n)$$

The graph obtained is as follows:



Part (c)

For SVD, I used the input function in Python. The graph obtained is as follows:

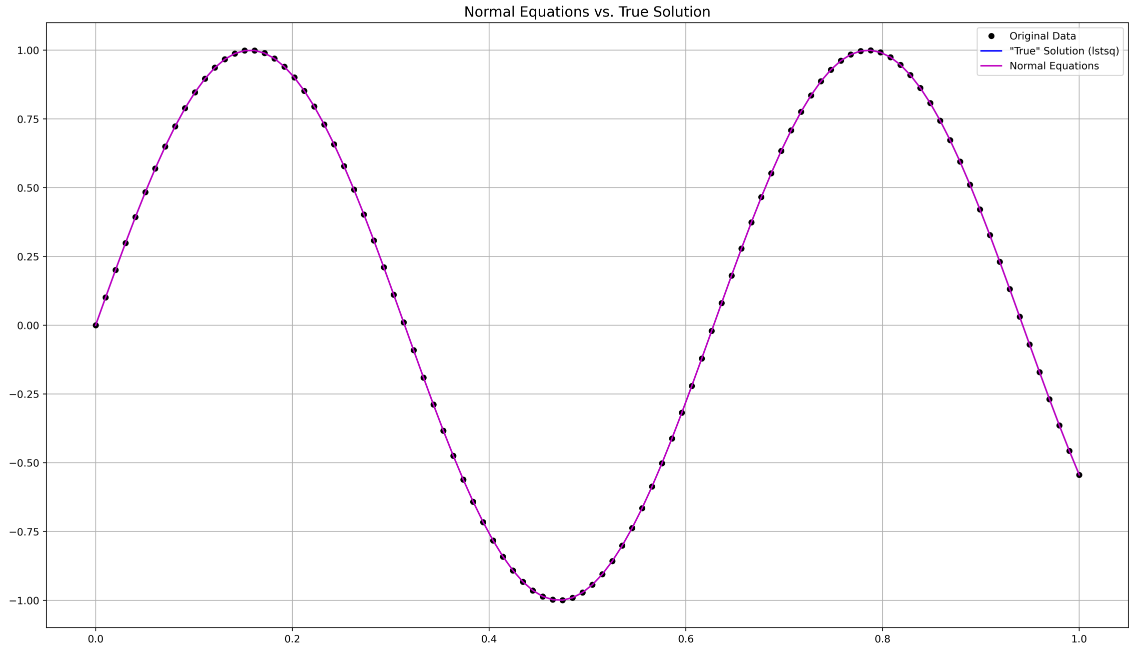


Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: September 29, 2025

Assignment No: Assignment 4
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

Part (d)

For this method, I simply solved the normal equation, i.e. $\mathbf{A^T A \hat{x}} = \mathbf{A^T b}$, using Python's in-built method. The graph obtained is as follows:



Algorithm, Code and Output

The algorithm is as follows:

1. Generate the domain and data points for $f(t) = \sin(10t)$.
2. Obtain the Vanderlinde matrix $\mathbf{A}$ using the `np.vander()` function.
3. Calculate the "true" least squares curve using Python's inbuilt method, i.e. `np.linalg.lstsq()`.
4. Calculate the least squares curve using Modified Gram Schmidt using the algorithm in part (a).
5. Calculate the least squares curve using Householder Trianglurization using the algorithm in part (b).
6. Calculate the least squares curve using Single Value Decomposition using Python's inbuilt method, i.e. `np.linalg.svd()`.
7. Calculate the least squares curve by solving the Normal equation using Python's inbuilt method, i.e. `np.linalg.solve()`.
8. Plot and save each method against the "true" solution.
9. Calculate and output the residual error, $||\mathbf{r}||_2 = ||\mathbf{b} - \mathbf{Ax}||_2$

The Python code which implements the above algorithm is as follows:

```
import numpy as np
import matplotlib.pyplot as plt
import os

```

```python
def f(t):
    """The function to be approximated."""
    return np.sin(10 * t)

def back_substitution(R, y):
    """
    Solves the upper triangular system Rx = y using back substitution.
    """
    n = R.shape[1]
    x = np.zeros(n)
    for i in range(n - 1, -1, -1):
        x[i] = (y[i] - np.dot(R[i, i+1:], x[i+1:])) / R[i, i]
    return x

def modified_gram_schmidt(A):
    """Performs QR factorization using the Modified Gram-Schmidt process."""
    m, n = A.shape
    Q = np.zeros((m, n))
    R = np.zeros((n, n))
    V = A.copy()

    for k in range(n):
        R[k, k] = np.linalg.norm(V[:, k])
        Q[:, k] = V[:, k] / R[k, k]
        for j in range(k + 1, n):
            R[k, j] = np.dot(Q[:, k].T, V[:, j])
            V[:, j] = V[:, j] - R[k, j] * Q[:, k]
    return Q, R

def householder_factorization(A):
    """Performs QR factorization using Householder reflections."""
    m, n = A.shape
    R = A.copy()
    Q = np.eye(m)
    for k in range(n):
        x = R[k:, k]
        e1 = np.zeros_like(x)
        e1[0] = np.copysign(np.linalg.norm(x), x[0])
        v = (x + e1)
        v = v / np.linalg.norm(v)

        R[k:, :] -= 2 * np.outer(v, v.T @ R[k:, :])
        Q[:, k:] -= 2 * Q[:, k:] @ np.outer(v, v.T)

    return Q[:,:n], R[:n,:]

# --- 1. Generate Data and Vandermonde Matrix ---
m = 100
degree = 14
domain = np.linspace(0, 1, m)
data_points = f(domain)
A = np.vander(domain, degree + 1)

# --- 2. Solve the Least Squares Problem using all 5 methods ---

# Inbuilt Python "True" Solution (Baseline for comparison)
```

```
true_coeffs = np.linalg.lstsq(A, data_points, rcond=None)[0]
print("Coefficients using np.linalg.lstsq ('True' Solution) calculated.")

# Method (a): Modified Gram-Schmidt
Q_mgs, R_mgs = modified_gram_schmidt(A)
mgs_coeffs = back_substitution(R_mgs, Q_mgs.T @ data_points)
print("Coefficients using Modified Gram-Schmidt calculated.")

# Method (b): Householder Factorization
Q_hh, R_hh = householder_factorization(A)
householder_coeffs = back_substitution(R_hh, Q_hh.T @ data_points)
print("Coefficients using Householder Factorization calculated.")

# Method (c): SVD
U, S, VT = np.linalg.svd(A, full_matrices=False)
svd_coeffs = VT.T @ ((U.T @ data_points) / S)
print("Coefficients using SVD calculated.")

# Method (d): Normal Equations
A_T_A = A.T @ A
A_T_b = A.T @ data_points
normal_eq_coeffs = np.linalg.solve(A_T_A, A_T_b)
print("Coefficients using Normal Equations calculated.")

# --- 3. Plot and Save Individual Comparisons ---

# Directory to save plots. Change this if you get a PermissionError.
save_dir = r"H:\My Drive\Numerical Linear Algebra\Assignments\Assignment 4"
print(f"\nAttempting to save plots to: {save_dir}")

# Plot 1: Modified Gram-Schmidt vs. True Solution
plt.figure(figsize=(19.2, 10.8))  # 4K resolution size
plt.title('Modified Gram-Schmidt vs. True Solution', fontsize=14)
plt.plot(domain, data_points, 'ko', markersize=5, label='Original Data')
plt.plot(domain, np.polyval(true_coeffs, domain), 'b-', label='"True" Solution
↪ (lstsq)')
plt.plot(domain, np.polyval(mgs_coeffs, domain), 'g--', label='Modified Gram-Schmidt')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(save_dir, 'mgs_comparison.png'), dpi=400) # Added dpi=400 for
↪ 4K resolution
#plt.show()

# Plot 2: Householder vs. True Solution
plt.figure(figsize=(19.2, 10.8))  # 4K resolution size
plt.title('Householder Factorization vs. True Solution', fontsize=14)
plt.plot(domain, data_points, 'ko', markersize=5, label='Original Data')
plt.plot(domain, np.polyval(true_coeffs, domain), 'b-', label='"True" Solution
↪ (lstsq)')
plt.plot(domain, np.polyval(householder_coeffs, domain), 'g--.', label='Householder')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(save_dir, 'householder_comparison.png'), dpi=400) # Added
↪ dpi=400
#plt.show()

```

```python
# Plot 3: SVD vs. True Solution
plt.figure(figsize=(19.2, 10.8))  # 4K resolution size
plt.title('SVD vs. True Solution', fontsize=14)
plt.plot(domain, data_points, 'ko', markersize=5, label='Original Data')
plt.plot(domain, np.polyval(true_coeffs, domain), 'b-', label='"True" Solution (lstsq)')
plt.plot(domain, np.polyval(svd_coeffs, domain), 'g--', label='SVD')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(save_dir, 'svd_comparison.png'), dpi=400) # Added dpi=400
#plt.show()

# Plot 4: Normal Equations vs. True Solution
plt.figure(figsize=(19.2, 10.8))  # 4K resolution size
plt.title('Normal Equations vs. True Solution', fontsize=14)
plt.plot(domain, data_points, 'ko', markersize=5, label='Original Data')
plt.plot(domain, np.polyval(true_coeffs, domain), 'b-', label='"True" Solution (lstsq)')
plt.plot(domain, np.polyval(normal_eq_coeffs, domain), 'm-', label='Normal Equations')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(save_dir, 'normal_equations_comparison.png'), dpi=400) # Added dpi=400
#plt.show()

print(f"\nSuccessfully saved 4 high-resolution plots to the '{save_dir}' directory.")

#Get least square errors for all methods
true_error = np.linalg.norm(data_points - np.polyval(true_coeffs, domain))
mgs_error = np.linalg.norm(data_points - np.polyval(mgs_coeffs, domain))
householder_error = np.linalg.norm(data_points - np.polyval(householder_coeffs, domain))
svd_error = np.linalg.norm(data_points - np.polyval(svd_coeffs, domain))
normal_eq_error = np.linalg.norm(data_points - np.polyval(normal_eq_coeffs, domain))

print("\nLeast Squares Errors:")
print(f"True Solution (lstsq): {true_error:.6e}")
print(f"Modified Gram-Schmidt: {mgs_error:.6e}")
print(f"Householder: {householder_error:.6e}")
print(f"SVD: {svd_error:.6e}")
print(f"Normal Equations: {normal_eq_error:.6e}")
```

The output of the code is as follows:

```
Coefficients using np.linalg.lstsq ('True' Solution) calculated.
Coefficients using Modified Gram-Schmidt calculated.
Coefficients using Householder Factorization calculated.
Coefficients using SVD calculated.
Coefficients using Normal Equations calculated.

Attempting to save plots to: H:\My Drive\Numerical Linear Algebra\Assignments\Assignment 4

Successfully saved 4 high-resolution plots to the 'H:\My Drive\Numerical Linear Algebra\Assignments\Assignment 4' directory.

Least Squares Errors:
```

```
True Solution (lstsq): 2.116652e-06
Modified Gram-Schmidt: 2.118126e-06
Householder: 2.116652e-06
SVD: 2.116652e-06
Normal Equations: 2.147932e-04
```

Results

Method	$\|\|\mathbf{r}\|\|_2 = \|\|\mathbf{b} - \mathbf{A}\mathbf{x}\|\|_2$
“True” solution	2.117×10^{-6}
Modified Gram Schmidt	2.118×10^{-6}
Householder	2.117×10^{-6}
SVD	2.117×10^{-6}
Normal Equation	2.148×10^{-4}

From the results, we can see that the Modifed Gram-Schmidt, Householder and SVD are quite close to the true solution. Normal equations, on the other hand result in error much larger, by almost a magnitude of 100.

We can also see the that Householder and SVD, both result in errors almost exactly equal to the true solution, whereas the Modified Gram Schmidt gave a slightly larger error. This error, however, is comparable to the true error.

Suhas Kumath

25945

Numerical Linear Algebra - Assignment 4

7) a)

True. The proof is as follows:

Let λ be an eigenvalue of $\underset{\sim}{A}$. We hence know there exists an $\underset{\sim}{x}$ s.t.

$$\underset{\sim}{A}\underset{\sim}{x} = \lambda \underset{\sim}{x}$$

We want to show that $(\underset{\sim}{A} - \mu \underset{\sim}{I})\hat{\underset{\sim}{x}} = (\lambda - \mu)\hat{\underset{\sim}{x}}$

$$\text{L.H.S.} = (\underset{\sim}{A} - \mu \underset{\sim}{I})\underset{\sim}{x} = \underset{\sim}{A}\underset{\sim}{x} - \mu \underset{\sim}{I}\underset{\sim}{x} \quad \text{[Opening the bracket]}$$

$$= \underset{\sim}{A}\underset{\sim}{x} - \mu \hat{\underset{\sim}{x}} \quad [\underset{\sim}{I}\hat{\underset{\sim}{x}} = \hat{\underset{\sim}{x}}]$$

$$= \lambda \underset{\sim}{x} - \mu \underset{\sim}{x} \quad [\underset{\sim}{A}\underset{\sim}{x} = \lambda \underset{\sim}{x}]$$

$$= (\lambda - \mu)\underset{\sim}{x} \quad [\text{Factorise w.r.t. } \underset{\sim}{x}]$$

Hence proven, L.H.S. = R.H.S..

7) b)

False. The counter-example is as follows:

Consider a 2×2 matrix $\underset{\sim}{A} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$

Let us calculate the eigen values of $\underset{\sim}{A}$:

$$|\underset{\sim}{A} - \lambda \underset{\sim}{I}| = 0$$

$$\begin{vmatrix} 1-\lambda & 1 \\ 0 & 2-\lambda \end{vmatrix} = 0$$

$$(1-\lambda)(2-\lambda) = 0$$

$$\lambda = 1, 2$$

If the statement is true, $\lambda = -1$ & $\lambda = -2$ must also be an eigenvalue. However, we just calculated both eigenvalues of $\underset{\sim}{A}$ and neither -1 nor -2 belong to the calculated values. Hence neither -1 nor -2 can be eigenvalues of $\underset{\sim}{A}$. This itself proves the statement to be false.

7) c) True. The proof is as follows:

Let $\underset{\sim}{A}$ be a real matrix, and λ be its complex-valued eigenvalue.

Then, $\underset{\sim}{A}\underset{\sim}{x} = \lambda \underset{\sim}{x}$

$\overline{(\underset{\sim}{A}\underset{\sim}{x})} = \overline{(\lambda \underset{\sim}{x})}$ [Take conjugate of both sides]

$\bar{\underset{\sim}{A}}\,\bar{\underset{\sim}{x}} = \bar{\lambda}\,\bar{\underset{\sim}{x}}$ [Property of conjugate : $\overline{(M N)} = \bar{M} \cdot \bar{N}$]

$\underset{\sim}{A}\,\bar{\underset{\sim}{x}} = \lambda^{*}\,\bar{\underset{\sim}{x}}$ [$\bar{A} = A$ because A is a real matrix]

Because $\underset{\sim}{x}$ is an eigenvector $\underset{\sim}{x} \neq \underset{\sim}{0}$. Therefore $\bar{\underset{\sim}{x}}$ also cannot be zero vector.

Hence proven, if λ is an eigenvalue of $\underset{\sim}{A}$, λ^{*} will also be an eigenvalue of $\underset{\sim}{A}$.

7) d) True. The proof is as follows:

We need to first prove that $\lambda^{-1} = 1/\lambda$ exists. For this, λ cannot be equal to 0.

We know that $\lambda \neq 0$. This is because $\underset{\sim}{A}$ is non-singular, i.e full rank. We also know that no. of non-zero eigenvalues of $\underset{\sim}{A}$ = rank of $\underset{\sim}{A}$.

Another way to prove it is as follows: (Proof by contradiction)

Let $\underset{\sim}{A}\underset{\sim}{x} = \lambda \underset{\sim}{x}$, then

$\det(\underset{\sim}{A} - \lambda \underset{\sim}{I}) = 0$

Let us take $\lambda = 0$:

$$\det(\underset{\sim}{A} - 0 \cdot \underset{\sim}{I}) = 0$$

$$\det(\underset{\sim}{A}) = 0$$ $\Rightarrow$ We know this cannot be true because A is given to be non-singular.

Let λ be the eigen value of $\underset{\sim}{A}$. Then:

$$\underset{\sim}{A}\underset{\sim}{x} = \lambda \underset{\sim}{x}$$

$$\underset{\sim}{A}^{-1}\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{A}^{-1}\lambda\underset{\sim}{x}$$ [Left-multiply both sides by $\underset{\sim}{A}^{-1}$]

$$\underset{\sim}{x} = \lambda \underset{\sim}{A}^{-1}\underset{\sim}{x}$$ [$\underset{\sim}{A}^{-1} \cdot \underset{\sim}{A} = \underset{\sim}{I}$ and $\underset{\sim}{I}\underset{\sim}{x} = \underset{\sim}{x}$]

$$\underset{\sim}{A}^{-1}\underset{\sim}{x} = \frac{1}{\lambda}\underset{\sim}{x}$$ [Divide both sides by λ]

$$\underset{\sim}{A}^{-1}\underset{\sim}{x} = \lambda^{-1}\underset{\sim}{x}$$

Hence proven, λ^{-1} is the eigen value of $\underset{\sim}{A}^{-1}$, as long as λ is an eigenvalue of $\underset{\sim}{A}$ and $\underset{\sim}{A}$ is non-singular.

7) e) False. Counter-example is as follows:

Let $\underset{\sim}{A} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

The eigenvalues of A are $0, 0$, whereas not all entries of $\underset{\sim}{A}$ are 0.

$$|A - \lambda I| = 0$$

$$\begin{vmatrix} -\lambda & 1 \\ 0 & -\lambda \end{vmatrix} = 0$$

$$(-\lambda)^2 = 0$$

$$\lambda = 0, 0$$

7) f)

True. The proof is as follows:

Because $\underset{\sim}{A}$ is diagonalisable, $\underset{\sim}{A}$ can be expressed as:

$\underset{\sim}{A} = \underset{\sim}{P}\,\underset{\sim}{D}\,\underset{\sim}{P}^{-1}$, where $\underset{\sim}{D}$ is a diagonal matrix, and $\underset{\sim}{P}$ is an invertible matrix.

The diagonal entries of $\underset{\sim}{D}$ are the eigenvalues of $\underset{\sim}{A}$. Because all eigenvalues of $\underset{\sim}{A}$ are given to be λ, we can say that:

$$\underset{\sim}{D} = \begin{bmatrix} \lambda & & 0 \\ & \lambda \ddots & \\ 0 & & \lambda \end{bmatrix} = \lambda \begin{bmatrix} 1 & & 0 \\ & 1 \ddots & \\ 0 & & 1 \end{bmatrix} = \lambda \underset{\sim}{I},$$

where $\underset{\sim}{I}$ is the identity matrix.

Substituting $\underset{\sim}{D} = \lambda \underset{\sim}{I}$ into the original equation,

$$\begin{aligned} \underset{\sim}{A} = \underset{\sim}{P}\,\underset{\sim}{D}\,\underset{\sim}{P}^{-1} &= \underset{\sim}{P}(\lambda \underset{\sim}{I})\underset{\sim}{P}^{-1} && [\underset{\sim}{D} = \lambda \underset{\sim}{I}] \\ &= \lambda(\underset{\sim}{P}\,\underset{\sim}{I}\,\underset{\sim}{P}^{-1}) && [\text{Re-arrange to take scalar out}] \\ &= \lambda(\underset{\sim}{P}\,\underset{\sim}{P}^{-1}) && [\underset{\sim}{P}\,\underset{\sim}{I} = \underset{\sim}{P}] \\ &= \lambda \underset{\sim}{I} && [\underset{\sim}{P}\,\underset{\sim}{P}^{-1} = \underset{\sim}{I}] \end{aligned}$$

Hence, because $\underset{\sim}{A} = \lambda \underset{\sim}{I}$, and I is diagonal, we know that $\underset{\sim}{A}$ is also diagonal.