

Problem 8.2

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

[m,n] = size(A);
if (m < n)
    print("Error: rows more than cols!")
end

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

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

for i = 1 : n
    R(i,i) = norm(Q(:,i),2);
    Q(:,i) = Q(:,i) / R(i,i);
    for j = (i + 1) : n
        R(i,j) = Q(:,i)' * Q(:,j);
        Q(:,j) = Q(:,j) - R(i,j) * Q(:,i);
    end
end
end
```

Problem 10.2

(a)

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

[m,n] = size(A);
if (m < n)
    print("Error: rows more than cols!")
end

W = zeros(m,n);

for k = 1:n
    x = A(k:m,k);
    if (x(1) > 0)
        sgn = 1;
    else
        sgn = -1;
    end
    v = sgn * norm(x,2) * eye(m - k + 1, 1) + x;
    v = v / norm(v,2);
```

```

        A(k:m,k:n) = A(k:m,k:n) - 2 * v * v' * A(k:m,k:n);
        W(k:m,k) = v;
    end

    R = A(1:n,:);

end

(b)
function Q = formQ(W)

[m,n] = size(W);
if (m < n)
    print("Error: rows more than cols!")
end

Q = eye(m,m);

for k = 1:m
    Q(:,k) = formQx(W,Q(:,k));
end

end

function y = formQx(W,x)

[m,n] = size(W);
if (m < n)
    print("Error: rows more than cols!")
end

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

y = x;
end

```

Problem 11.3

```
function x = ploy_apprx
```

```
format long;
```

```
m = 50; n = 12;
```

```

t = linspace(0, 1, m);
A = fliplr(vander(t));
A = A(:,1:n);
b = cos(4 * t)';

%(a)
R = chol(A' * A);
x1 = R \ (R' \ (A' * b));

%(b)
[Q, R] = mgs(A);
x2 = R \ (Q' * b);

%(c)
[W, R] = house(A);
Q = formQ(W);
Q = Q(:,1:n);
x3 = R \ (Q' * b);

%(d)
[Q, R] = qr(A);
x4 = R \ (Q' * b);

%(e)
x5 = A \ b;

%(f)
[U, S, V] = svd(A, 0);
x6 = V * (S \ (U' * b));

x = [x1, x2, x3, x4, x5, x6];

end

```

Columns 1 through 5

0.999999996787553	0.999999998386318	1.000000000996608	1.000000000996608	1.000000000996607
0.000000350916732	0.000000361260742	-0.000000422743090	-0.000000422743080	-0.000000422743364
-8.000003028795119	-8.000010909012820	-7.999981235684265	-7.999981235685203	-7.999981235676154
-0.000077893877909	0.000120971851973	-0.000318763242871	-0.000318763231287	-0.000318763346323
10.668084035612900	10.666048872592777	10.669430795921651	10.669430795858052	10.669430796641096
-0.009615585352545	0.001499873032843	-0.013820287883617	-0.013820287698367	-0.013820290914619
-5.654480747960067	-5.690652240820818	-5.647075628133985	-5.647075628404193	-5.647075619959385
-0.068885958631546	0.004709001274402	-0.075316022140262	-0.075316022079922	-0.075316036589419
1.693354534665567	1.598825773422053	1.693606960130184	1.693606960559036	1.693606976803618
0.001461547101732	0.075948858071100	0.006032111747578	0.006032111063859	0.006032099645104
-0.370576739064076	-0.403455413478070	-0.374241704896969	-0.374241704456940	-0.374241699881279
0.087095866530693	0.093321235777330	0.088040576367275	0.088040576259675	0.088040575462356

Column 6

```
1.000000000996608
-0.000000422743088|
-7.999981235684747
-0.000318763237547
10.669430795900578
-0.013820287867134
-5.647075627982760
-0.075316022763597
1.693606961280185
0.006032110585745
-0.374241704275633
0.088040576229625
```

We can see that (c)-(f) are consistent, (b) computes some answer not correct, and (a) is not stable.