```
clear;clc;close;

%Variante 1
% (a)

err1 = [];

for n = 2:1000

e1 = zeros(n,1); e1(1) = 1;
x = e1 + 1.0e-7*sin(1:n)'/sqrt(n);

v = x / norm(x, inf);

if (v(1) >= 0 )
  rho = norm(v);
else
  rho = -norm(v);
end

u = v + rho*e1;
theta = rho*u(1);
sigma = rho*norm(x,inf);
P = eye(n,n) - (u*u')/theta;

##if(norm(P*x + sigma*e1,2)> (10^-14)*norm(x,2))
##    fprintf("(a) falsch bei Dimension : ")
##    n
##    break
##end

err1 = [err1,norm(P*x+sigma*e1,2)/norm(x,2)];

end

subplot(2,2,1)
plot(log10(err1))
title('(a) - richtig ')
err2 = [];

% (b) y statt x, sonst gleich wie (a)

for n = 2:1000

e1 = zeros(n,1); e1(1) = 1;
y = realmax*sin(1:n)'/sqrt(n);

v = y / norm(y, inf);

if (v(1) >= 0 )
  rho = norm(v);
else
  rho = -norm(v);
end

u = v + rho*e1;
theta = rho*u(1);
sigma = rho*norm(y,inf);
P = eye(n,n) - (u*u')/theta;

##if(norm(P*y + sigma*e1,2) > (10^-14)*norm(y,2))
##    fprintf("(b) falsch bei Dimension : ")
##    n
##    break
##end

err2 = [err2,norm(P*y+sigma*e1,2)/norm(y,2)];

end

subplot(2,2,2)
plot(log10(err2))
```

```
title('(b) - richtig')

err3 = [];

%Variante 2
% (c) nicht skaliert, also v= x, sigma=rho, sonst gleich wie (a)

for n = 2:1000

e1 = zeros(n,1); e1(1) = 1;
y = realmax*sin(1:n)'/sqrt(n);

v = y ;

if (v(1) >= 0 )
  rho = norm(v);
else
  rho = -norm(v);
end

u = v + rho*e1;
theta = rho*u(1);
sigma = rho;
P = eye(n,n) - (u*u')/theta;

err3 = [err3,norm(P*y+sigma*e1,2)/norm(y,2)];


##if(norm(P*y + sigma*e1,2)> (10^-14)*norm(y,2))
##   fprintf("(c) falsch bei Dimension : ")
##    n
##    break
##end

end

subplot(2,2,3)
plot(log10(err3))
title('(c) - Falsch ')

err4 = [];

% %Variante 3
% (d) Vorzeichen von rho verändert, sonst gleich wie (a)

for n = 2:1000

e1 = zeros(n,1); e1(1) = 1;
x = e1 + 1.0e-7*sin(1:n)'/sqrt(n);

v = x/norm(x,inf);

if (v(1) >= 0 )
  rho = -norm(v);
else
  rho = norm(v);
end

u = v + rho*e1;
theta = rho*u(1);
sigma = rho*norm(x,inf);
P = eye(n,n) - (u*u')/theta;

##if(norm(P*x + sigma*e1,2)> (10^-14)*norm(x,2))
##    fprintf("(d) falsch bei Dimension : ")
##    n
##    break
##end

err4 = [err4,norm(P*x+sigma*e1,2)/norm(x,2)];

end
```

```
subplot(2,2,4)
plot(log10(err4))
title('(d) - Falsch')
```

```
% Blatt 6 - Aufgabe 3

clear, clc

N = 5; h = 1/N; mu = h^2;

C = spdiags([-ones(N,1),ones(N,1)],[0,1],N,N);
I = speye(N);
A = I + mu * C' * C/h^2;

kmax = 20;

for k =  1:kmax
    for i = 1:N-1
    a(i) = A(i,i);
    b(i) = A(i,i+1);
    x(1) = a(1);
    y(1) = b(1);
    a(N) = A(N,N);
    end

    for j = 1:N-1
        g(j) = sqrt(x(j)^2 + b(j)^2);
        c(j) = x(j)/g(j);
        s(j) = b(j)/g(j);
        d(j) = c(j) * y(j) + s(j) * a(j+1);
        x(j+1) = c(j) * a(j+1) - s(j) * y(j);
        if j < (N-1)
            e(j) = s(j) * b(j+1);
            y(j+1) = c(j) * b(j+1);
        end
    end

    g(N) = x(N);
    d1=[0;d.'];
    e1=[0;0;e.'];
    R = spdiags([g.',d1,e1],0:2,N,N);
    % besser: R = diag(g) + diag(d,1) + diag(e,2);
    R = full(R);

    a(1) = s(1) * d(1) + c(1) * g(1);
    b(1) = s(1) * g(2);
    for j = 2:N-1 % R * Q
        a(j) = s(j) * d(j) + c(j) * c(j-1) * g(j);
        b(j) = s(j) * g(j+1);
    end

    a(N) = c(N-1) * g(N);

    b1 = [0;b.'];
    b2 = [b.';0];
    A = spdiags([b2,a.',b1],-1:1,N,N);
    A = full(A); % = A^(k+1)

    % a) falsch, weil Q obere Hessenberg sein muss

    if k == 9   % b) richtig
        A
    end
    if k == 15  % c) richtig
        R
    end
    if k == 19  % d) falsch
        A
    end

end
```

```
A =

    4.6049   -0.2467         0         0         0
   -0.2467    3.8839   -0.1671         0         0
         0   -0.1671    2.7374   -0.0510         0
         0         0   -0.0510    1.6914   -0.0292
         0         0         0   -0.0292    1.0824


R =

    4.6723   -0.1769    0.0006         0         0
         0    3.8385   -0.0500    0.0000         0
         0         0    2.7158   -0.0077    0.0000
         0         0         0    1.6903   -0.0051
         0         0         0         0    1.0810


A =

    4.6810   -0.0360         0         0         0
   -0.0360    3.8323   -0.0052         0         0
         0   -0.0052    2.7154   -0.0004         0
         0         0   -0.0004    1.6903   -0.0003
         0         0         0   -0.0003    1.0810
```

Published with MATLAB® R2020b

```
clear all;
clc;
##Code Householder-Transformation Seite 106
function [theta,sigma,u]=Householder(x)
  k=length(x);
  e(1)=1;
  e(2:k)=0;
  v=x*1.0/norm(x,inf);
  if v(1)>=0
    s=1;
  else
    s=-1;
  endif
  phi=s*norm(v,2);
  u=v+phi*e;
  theta=phi*u(1);
  sigma=phi*norm(x,inf);
  u=u.';
endfunction

##Definition x
m=10;
for i=1:m
  x(i)=(i-0.5)/m;
endfor

##Definition von V=A
n=5;
for i=1:n
  A(:,i)=x.^(i-1);
endfor

##A=[d,C] , d=1.spalte von A
for i=1:m
  d(i)=A(i,1);
  for j=1:n-1
    C(i,j)=A(i,j+1);
  endfor
endfor

##Code Seite 130-131
P=speye(m);
Q=speye(n);
a=0;
for k=1:n

  ##Householder-Transformation mit d
  [theta,sigma,u]=Householder(d);

  B(k,k)=-sigma;

  if k>1
    B(k-1,k)=a;
  endif
  ##Bestimme C und P
  for l=1:n-k
    sm=C(1:m-k+1,l).'*u;
    C(1:m-k+1,l)=C(1:m-k+1,l)-sm*u/theta;
  endfor
  for l=1:m
    sm=P(k:m,l).'*u;
    P(k:m,l)=P(k:m,l)-sm*u*1.0/theta;
  endfor

  ##Householder-Transformation mit c
  if k<n-1
    c=C(1,1:n-k);
    [phi,tau,v]=Householder(c);
    a=-tau;
    ##Bestimme C und Q
    for l=2:m-k+1
      sm=C(l,1:n-k)*v;
      C(l,1:n-k)=C(l,1:n-k)-sm*v.'*1.0/phi;
    endfor
    for l=1:n
      sm=Q(l,k+1:n)*v;
      Q(l,k+1:n)=Q(l,k+1:n)-sm*v.'*1.0/phi;
    endfor
  endif
  if k==n-1
    a=C(1,1);
  endif
  ##Verkleinere d und  C
  if k<n
    d=C(2:m-k+1,1).';
    C=C(2:m-k+1,2:n-k);
  endif

endfor
##Kontrolle
B=full(B)
Q=full(Q)
Z= speye(n)-speye(n);   ##Nullmatrix
if norm(P*A*Q - [B;Z])<10^(-15)
  disp("Rechnung stimmt")
endif
```

```
m =10;
n=5;
x = ([1:m]-0.5)/m;
V = x'.^[0:n-1];
[U,S,W] = svd(V);                %singulärwertzerlegung
cstar = ones(n,1);
ystar = V*cstar;

[E,L] = eig(V'*V);

%%%%%%%%%%%%%%%%%%%% (a) %%%%%%%%%%%%%%%%%%

e1 = E(1:n,1)*0.01;
cdd = (V'*V)\(V'*ystar + e1);
cd = ([diag(ones(n,1)),zeros(n,m-n)]*S*W')\...
    ([diag(ones(n,1)),zeros(n,m-n)]*U'*ystar + e1);

la = norm(cstar-cd)<norm(cstar-cdd)

%%%%%%%%%%%%%%%%%%%% (b) %%%%%%%%%%%%%%%%%%

en = E(1:n,n)*0.01;
cdd = (V'*V)\(V'*ystar + en);
cd = ([diag(ones(n,1)),zeros(n,m-n)]*S*W')\...
    ([diag(ones(n,1)),zeros(n,m-n)]*U'*ystar + en);

lb = norm(cstar-cd)<norm(cstar-cdd)

%%%%%%%%%%%%%%%%%%%% (c) %%%%%%%%%%%%%%%%%%

cdd = (V'*V)\(V'*ystar + e1);
cd = ([diag(ones(n,1)),zeros(n,m-n)]*S*W')\...
    ([diag(ones(n,1)),zeros(n,m-n)]*U'*ystar + e1);

a = norm(cstar - cd)*norm([diag(ones(n,1)),zeros(n,m-n)]*U'*ystar);
a = a/(norm(cstar)*norm(e1));

b = norm(cstar-cdd)*norm(V'*ystar)/(norm(cstar)*norm(e1));

lc = a^2<b

%%%%%%%%%%%%%%%%%%%% (d) %%%%%%%%%%%%%%%%%%

cdd = (V'*V)\(V'*ystar + en);
cd = ([diag(ones(n,1)),zeros(n,m-n)]*S*W')\...
    ([diag(ones(n,1)),zeros(n,m-n)]*U'*ystar + en);

x = linspace(0,1,1000);

cstar = flip(cstar);
cd = flip(cd);
cdd = flip(cdd);

ld = max(abs(polyval(cstar,x)-polyval(cd,x)))<...
    max(abs(polyval(cstar,x)-polyval(cdd,x)))


plot(x,abs(polyval(cstar,x)-polyval(cd,x)),'r')
hold on
plot(x,abs(polyval(cstar,x)-polyval(cdd,x)),'g')
legend('|P(x;c*)-P(x;cd)|','|P(x;c*)-P(x;cdd)|')
hold off
```