Assignment 5

Zou Yuan
Student No. 21960216

October 21, 2019

1 P199, Computer Problems:1.

Form the normal equations, and compute the least squares solution and 2-norm error for the inconsistent systems.

The Matlab code is as follows:

```
A=[3,-1 2;4 1 0;-3 2 1;1 1 5;-2 0 3];
b=[10;10;-5;15;0];
x1=GaussElimination(A'*A,(A'*b)')
r1=norm(b-A*x1,2)
B=[4 2 3 0;-2 3 -1 1;1 3 -4 2;1 0 1 -1;3 1 3 -2];
c=[10;0;2;0;5];
x2=GaussElimination(B'*B,(B'*c)')
r2=norm(c-B*x2,2)

function [r_matrix] = GaussElimination(a,b)
%  inputs:
%          a:系数矩阵，为n*n维方阵
%          b:载荷矩阵，为n*1维矩阵
%  outputs:
%          r_matrix:计算结果向量,为n*1为矩阵

% 判断输入矩阵维度是否满足要求
```

```
[row_coeff,col_coeff] = size(a);
[row_load,~] = size(b');
% 初始化r_matrix矩阵
r_matrix = zeros(row_load,1);
% 判断输入的维度是否满足要求
if (row_coeff ~= col_coeff) || (row_coeff ~= row_load)
% 不满足则输出错误提示
print('输入错误！ ');
%消去过程
else
n=row_coeff;
for j = 1 : n-1
if abs(a(j,j))<eps; error('zero pivot encountered');
end
for i = j+1 : n
mult = a(i,j)/a(j,j);
for k = j+1:n
a(i,k) = a(i,k) - mult*a(j,k);
end
b(i) = b(i) - mult*b(j);
end
end
%回代过程
r_matrix(n) = b(n)/a(n,n);
for k = n-1:-1:1
sum_temp = 0;
for j = k+1:n
sum_temp = sum_temp + a(k,j)*r_matrix(j);
end
r_matrix(k) = (b(k) - sum_temp)/a(k,k);
end
end % 条件判断结束
end
```

The input is the system matrix,where the least squares solution is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.5246 \\ 0.6616 \\ 2.0934 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1.2739 \\ 0.6885 \\ 1.2124 \\ 1.7497 \end{bmatrix}$$

the 2-norm error is:

$$r1 = 2.413 \qquad r2 = 0.8256$$

2 P225, Computer Problems:1.

Write a Matlab program that implements classical Gram–Schmidt to find the reduced QR factorization. Check your work by comparing factorizations of the matrices in Exercise 1 with the Matlab qr(A,0) command or equivalent. The factorization is unique up to signs of the entries of Q and R.

```
A=[4,0;3,1];
B=[1,2;1,1];
C=[2,1;1,-1;2,1];
D=[4,8,1;0,2,-2;3,6,7];
[q1,r1]=Schmidt(A)
[q11,r12]=qr(A,0)

[q2,r2]=Schmidt(B)
[q22,r22]=qr(B,0)

[q3,r3]=Schmidt(C)
[q33,r33]=qr(C,0)

[q4,r4]=Schmidt(D)
[q44,r44]=qr(D,0)

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

```
[row, col] = size(A);
R=zeros(row, col);
Q=zeros(row, row);
for j=1:col
y=A(:, j);
for i=1:j-1
R(i, j)=(Q(:, i))'*A(:, j);
y=y-R(i, j)*Q(:, i);
end
R(j, j)=norm(y, 2);
Q(:, j)=y/R(j, j);
end
end
```

To compare the outputs between Schmidt and the Matlab qr(A,0) command:

$$q1 = \begin{bmatrix} 0.8000 & -0.6000 \\ 0.6000 & 0.8000 \end{bmatrix} \qquad r1 = \begin{bmatrix} 5.0000 & 0.6000 \\ 0 & 0.8000 \end{bmatrix}$$

$$q11 = \begin{bmatrix} -0.8000 & -0.6000 \\ -0.6000 & 0.8000 \end{bmatrix} \qquad r11 = \begin{bmatrix} -5.0000 & -0.6000 \\ 0 & 0.8000 \end{bmatrix}$$

$$q2 = \begin{bmatrix} 0.7071 & 0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} \qquad r2 = \begin{bmatrix} 1.4142 & 2.1213 \\ 0 & 0.7071 \end{bmatrix}$$

$$q22 = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \qquad r22 = \begin{bmatrix} -1.4142 & -2.1213 \\ 0 & -0.7071 \end{bmatrix}$$

$$q3 = \begin{bmatrix} 0.6667 & 0.2357 & 0 \\ 0.3333 & -0.9428 & 0 \\ 0.6667 & 0.2357 & 0 \end{bmatrix} \qquad r3 = \begin{bmatrix} 3.0000 & 1.0000 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix}$$

$$q33 = \begin{bmatrix} -0.6667 & 0.2357 \\ -0.3333 & -0.9428 \\ -0.6667 & 0.2357 \end{bmatrix} \qquad r33 = \begin{bmatrix} -3.0000 & -1.0000 \\ 0 & 1.4142 \end{bmatrix}$$

$$q4 = \begin{bmatrix} 0.8000 & 0 & -0.6000 \\ 0 & 1.0000 & 0 \\ 0.6000 & 0 & 0.8000 \end{bmatrix} \qquad r4 = \begin{bmatrix} 5 & 10 & 5 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix}$$

$$q44 = \begin{bmatrix} -0.8000 & 0 & -0.6000 \\ 0 & 1.0000 & 0 \\ -0.6000 & 0 & 0.8000 \end{bmatrix} \qquad r44 = \begin{bmatrix} -5 & -10 & -5 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix}$$

3 P225, Computer Problems:3.

Repeat Computer Problem 1, but implement Householder reflections.

Code is below:

```
A=[4 0;3 1];
B=[1 2;1 1];
C=[2 1;1 -1;2 1];
D=[4 8 1;0 2 -2;3 6 7];
[Q1,R1]=qr(A)
[q1,r1]=qrhs(A)
[Q2,R2]=qr(B)
[q2,r2]=qrhs(B)
[Q3,R3]=qr(C)
[q3,r3]=qrhs(C)
[Q4,R4]=qr(D)
[q4,r4]=qrhs(D)
function [H,rho]=householder(x,y)

x=x(:);
y=y(:);
if length(x)~=length(y)
error('Error!');
end
rho=-sign(x(1))*norm(x)/norm(y);
```

```
y=rho*y;
v=(x-y)/norm(x-y);
I=eye(length(x));
H=I-2*v*v';
end

function [Q,R]=qrhs(A)
n=size(A,1);
R=A;
Q=eye(n);
for i=1:n-1
x=R(i:n,i);
y=[1;zeros(n-i,1)];
Ht=householder(x,y);
H=blkdiag(eye(i-1),Ht);
Q=Q*H;
R=H*R;
end
end
```

The output is:

$$Q1 = \begin{bmatrix} -0.8000 & -0.6000 \\ -0.6000 & 0.8000 \end{bmatrix} \qquad R1 = \begin{bmatrix} -5.0000 & -0.6000 \\ 0 & 0.8000 \end{bmatrix}$$

$$q1 = \begin{bmatrix} -0.8000 & -0.6000 \\ -0.6000 & 0.8000 \end{bmatrix} \qquad r1 = \begin{bmatrix} -0.8000 & -0.6000 \\ -0.6000 & 0.8000 \end{bmatrix}$$

$$Q2 = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \qquad R2 = \begin{bmatrix} -1.4142 & -2.1213 \\ 0 & -0.7071 \end{bmatrix}$$

$$q2 = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \qquad r2 = \begin{bmatrix} -1.4142 & -2.1213 \\ 0 & -0.7071 \end{bmatrix}$$

$$Q3 = \begin{bmatrix} -0.6667 & 0.2357 & -0.7071 \\ -0.3333 & -0.9428 & -0.0000 \\ -0.6667 & 0.2357 & 0.7071 \end{bmatrix} \qquad R3 = \begin{bmatrix} -3.0000 & -1.0000 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix}$$

$$q3 = \begin{bmatrix} -0.6667 & 0.2357 & -0.7071 \\ -0.3333 & -0.9428 & -0.0000 \\ -0.6667 & 0.2357 & 0.7071 \end{bmatrix} \qquad r3 = \begin{bmatrix} -3.0000 & -1.0000 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix}$$

$$Q4 = \begin{bmatrix} -0.8000 & 0 & -0.6000 \\ 0 & 1.0000 & 0 \\ -0.6000 & 0 & 0.8000 \end{bmatrix} \qquad r4 = \begin{bmatrix} -5 & -10 & -5 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix}$$

$$q4 = \begin{bmatrix} -0.8000 & 0 & -0.6000 \\ 0 & 1.0000 & 0 \\ -0.6000 & 0 & 0.8000 \end{bmatrix} \qquad r4 = \begin{bmatrix} -5 & -10 & -5 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix}$$