Experiment: 6
Gram Schmidt and QR Factorization

1. Gram Schmidt Process

Let V be any vector space and $\{v_1, v_2, \ldots, v_n\} \subseteq V$. Now we want to convert this n-vectors into orthonormal vectors using Gram Schimdt process.

(a) Convert the vectors $\left\{\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\-2\end{bmatrix}\right\}$into orthonormal vectors.

(b) Convert the vectors into $\left\{\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\-1\end{bmatrix}\right\}$ into orthonormal vectors.

Algorithm

// Output: Matrix B

- Define a matrix A whose columns are $v_1, v_2, \ldots, v_n$ ($A = [v_1\ v_2\ ..\ v_n]$)
- if the rank of A is not equal to n.
  Display the given vectors are linearly dependent
  else
  Define an empty array B in which you want to store the orthonormal vectors
- Set $v =$ first column of A and $u = \frac{v}{||v||}$
- Add u into B as first column.
- for i from 2 to n
  Set v=zeros(n,1), parlv=0, u =zeros(n,1)
  for j form 1 to i-1
  parlv=parlv+$(B(:,j)' * A(:,i)) * B(:,j)$
  end for
  $v = A(:,i) -$ parlv
  $u = \frac{v}{||v||}$
  Add u into B as a column.
  end for
- Display the matrix B.

```
v1=[2 1]';
v2=[2 -2 ]';
A=[v1 v2];
[m,n]=size(A);
B=[];
```

```
v=A(:,1);
u=v(:,1)./norm(v(:,1),2);
B(:,end+1)=u;
for i=2:n
    parlv=0;
    v=zeros(n,1);
    u=zeros(n,1);
    for j=1:i-1
        parlv=parlv+(B(:,j)'*A(:,i)).*B(:,j);
    end
    v=A(:,i)-parlv;
    u=v/norm(v,2);
    B(:,end+1)=u;
end
format default

disp(B)
```

```
    0.8944    0.4472
    0.4472   -0.8944
```

```
v1=[1 0 0]';
v2=[1 1 1]';
v3=[1 1 -1]';
A=[v1 v2 v3];
[m,n]=size(A);
B=[];
v=A(:,1);
u=v(:,1)./norm(v(:,1),2);
B(:,end+1)=u;
for i=2:n
    parlv=0;
    v=zeros(n,1);
    u=zeros(n,1);
    for j=1:i-1
        parlv=parlv+(B(:,j)'*A(:,i)).*B(:,j);
    end
    v=A(:,i)-parlv;
    u=v/norm(v,2);
    B(:,end+1)=u;
end
format default
```

```
disp(B)
```

```
    1.0000         0         0
         0    0.7071    0.7071
         0    0.7071   -0.7071
```

2. QR factorization using Gram–Schmidt process

Implement this algorithm as MATLAB function script.

(a) Find the QR factorization of following matrix

$$\begin{bmatrix} 1 & -1 & 1 \\ -2 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

Algorithm

// Input: Matrix A

// Output: Matrix Q and R.

- Create a function which take the matrix A as input
- Define dimension of matrix A(say $m \times n$)
- if rank of A is not equal to n
  Factorization does not exists
  return to main script
  else
  Define an empty array Q
  and a null matrix R of size $n \times n$
- Set $v =$ first column of A and $u = \frac{v}{||v||}$
- Add u into Q as first column.
- for i from 2 to n
  Set v=zeros(n,1), parlv=0, u =zeros(n,1)
  for j form 1 to i-1
  parlv=parlv+$(Q(:,j)' * A(:,i)) * Q(:,j)$
  end for
  $v = A(:,i) -$ parlv
  $u = \frac{v}{||v||}$
  Add u into Q as a column.
  end for
- for i from 1 to n
  for j from i to n
  $R_{ij} =< Q(:,i), A(:,j) >$
  end for
  end for

Solution.

```
function [Q,R]=QR_fact(A)
[~,n]=size(A);
if rank(A)~=n
    disp("Factorization does not exists")
    return;
else

Q=[];
R=zeros(n);
v=A(:,1);
u=v(:,1)./norm(v(:,1),2);
Q(:,end+1)=u;
for i=2:n
    parlv=0;
    v=zeros(n,1);
    u=zeros(n,1);
    for j=1:i-1
        parlv=parlv+(Q(:,j)'*A(:,i)).*Q(:,j);
    end
    v=A(:,i)-parlv;
    u=v/norm(v,2);
    Q(:,end+1)=u;
end
for i=1:n
    for j=1:n
        R(i,j)=round(Q(:,i)'*A(:,j));
    end
end
format rational
end
```

```
 A=[1 -1 1;-2 1 0;2 0 1];
[Q,R]=QR_fact(A)
```

```
    Q =
       1/3          -2/3          2/3
      -2/3           1/3          2/3
       2/3           2/3          1/3
R =
       3            -1            1
       0             1            0
       0             0            1
```

■