Assignment # 3

Yu-Sheng Lo
YL25288

Problem 1

$A \in \mathbb{R}^{m \times n}$ and full rank, $P_j = q_j q_j^*$

$$A = \begin{bmatrix} | & | & | & & | \\ a_1 & a_2 & a_3 & \dots & a_n \\ | & | & | & & | \end{bmatrix}, \quad \{a_i\}_{i=1}^n \in \mathbb{R}^m$$

Classical Gram-Schmidt

$$\begin{aligned} v_1 &= a_1 \to q_1 = \frac{v_1}{\|v_1\|_2} \\ v_2 &= a_2 - q_1 q_1^* a_2 \to q_2 = \frac{v_2}{\|v_2\|_2} \\ &\vdots \\ &\vdots \\ v_i &= a_i - \sum_{j=1}^{i-1} q_j q_j^* a_i \to q_i = \frac{v_i}{\|v_i\|_2} \\ v_i &= (I - \sum_{j=1}^{i-1} P_j) A(:, i) \quad (1) \end{aligned}$$

Modified Gram-Schmidt

$$\begin{aligned} w_i &= [\prod_{j=1}^{i-1} (I - P_j)] A(:, i) \quad (2) \\ q_i &= \frac{w_i}{\|w_i\|_2} \end{aligned}$$

From equations (1) and (2),in order to show these two algorithm are identical,we need to show

$$\begin{aligned} v_i &= w_i \\ \prod_{j=1}^{i-1} (I - P_j) &= (I - \sum_{j=1}^{i-1} P_j) \quad , i = 1 \sim n \end{aligned}$$

Claim: $\prod_{j=1}^{i-1} (I - P_j) = (I - \sum_{j=1}^{i-1} P_j) \quad , i = 1 \dots n$
where$P_j = q_j q_j^*$,and $\{q_j\}_{j=1}^n$ are orthonormal

for i=1 and 2, the claim is obviously true.
for i=3

$$
\begin{aligned}
\prod_{j=1}^{2}(I-P_j) &= (I-P_1)(I-P_2) \\
&= I - P_2 - P_1 + P_1P_2 \\
&= I - (P_1+P_2) + q_1q_1^*q_2q_2^* \\
(q_i^*q_j = 0, \forall i \neq j) \rightarrow \quad &= I - (P_1+P_2) \\
&= I - \sum_{i=1}^{2} P_j
\end{aligned}
$$

The claim is true for i=3
Assume the claim is true for i=n-1, we have

$$
\prod_{j=1}^{n-2}(I-P_j) = I - (\sum_{i=1}^{n-2} P_j) \quad (3)
$$

for i=n

$$
\begin{aligned}
\prod_{j=1}^{n-1}(I-P_j) &= [\prod_{j=1}^{n-2}(I-P_j)](I-P_{n-1}) \\
(3) \rightarrow \quad &= [I - \sum_{i=1}^{n-2} P_j](I-P_{n-1}) \\
&= I - P_{n-1} - (\sum_{j=1}^{n-2} P_j) + [(\sum_{j=1}^{n-2} P_j)P_{n-1}]
\end{aligned}
$$

since $(\prod_{j=1}^{n-2} P_j)P_{n-1} = 0$

$$
\prod_{j=1}^{n-1}(I-P_j) = I - \sum_{j=1}^{n-1} P_j
$$

The claim is true for i=n, and by mathematical induction the claim is true.
The modified Gram-Schmidt algorithm is mathematically equivalent to the classical Gram-Schmidt algorithm #

Problem 2

The MATLAB code begins from page 6, function *gramschmidt* generates Q and R matrix.

Problem 3

A is constructed by MATLAB intrinsic function *gallery*
A=gallery('randsvd',100,kappa) creates matrix A $\in \mathbb{R}^{100\times 100}$ with cond(A)=kappa
A function *test* is created to test the following

- The orthogonality of Q by $e = \frac{\|Q^*Q-I\|_2}{\|I\|_2} = \|Q^*Q - I\|_2$
- The accuracy of QR factorization by $e = \frac{\|QR-A\|_2}{\|A\|_2}$
- Up-triangular of R by MATLAB intrinsic function *istriu*

The following is the summary. In Table.2, for each kappa, the error e is normalized with respect to *Classical Gram-Schmidt*

	Orthogonality			Accuracy			Up-triangular		
	Classical	*Modified*	*qr*	*Classical*	*Modified*	*qr*	*Classical*	*Modified*	*qr*
1	1.14E-15	1.08E-15	2.23E-15	5.87E-16	5.85E-16	1.8E-15	TRUE		
1000	1.08E-11	1.1E-13	2.31E-15	2.39E-16	2.1E-16	9.39E-16			
1E6	4.03E-5	5.88E-11	2.25E-15	1.7E-15	1.87E-16	8.36E-16			
1E9	1.05E+1	5.23E-8	2.48E-15	1.67E-16	1.93E-16	6.07E-16			

Table 1: non-Normalized

	Orthogonality			Accuracy			Up-triangular		
	Classical	*Modified*	*qr*	*Classical*	*Modified*	*qr*	*Classical*	*Modified*	*qr*
1	1	0.953	1.97	1	0.997	3.07	TRUE		
1000	1	0.0102	2.14E-4	1	0.877	3.92			
1E6	1	1.46E-6	5.58E-11	1	0.11	0.493			
1E9	1	4.97E-9	2.36E-16	1	1.16	4.01			

Table 2: Normalized

Discussion

- As the matrix becomes more ill-conditioned(as kappa increases), both Classical Gram-Schmidt and Modified Gram-Schmidt begins to lose orthogonality of Q, but qr remains stable.

- Classical Gram-Schmidt loses orthogonality by the order of $O(10^4)$, and modified Gram-Schmidt by the order of $O(10^2)$.

- Although all three algorithm gives good accuracy for each kappa, but once Q loses orthogonality, error is introduced when solving $Ax = b$. This is because we take advantage that $Ax = b \to QRx = b \to Rx = Q^*b$, but once Q begins to lose orthogonality, Q is no longer unitary, and so $Q^*Q \neq I$.

```
%%%%% Numerical Linear Algebra
%%%%% Homework #3
function [Q,R]=gramschmidt(A,flag)


% check input argument
if (nargin < 2)
    flag = false;
end

[m,n]=size(A);


% Classical G-S
if (flag == true)
    Q(:,1)=A(:,1);
    for j=1:n
        v(:,j)=A(:,j);
        for i=1:j-1
            R(i,j)=dot(Q(:,i)',A(:,j));
            v(:,j)=v(:,j)-R(i,j)*Q(:,i);
        end
        R(j,j)=norm(v(:,j));
        Q(:,j)=v(:,j)/R(j,j);
    end

% Modified G-S
else

    Q(:,1)=A(:,1);
    for i=1:n
        v(:,i)=A(:,i);
    end

    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)=dot(Q(:,i)',v(:,j));
            v(:,j)=v(:,j)-R(i,j)*Q(:,i);
        end
    end
end


return
end
```

```
% Test

function [orth,up,accu]=test(A,Q,R)

up=istriu(R); % is R up-triangular
accu=norm(A-Q*R)/norm(A); % the accuracy of QR factorization
orth=norm(Q'*Q-eye(size(A,2))); % the orthogonality


return
end
```