Homework – 3

1) $A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$ $A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{bmatrix}$

a) Find orthogonal projector onto range (A_1)

Orthogonal projector P onto the range (A_1) is given by

$$P = A_1 A_1^*$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ The obtained P is a orthgonal projector (Since $P^2 = P$)

$$I - P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

is also orthgonal projector

$\therefore$ P and $I - P$ are orthogonal projectors onto range (A_1)

b) Find the orthogonal projector P onto range (A_2)

The orthogonal projector P onto range (A_2) is given by

$$P = A_2 (A_2^* A_2)^{-1} A_2^*$$

Finding, $A_2^* A_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}$

$$\Rightarrow A_2^* A_2 = \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix}$$

$$(A_2^* A_2)^{-1} = \frac{1}{10-4} \begin{pmatrix} 5 & -2 \\ -2 & 2 \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 5 & -2 \\ -2 & 2 \end{pmatrix}$$

$$A_2 (A_2^* A_2)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 5 & -2 \\ -2 & 2 \end{bmatrix} \frac{1}{6} = \frac{1}{6} \begin{bmatrix} 5 & -2 \\ -2 & 2 \\ 1 & 2 \end{bmatrix}$$

$$\therefore P = \frac{1}{6} \begin{bmatrix} 5 & -2 \\ -2 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 5 & -2 & 1 \\ -2 & 2 & 2 \\ 1 & 2 & 5 \end{bmatrix}$$

$\therefore$ The obtained P is the orthogonal projector onto range (A_2)

Question 2A

```
function [Q, R] = mgs(A)

[m, n] = size(A);

Q = zeros(m, n);
R = zeros(n, n);

v = A;

for i = 1:n
    R(i,i) = norm(v(:,i));
    Q(:,i) = v(:,i) / R(i,i);

    for j = i+1:n
        R(i,j) = conj(Q(:,i))' * v(:,j);
        v(:,j) = v(:,j) - R(i,j) * Q(:,i);
    end
end

Q, R
end
```

```
Not enough input arguments.

Error in mgs (line 3)
[m, n] = size(A);
```

Published with MATLAB® R2022b

```
function X = backward_substitution(R, b)
    n = size(R, 1);

    X = zeros(1, n);

    X(n) = b(n) / R(n, n);

    for i = n - 1:-1:1
        sum_ = 0;

        for j = i + 1:n
            sum_ = sum_ + R(i, j) * X(j);
        end

        X(i) = (b(i) - sum_) / R(i, i);

    end
```

```
Not enough input arguments.

Error in backward_substitution (line 2)
    n = size(R, 1);
```

Published with MATLAB® R2022b

Question 2B

```
A = [1 0; 0 1; 1 2];

b = [1 1 1]';

[Q, R] = mgs(A);

RHS = Q.' * b;
x = backward_substitution(R, RHS);

x
```

```
Q =

    0.7071   -0.5774
         0    0.5774
    0.7071    0.5774


R =

    1.4142    1.4142
         0    1.7321


x =

    0.6667    0.3333
```

Question 3A

```
function [V, R] = house(A)

    [m, n] = size(A);
    V = zeros(m, n);

    for k = 1:n
        x = A(k:m, k);
        e = [1; zeros(length(x) - 1, 1)];

        if x(1) ~= 0
            S = sign(x(1));
        else
            S = 1;
        end

        v = S * norm(x) * e + x;
        v = v / norm(v);
        A(k:m, k:n) = A(k:m, k:n) - 2 * v * (v' * A(k:m, k:n));
        V(k:m, k) = v;

    end

    R = A;

    V, R;

end
```

```
Not enough input arguments.

Error in house (line 3)
    [m, n] = size(A);
```

Published with MATLAB® R2022b

Question 3B

```
function Q = formQ(V)
    [m, n] = size(V);


    for k = n:-1:1
        I(k:m, k:m);
        I(k:m, k:m) = I(k:m, k:m) - 2 * V(k:m, k) * (V(k:m, k)' * I(k:m, k:m));
    end

    Q = I;
    Q;

end
```

```
Not enough input arguments.

Error in formQ (line 2)
    [m, n] = size(V);
```

Question 3C

```
A = [1 0; 0 1; 1 2];

[V, R] = house(A);
R

Q = formQ(V);

Q
```

```
V =

    0.9239         0
         0    0.8881
    0.3827    0.4597


R =

   -1.4142   -1.4142
         0   -1.7321
    0.0000         0


Q =

   -0.7071    0.5774   -0.4082
         0   -0.5774   -0.8165
   -0.7071   -0.5774    0.4082
```

Published with MATLAB® R2022b

Question 4

```
function A = VanderMonde(x, n)
    x = x(:);
    A = ones(length(x), n);

    for i = 2:n
        A(:, i) = x .* A(:, i - 1);
    end

end
```

```
Not enough input arguments.

Error in VanderMonde (line 3)
    x = x(:);
```

```
function x = lsp(b, Q, R)

    [temp] = Q' * b;
    x = zeros(size(temp, 1), 1);
    x = inv(R) * temp;
end
```

```
Not enough input arguments.

Error in lsp (line 3)
    [temp] = Q' * b;
```

Published with MATLAB® R2022b

```
format long

t = linspace(0, 1, 50);


A = VanderMonde(cos(4*t), 10);

b = zeros(50, 1);
x = zeros(10, 1);

% build the vector b
for i = 1:50
    b(i) = cos(4 * t(i));
end

% (a)
Ac = A' * A;
bc = A' * b;
L = chol(Ac, "lower");
y = L \ bc;
xa = L' \ y;

xa

% (b)
[Q, R] = mgs(A);
xb = (lsp(b, Q, R))';

xb

% (c)
[V, R] = house(A);
Q = formQ(V);
bh = Q' * b;
xc = R \ bh;

xc

% (d)
xd = R \ (Q' * b);

xd
```

```
xa =

  -0.000000000000056
   0.999999999998957
   0.000000000001646
   0.000000000013290
  -0.000000000007876
  -0.000000000046570
   0.000000000012180
   0.000000000060982
  -0.000000000005922
  -0.000000000026745


xb =
```

```
  Columns 1 through 3

 -0.000000000000001   0.999999999999492   0.000000000000050

  Columns 4 through 6

  0.000000000006089  -0.000000000000319  -0.000000000020318

  Columns 7 through 9

  0.000000000000535   0.000000000025631  -0.000000000000265

  Column 10

 -0.000000000010922


xc =

 -0.000000000000000
  0.999999999999999
  0.000000000000001
  0.000000000000006
 -0.000000000000004
 -0.000000000000016
  0.000000000000003
  0.000000000000015
  0.000000000000000
 -0.000000000000004


xd =

 -0.000000000000000
  0.999999999999999
  0.000000000000001
  0.000000000000006
 -0.000000000000004
 -0.000000000000016
  0.000000000000003
  0.000000000000015
  0.000000000000000
 -0.000000000000004
```

Comment:
1. Here xc and xd are same.
2. Here highlighted part is zero or almost zero (0). So we need more than 16 digit to get more acurate value.

Published with MATLAB® R2022b