Report for Home Work 9:

PART I:

1. As it is pointed out in the slides, we have the following procedure for

Classical Gram-Schmidt (CGS):

$$Proj_e(u) = \frac{(u,e)}{||e||^2}e \quad (1)$$

Now let's take vectors *Vi* and project them orthogonally onto the line spanned by vectors *Ui*:

$$u_1 = v_1 \ ; \quad e_1 = \frac{u_1}{||u_1||}$$

$$u_2 = v_2 - Proj_{u_1}(v_2) \ ; \quad e_2 = \frac{u_2}{||u_2||}$$

$$u_3 = v_3 - Proj_{u_1}(v_3) - Proj_{u_2}(v_3) \ ; \quad e_3 = \frac{u_3}{||u_3||} \qquad (2)$$

$$\vdots$$

$$u_k = v_k - \sum_{j=1}^{k-1} Proj_{u_j}(v_k) \ ; \quad e_k = \frac{u_k}{||u_k||}$$

$u_1, u_2 .. u_k$ are the system of orthogonal vectors, and the normalized vectors are $e_1, e_2 .. e_k$.

- proof of orthogonality of each pair of *u* and orthonormality of each pair of *e*:

using the relation (1) and substituting u_2, one can confirm that (u_2, u_1)=0:

$$(u_2, u_1) = (v_2, u_1) - \frac{(v_2,u_1)}{(u_1,u_1)}(u_1, u_1) = 0 \quad (3)$$

The same is true for (u_3, u_1):

$$(u_3, u_1) = (v_3, u_1) - \frac{(v_3,u_1)}{(u_1,u_1)}(u_1, u_1) - \frac{(v_3,u_1)}{(u_2,u_2)}(u_2, u_1) = 0 \qquad (4)$$

The third term is zero as we have concluded in equation (4).

If then we assume that **any one** statement in the infinite sequence of statements is true (i.e. (u_{k-1}, u_1)), then so is the **next** one:

$$(u_k, u_1) = (v_k, u_1) - \frac{(v_k,u_1)}{(u_1,u_1)}(u_1, u_1) - \frac{(v_k,u_2)}{(u_2,u_2)}(u_2, u_1) - \frac{(v_k,u_3)}{(u_3,u_3)}(u_3, u_1) \ldots - \frac{(v_k,u_{k-1})}{(u_{k-1},u_{k-1})}(u_{k-1}, u_1) = 0$$

(5)

As mentioned in the last statement, as we know that terms three to end is zero, then we have (u_k, u_1) equal to zero. So by mathematical induction, we have concluded that (u_i, u_1) for all u_i is true. The exact same procedure holds for (u_i, u_2). We proceed by induction on *k*.

Suppose we know $(u_i, u_j) = 0$ for $1 \le i < j \le k-1$. Then for $i = k$,

$$(u_i, u_k) = (u_i, v_k) - \frac{(v_k,u_1)}{(u_1,u_1)}(u_i, u_1) - \frac{(v_k,u_2)}{(u_2,u_2)}(u_i, u_2) - \frac{(v_k,u_3)}{(u_3,u_3)}(u_i, u_3)\ldots - \frac{(v_k,u_{k-1})}{(u_{k-1},u_{k-1})}(u_i, u_{k-1}) = 0 \quad (6)$$

Since all the terms (u_i, u_j) are 0 except when $i = j$, and we have :

$$(u_i, u_k) = (u_i, v_k) - \frac{(v_k,u_i)}{(u_i,u_i)}(u_i, u_i) = 0 \quad (7)$$

Therefore, each pair of (u_i, u_j) are mutually orthogonal. As $u_1, u_2..u_k$ are nonzero orthogonal vectors, and we let $e_k = \frac{u_k}{||u_k||}$, then $e_1, e_2..e_k$ are orthonormal vectors. Orthonormal means that the vectors are orthogonal and that they each have length 1.

Modified Gram-Schmidt (MGS):

- proof of arithmetic equality of CGS and MGS:

$$u_1 = v_1 \ ; \quad e_1 = \frac{u_1}{||u_1||}$$

$$u_2 = v_2 - Proj_{u_1}(v_2) \ ; \quad e_2 = \frac{u_2}{||u_2||} \quad (8)$$

$$u_3{}^{(1)} = v_3 - Proj_{u_1}(v_3); \ u_3{}^{(2)} = u_3{}^{(1)} - Proj_{u_2}(u_3{}^{(1)}) \ ; \quad e_3 = \frac{u_3{}^{(2)}}{||u_3{}^{(2)}||}$$

It has to be shown that $u_3{}^{(2)}$ of MGS is exactly similar to $u_3$of CGS:

$$u_3{}^{(2)} = u_3{}^{(1)} - Proj_{u_2}(u_3{}^{(1)}) \ = v_3 - Proj_{u_1}(v_3) - Proj_{u_2}(v_3 - Proj_{u_1}(v_3)) \quad (9)$$

then

$$u_3{}^{(2)} = v_3 - Proj_{u_1}(v_3) - \frac{(v_3 - Proj_{u_1}(v_3), u_2)}{(u_2,u_2)}u_2 \quad (10)$$

the third term on RHS:

$$\frac{(v_3,u_2) - (Proj_{u_1}(v_3),u_2)}{(u_2,u_2)}u_2 = \frac{(v_3,u_2) - \frac{(v_3,u_1)}{(u_1,u_1)}(u_1,u_2)}{(u_2,u_2)}u_2 \quad (11)$$

As we know that u_1 and u_2 are orthogonal to each other, therefore $(u_1, u_2) = 0$. So,

$$u_3{}^{(2)} = v_3 - Proj_{u_1}(v_3) - \frac{(v_3,u_2)}{(u_2,u_2)}u_2 = v_3 - Proj_{u_1}(v_3) - Proj_{u_2}(v_3) \ = u_3 \ (of \ CGM) \quad (12)$$

Again by <u>mathematical induction</u>, we presume that the equality holds for $u_3{}^{(2)}, u_4{}^{(3)} \ldots u_{k-1}{}^{(k-2)}$ and we consider $u_k{}^{(k-1)}$:

$$u_k{}^{(1)} = v_k - Proj_{u_1}(v_k);\;\; u_k{}^{(2)} = u_k{}^{(1)} - Proj_{u_2}\left(u_k{}^{(1)}\right);$$

$$..\; u_k{}^{(k-2)} = u_k{}^{(k-3)} - Proj_{u_{k-2}}\left(u_k{}^{(k-3)}\right);\;\; u_k{}^{(k-1)} = u_k{}^{(k-2)} - Proj_{u_{k-1}}\left(u_k{}^{(k-2)}\right);\; e_3 = \frac{u_k{}^{(k-1)}}{||u_k{}^{(k-1)}||} \quad (13)$$

$$u_{k-1}{}^{(k-2)} = v_{k-1} - \sum_{j=1}^{k-2} Proj_{u_j}(v_{k-1})$$

$$u_k{}^{(k-1)} = u_k{}^{(k-3)} - Proj_{u_{k-2}}\left(u_k{}^{(k-3)}\right) - Proj_{u_{k-1}}\left(u_k{}^{(k-3)} - Proj_{u_{k-2}}\left(u_k{}^{(k-3)}\right)\right)$$

$$u_k{}^{(k-1)} = u_k{}^{(k-3)} - Proj_{u_{k-2}}\left(u_k{}^{(k-3)}\right) - Proj_{u_{k-1}}\left(u_k{}^{(k-3)}\right) - Proj_{u_{k-1}}{}^{(Proj_{u_{k-2}}(u_k{}^{(k-3)}))} \quad (14)$$

the above last term is zero, as u_{k-1} and u_{k-2} are orthogonal. Substituting the $u_k{}^{(k-3)}$ and doing the same procedure until reaching $u_k{}^{(1)}$:

$$u_k{}^{(k-1)} = v_k - Proj_{u_1}(v_k) - Proj_{u_2}(v_k) - \ldots - Proj_{u_{k-2}}(v_k) - Proj_{u_{k-1}}(v_k) = v_k - \sum_{j=1}^{k-1} Proj_{u_j}(v_k) \quad (15)$$

So the exact equality of MGS and CGS has been proved.

2. As it is stated in slide 18:

$$G = A^T.A \quad (16)$$

$$G = L.L^T \quad (17)$$

$$Q = A.(L^T)^{-1} \quad (18)$$

(There would be three concepts involved here which needs to be proved.

- $(A.B)^T = B^T.A^T$
- $(A^T)^{-1} = (A^{-1})^T$
- $(A^T)^T = A$

<u>first:</u>

$$(A.B)^T = \left(\sum_{k=1}^{n} a_{ik} b_{kj}\right)^T = \sum_{k=1}^{n} a_{jk} b_{ki} = B^T.A^T$$

second:

The inverse matrix of A^T is given by $(A^T)^{-1}$. We may prove by showing that $(A^{-1})^T.A^T = I \; and \; A^T.(A^{-1})^T = I$:

We have $(A^{-1})^T.A^T = (A.A^{-1})^T = (I)^T = I$; $\quad$ $using\ (A.B)^T = B^T.A^T$

Hence $(A^{-1})^T$ is the inverse of A^T. Since the inverse matrix is unique, therefore $(A^T)^{-1} = (A^{-1})^T$.

third:

Let $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$ then interchanging rows and columns gives $A^T = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nm} \end{pmatrix}$ transosing it back yields A again $(A^T)^T = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = A.$

Now the main proof:

$$Q^T.Q = (A.(L^T)^{-1})^T.(A.(L^T)^{-1}) \xrightarrow{using(A.B)^T=B^T.A^T} = ((L^T)^{-1})^T.A^T.A.(L^T)^{-1}$$

As $G = A^T.A = L.L^T$;

$$Q^T.Q = ((L^T)^{-1})^T.L.L^T.(L^T)^{-1} \xrightarrow{L^T.(L^T)^{-1}=I \ \ \& \ \ \left((L^T)^{-1}\right)^T=\left((L^{-1})^T\right)^T=L^{-1}} Q^T.Q = L^{-1}.L.I = I$$

3. Based on the guidance in the class, there are two approaches for solving this example:

first: Using the CGS for constructing the orthogonal basis followed by orthonormalizing

$(x, x^3, x^5) \rightarrow (y_1, y_2, y_3)$

$$y_1 = \frac{x}{||x||}$$

$$y_2 = x^3 - (x^3, y_1)y_1 \quad ; \quad y_2 = \frac{y_2}{||y_2||}$$

$$y_3 = x^5 - (x^5, y_1)y_1 - (x^5, y_2)y_2 \quad ; \quad y_3 = \frac{y_3}{||y_3||}$$

$$Proj(f(x)) = (sin(x), y_1)y_1 + (sin(x), y_2)y_2 + (sin(x), y_3)y_3$$

Second: directly relating sin(x) to (x, x^3, x^5)*:*

$sin(x) \approx proj(f(x)) = C_1x + C_2x^3 + C_3x^5$

Now performing the inner product by (x, x^3, x^5), then we have three equations and three unknown:

$$(sin(x), x) = C_1(x, x) + C_2(x^3, x) + C_3(x^5, x)$$

$$(sin(x), x^3) = C_1(x, x^3) + C_2(x^3, x^3) + C_3(x^5, x^3)$$

$$(sin(x), x^5) = C_1(x, x^5) + C_2(x^3, x^5) + C_3(x^5, x^5)$$

Now using the relation $(f, g) = \int_{-1}^{1} f(x)g(x)dx$

$$(sin(x), x) = \int_{-1}^{1} sin(x)xdx = sin(x) - xcos(x) = 0.6023$$

$$(sin(x), x^3) = \int_{-1}^{1} sin(x)x^3dx = 3x^2sin(x) - x^3cos(x) - 6sin(x) + 6xcos(x) = 0.3541$$

$$(sin(x), x^5) = \int_{-1}^{1} sin(x)x^5dx = 120sin(x) - 20x^3cos(x) - x^5cos(x) - 60x^2sin(x) + 5x^4sin(x) - 120xcos(x) = 0.2502$$

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

$$(x^3, x) = (x, x^3) = \int_{-1}^{1} x^4dx = 2/5$$

$$(x^5, x) = (x, x^5) = (x^3, x^3) = \int_{-1}^{1} x^6dx = 2/7$$

$$(x^5, x^3) = \int_{-1}^{1} x^8dx = 2/9$$

$$(x^5, x^5) = \int_{-1}^{1} x^{10}dx = 2/11$$

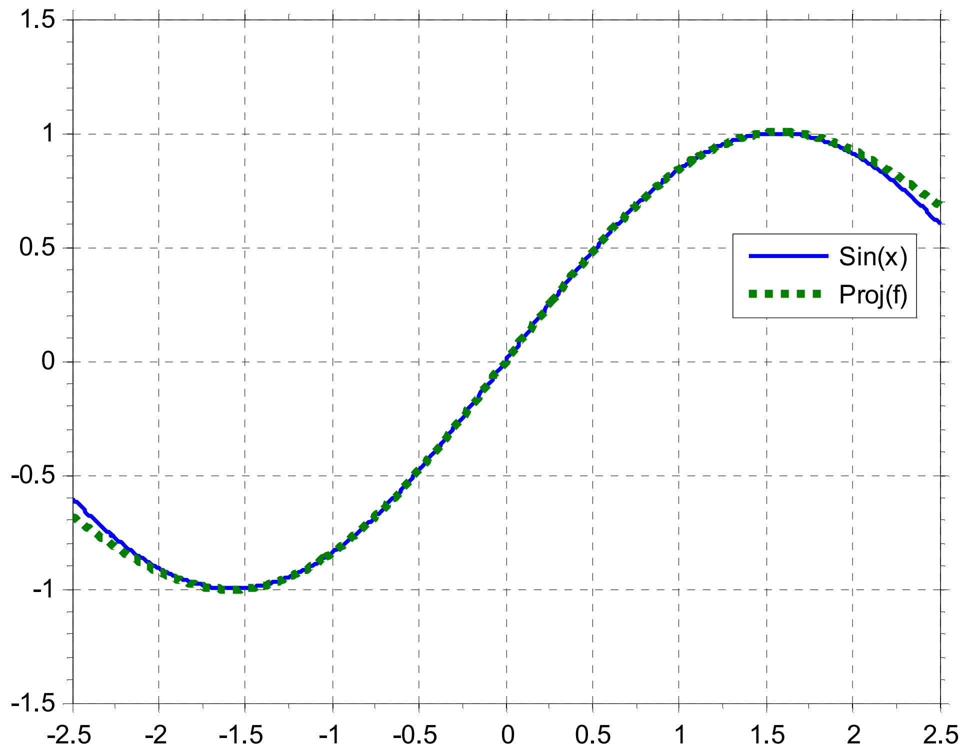
Therefore, if we write the short form of A.C=b where C=[C_1;C_2;C_3] and A and b are marix and vector of calculated items:

$$A = \begin{pmatrix} 2/3 & 2/5 & 2/7 \\ 2/5 & 2/7 & 2/9 \\ 2/7 & 2/9 & 2/11 \end{pmatrix} \qquad \& \qquad b \approx \begin{pmatrix} 0.6023 \\ 0.3541 \\ 0.2502 \end{pmatrix}$$

Then,

$$C \approx \begin{pmatrix} 0.9999 \\ -0.1665 \\ 0.0080 \end{pmatrix}$$

Result: Using the second approach, I have got:



PART II:

1. Explanation of lines:

> n=32; m=1000; setting the n and m to be 32 and 1000, respectively.

> j=0:n-1; j is defined to be from 0 to n-1; n values.

>sigma = 2.^(-j); sigma is a vector function of j and has n components and by using "." we defined it to be $2^{(-j)}$ component wise.

> X = randn(n); By definition, this function generates normally distributed pseudorandom numbers. randn(n) returns an n x n matrix containing pseudorandom values drawn from the standard normal distribution.

> [u,s,v]=svd(X); This function produces a diagonal matrix s, of the same dimension as X and with nonnegative diagonal elements in decreasing order, and unitary matrices u and v so that X = u*s*v'. Where unitary matrix is a matrix which satisfies $A.A^T=I$ or $A^T.A=I$.

> norm(X-u*s*v'); This is the second norm of a vector: norm(A)=(Sum(A(:)^2))^(1/2). For matrix, it is also the square root of all squared elements of that matrix. Now as we know that X = u*s*v', then norm of X-u*s*v' should be very small number in the order of matlab's precision (~10^-15).

> X=u*diag(sigma)*v'; Firstly, diag(sigma) , when sigma is a vector with n components, is a squared diagonal n by n matrix in which the components of sigma are the components of the main diagonal and all other components are zeros. u and v' are the matrices which we have obtained before by using svd(). Then, "*" is used to perform matrix by matrix multiplication operations. So now, X is the product of three n by n matrices; namely u, diag(sigma) and v' .

> cond(X); This function returns the second norm condition number of matrix X. (i.e. the ratio of the largest singular value of X to the smallest). Large condition numbers indicate a nearly singular matrix. The general formulation for cond(X,p) which is condition number based on p-norm is:

$$cond(X,p) = NORM(X,p) * NORM(INV(X),p)$$

In the case here, as we have diag(sigma) as singular value of X and then we have $cond(X,p) = max(sigma)/min(sigma)$. I've got

> [q,r]=chol_qr_it(X);

```
function [Q,R] = chol_qr_it(A)
  i=0;
  cn = 200;
  Q = A;                          (1)
  G = Q'*Q;                       (2)
  n = size(A,2);
  R = eye(n);                     (3)

  while cn > 100,
        i = i + 1
```

```
            [u,s,v]=svd(G);                  (4)
            [q,r]=qr(sqrt(s)*v');            (5)
            R = r * R;                       (6)
            cn = sqrt(cond(s));              (7)
            Q = Q * inv(r);                  (8)
            if cn>100
               G = Q'*Q;                     (9)
            end;
      end;
return
```

This line is a call to function chol_qr_it() which we have defined before. The lines of the function are explained briefly here:

(1) is to set the matrix which has been passed in (A) to the function as Q

(2) is to define G = $Q^T.Q$

(3) is to define R = I_n where n is the number of columns of A.

Now a while statement starts which continues until cn>100. Inside the loop:

(4) setting u,s and v as singular value decomposition of G.

(5) based on the fact that sqrt($A_{n \times n}$) is the n by n matrix of square root of A elements, we construct (sqrt(s)*v'). Then a n orthogonal-triangular decomposition is performed (QR) on the constructed matrix.

(6) Now R is updated as R = r*R in which r is the triangular part of previous decomposition.

(7) cn now is updated as taking the 2-norm condition number of marix s, followed by a square root of the result.

(8) Q is updated as Q=Q*inv(r).

(9) Now if cn>100, G=$Q^T.Q$

End of while loop and end of function.

<u>running this command resulted in 2 iterations.</u>

> norm(X-q*r);   As we have done a kind of QR factorization, so we expect that the norm of X-q*r would be again a small number in order of 10^{-15}. In this case we have got 1.6923e-15.

> norm(eye(n) – q'*q)   We also expect that $Q^T.Q=I$, so again the norm of eye(n) – q'*q should be in order of 10^{-15}. The result was 5.9449e-14.

> tic, [q,r]=chol_qr_it(X); toc

> tic, [Q,R]=qr(X,0); toc

These two lines are to compute the elapsed time between performing chol_qr_it() function and qr() function. The result was 0.003567 for former and 0.000565 for latter.

2. The code is attached to the folder I have sent.

Dimension of the matrix	Norm(X-Q*R)	Norm(G-I)
1000*32	9.623092e-16	1.904367e-14
2000*32	1.823188e-15	2.107398e-14
3000*32	1.513431e-15	2.725891e-14