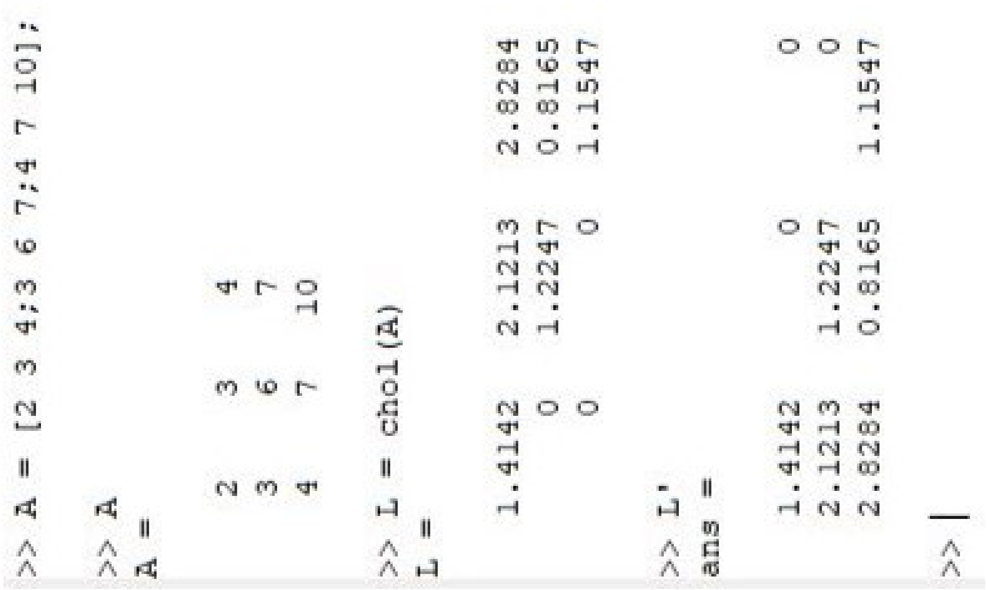
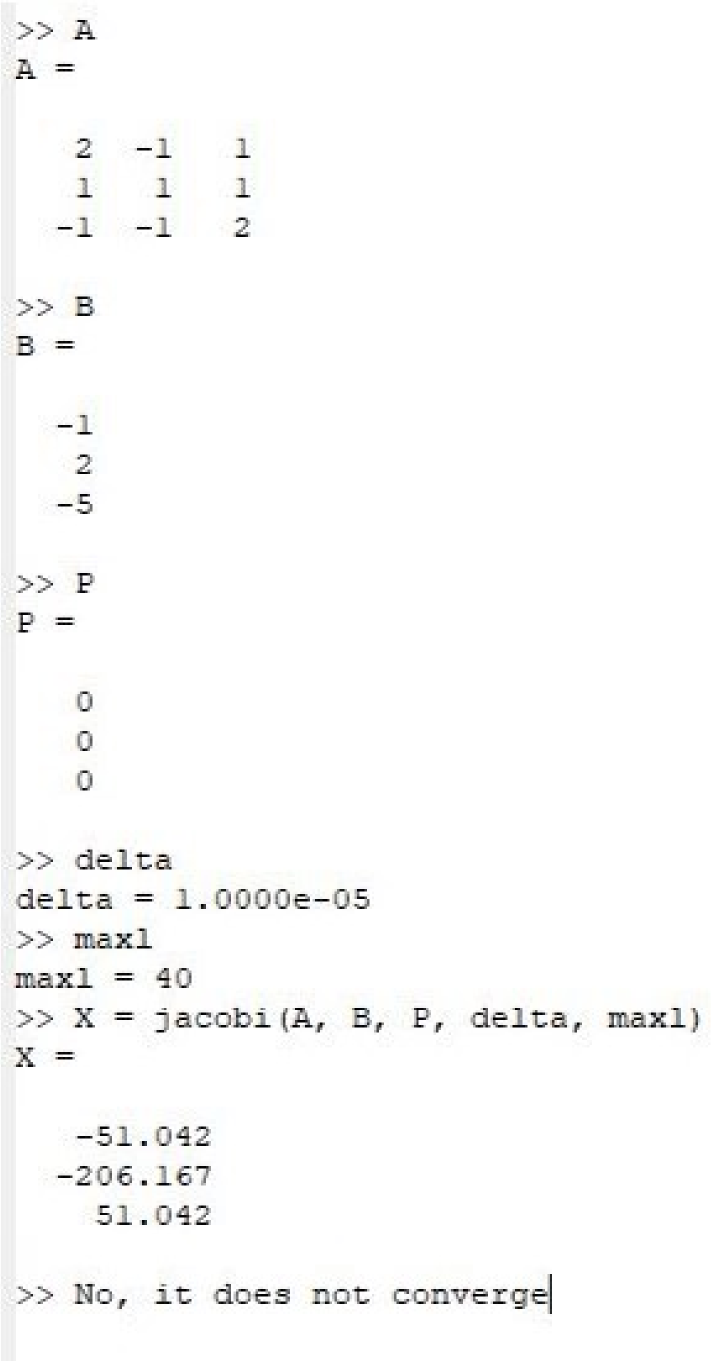
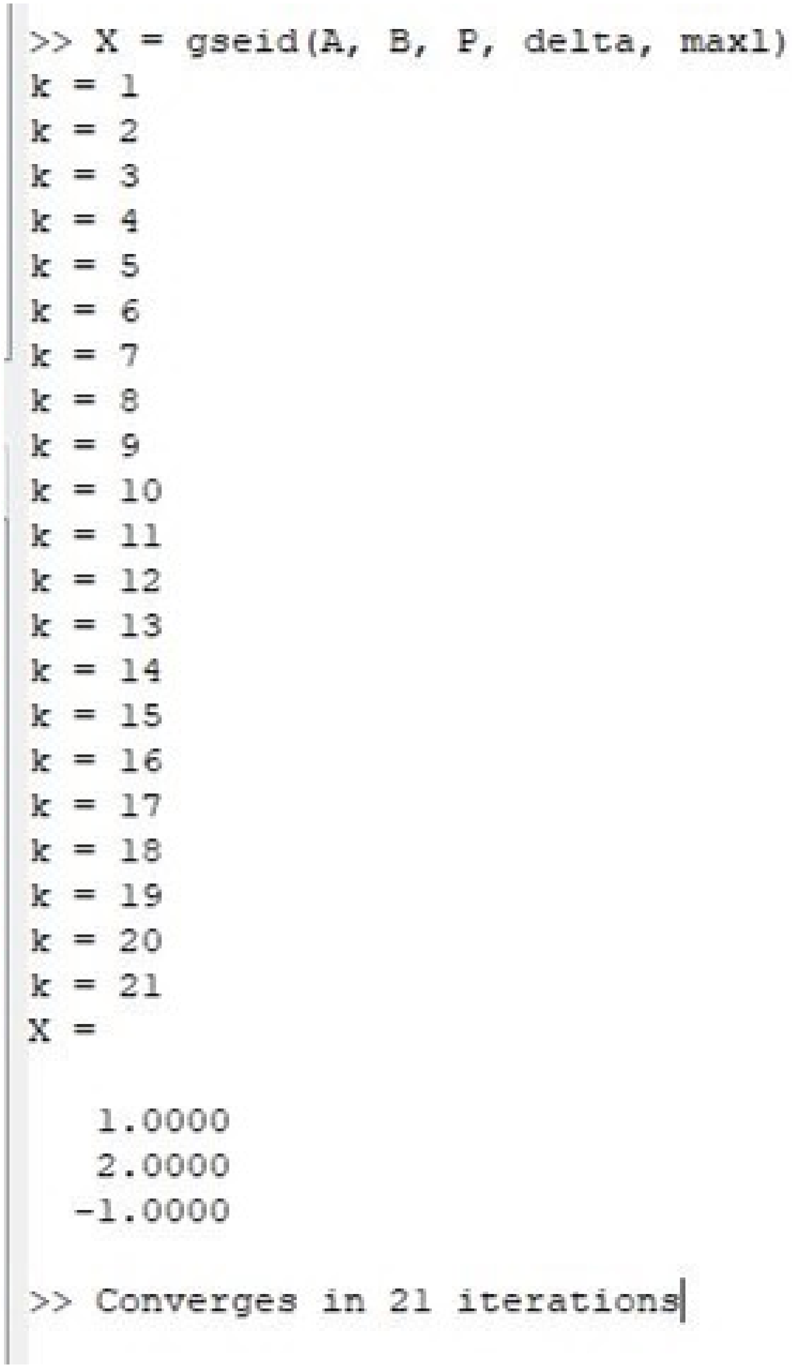


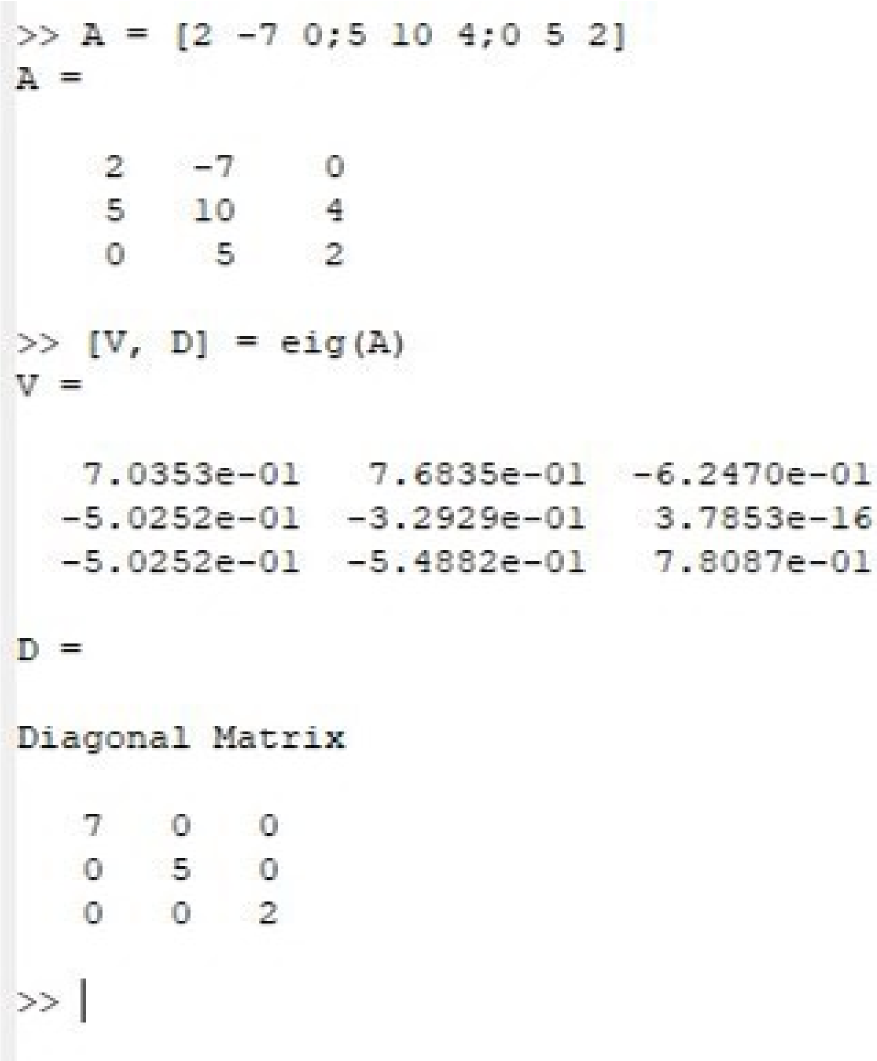
```
>> A = [2 3 4;3 6 7;4 7 10];
>> A
A =

     2     3     4
     3     6     7
     4     7    10

>> L = chol(A)
L =

   1.4142   2.1213   2.8284
        0   1.2247   0.8165
        0        0   1.1547

>> L'
ans =

   1.4142        0        0
   2.1213   1.2247        0
   2.8284   0.8165   1.1547

>> 
```



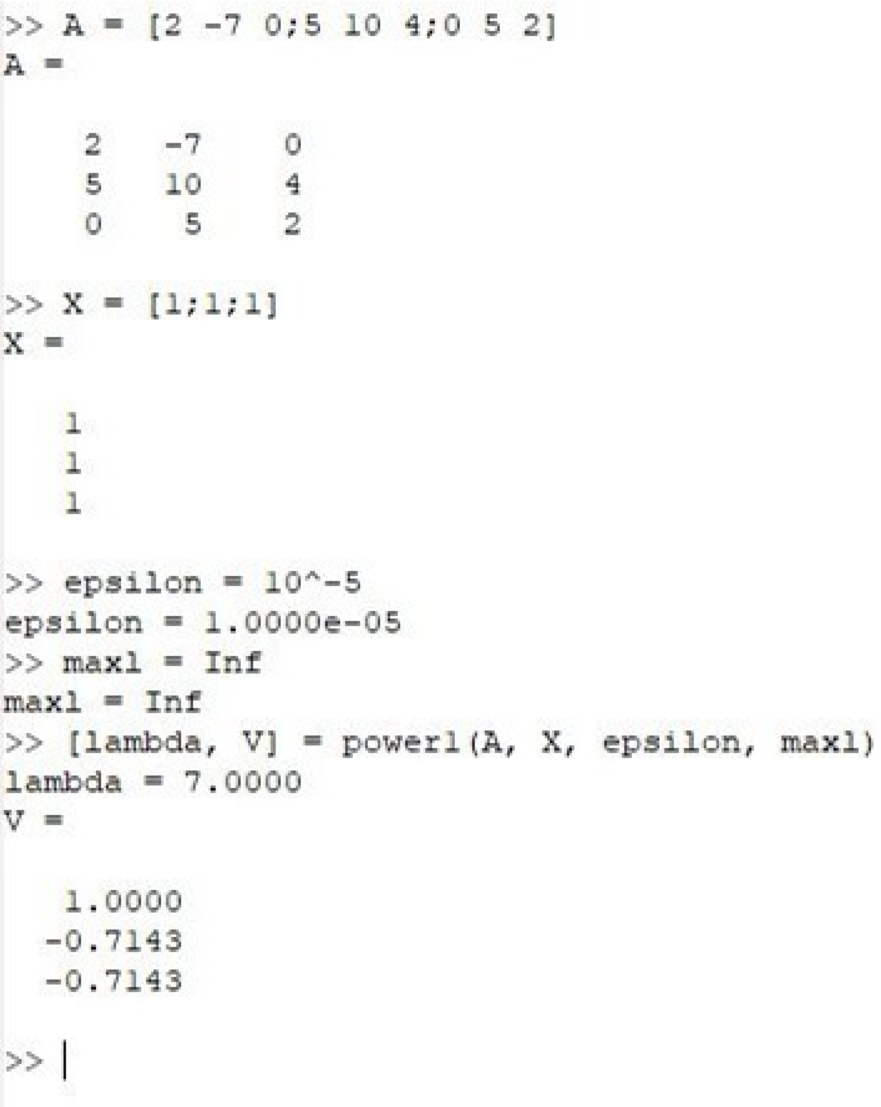
```
>> A
A =

    2   -1    1
    1    1    1
   -1   -1    2

>> B
B =

   -1
    2
   -5

>> P
P =

    0
    0
    0

>> delta
delta = 1.0000e-05
>> maxl
maxl = 40
>> X = jacobi(A, B, P, delta, maxl)
X =

    -51.042
   -206.167
     51.042

>> No, it does not converge
```



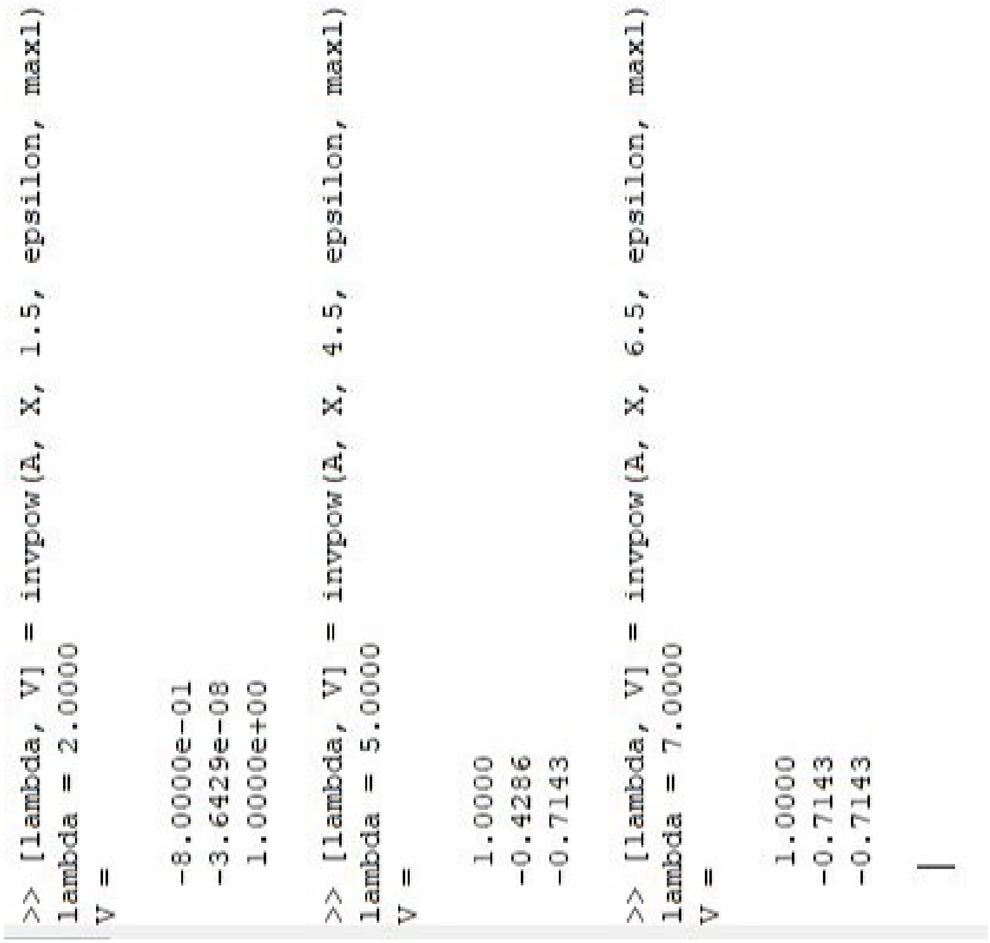
```
>> X = gseid(A, B, P, delta, max1)
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8
k = 9
k = 10
k = 11
k = 12
k = 13
k = 14
k = 15
k = 16
k = 17
k = 18
k = 19
k = 20
k = 21
X =

   1.0000
   2.0000
  -1.0000

>> Converges in 21 iterations
```



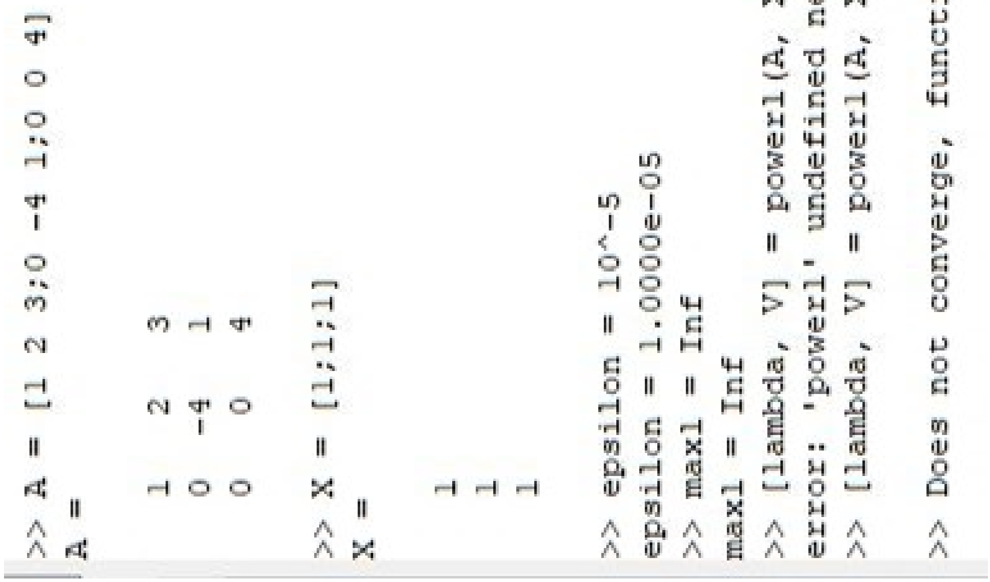
```
>> A = [2 -7 0;5 10 4;0 5 2]
A =


    2   -7    0
    5   10    4
    0    5    2

>> [V, D] = eig(A)
V =


   7.0353e-01   7.6835e-01  -6.2470e-01
  -5.0252e-01  -3.2929e-01   3.7853e-16
  -5.0252e-01  -5.4882e-01   7.8087e-01

D =

Diagonal Matrix

   7   0   0
   0   5   0
   0   0   2

>> 
```



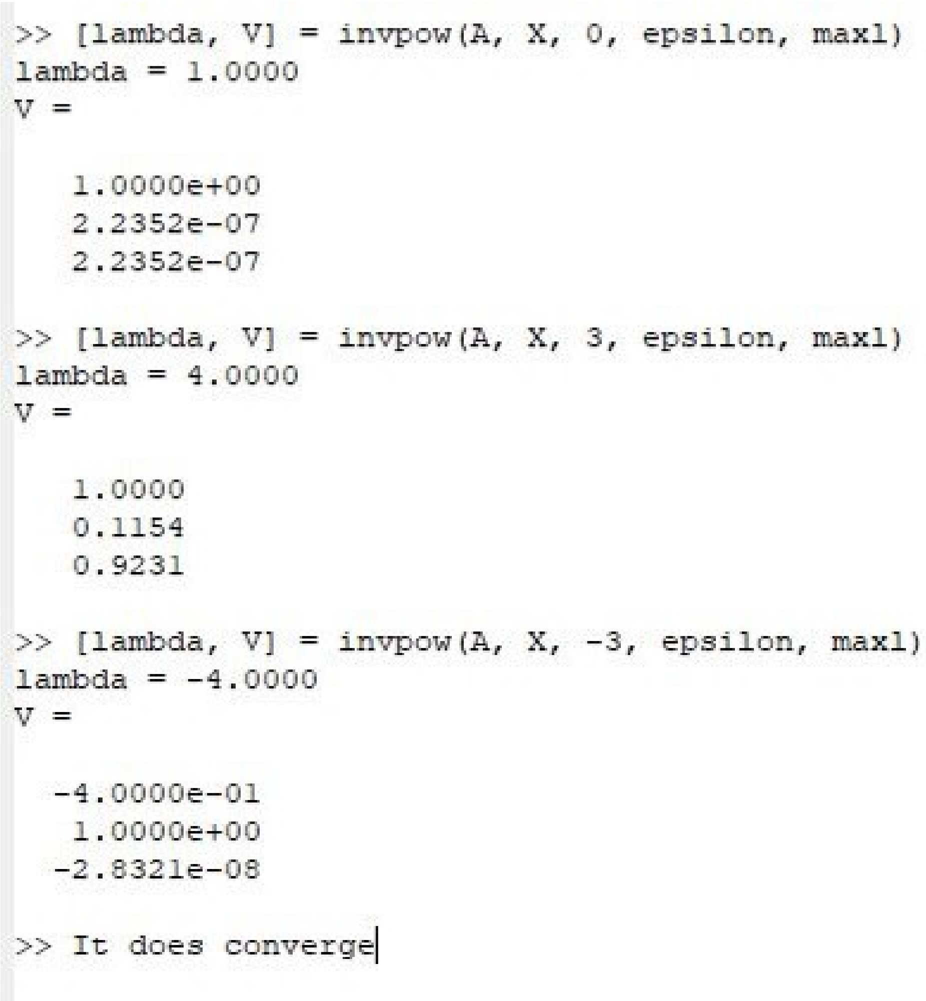
```
>> A = [2 -7 0;5 10 4;0 5 2]
A =

     2    -7     0
     5    10     4
     0     5     2

>> X = [1;1;1]
X =

   1
   1
   1

>> epsilon = 10^-5
epsilon = 1.0000e-05
>> maxl = Inf
maxl = Inf
>> [lambda, V] = power1(A, X, epsilon, maxl)
lambda = 7.0000
V =

   1.0000
  -0.7143
  -0.7143

>> 
```



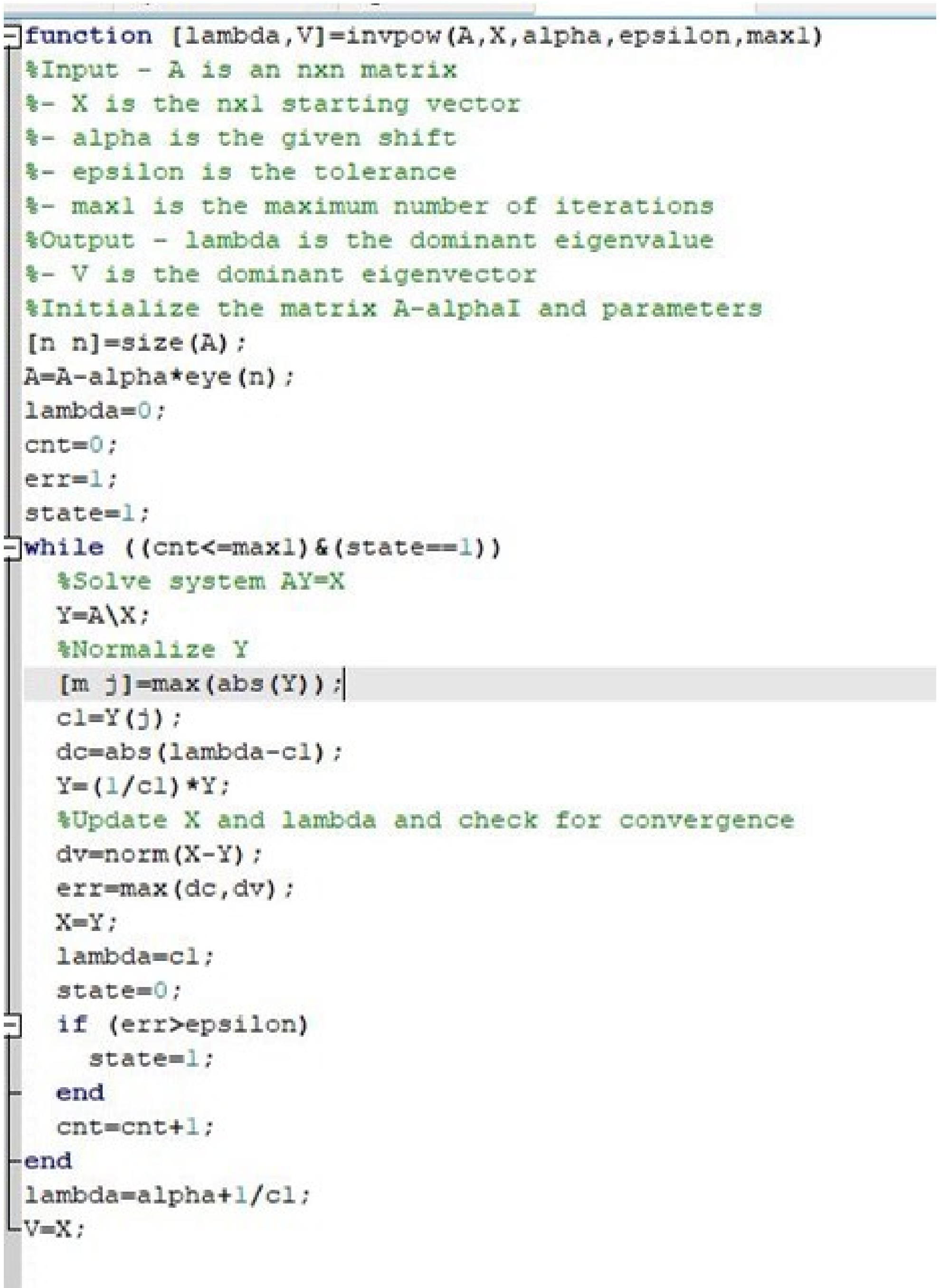
```
>> [lambda, V] = invpow(A, X, 1.5, epsilon, maxl)
lambda = 2.0000
V =

  -8.0000e-01
  -3.6429e-08
   1.0000e+00

>> [lambda, V] = invpow(A, X, 4.5, epsilon, maxl)
lambda = 5.0000
V =

   1.0000
  -0.4286
  -0.7143

>> [lambda, V] = invpow(A, X, 6.5, epsilon, maxl)
lambda = 7.0000
V =

   1.0000
  -0.7143
  -0.7143
```



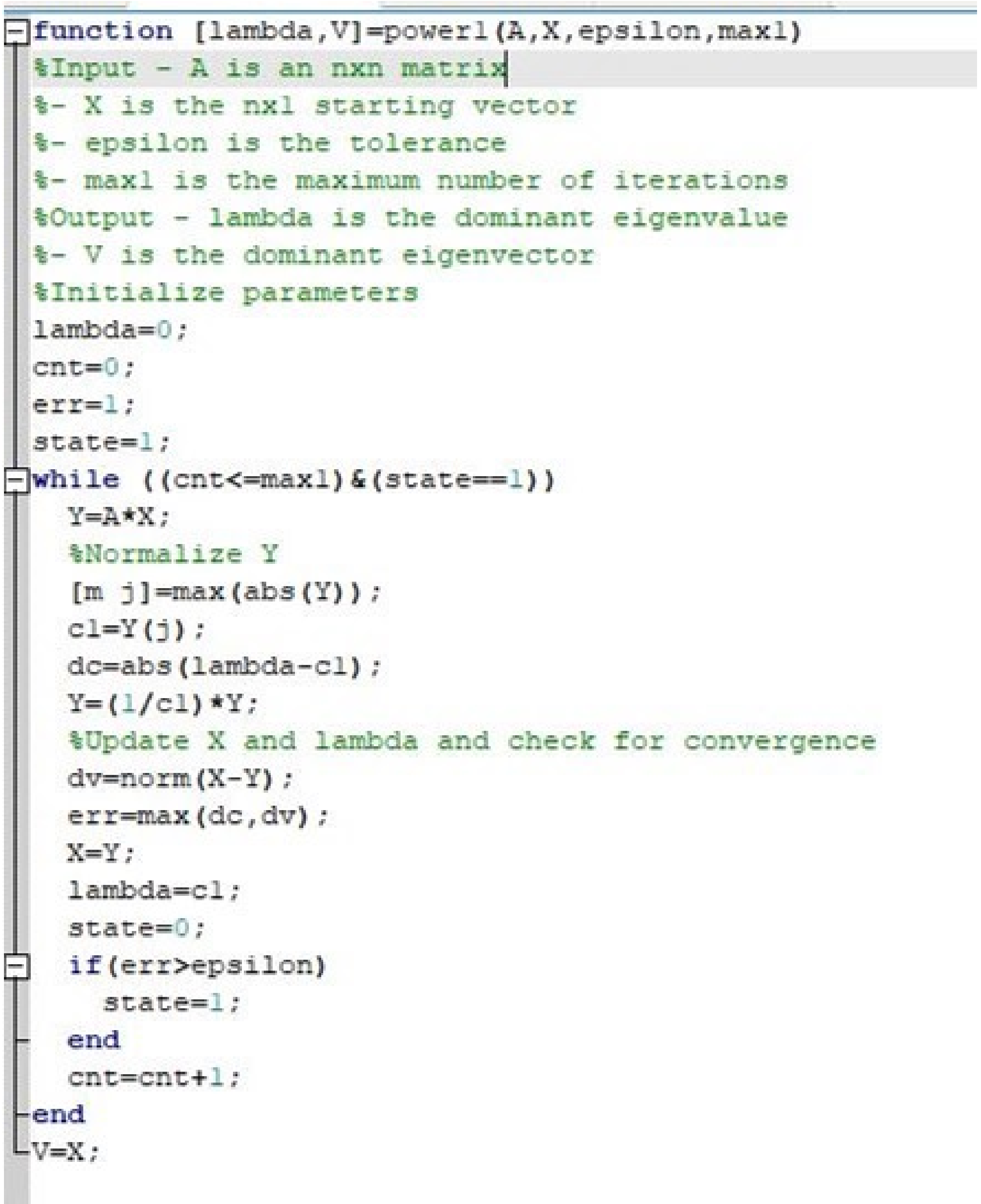
```
>> A = [1 2 3;0 -4 1;0 0 4]
A =

   1   2   3
   0  -4   1
   0   0   4

>> X = [1;1;1]
X =

   1
   1
   1

>> epsilon = 10^-5
epsilon = 1.0000e-05
>> maxl = Inf
maxl = Inf
>> [lambda, V] = power1(A, X, epsilon, maxl)
error: 'power1' undefined near line 1, column 15
>> [lambda, V] = power1(A, X, epsilon, maxl)

>> Does not converge, function never terminates
```



```
>> [lambda, V] = invpow(A, X, 0, epsilon, maxl)
lambda = 1.0000
V =

   1.0000e+00
   2.2352e-07
   2.2352e-07

>> [lambda, V] = invpow(A, X, 3, epsilon, maxl)
lambda = 4.0000
V =

   1.0000
   0.1154
   0.9231

>> [lambda, V] = invpow(A, X, -3, epsilon, maxl)
lambda = -4.0000
V =

  -4.0000e-01
   1.0000e+00
  -2.8321e-08

>> It does converge
```



```
function [lambda,V]=invpow(A,X,alpha,epsilon,max1)
%Input - A is an nxn matrix
%- X is the nx1 starting vector
%- alpha is the given shift
%- epsilon is the tolerance
%- max1 is the maximum number of iterations
%Output - lambda is the dominant eigenvalue
%- V is the dominant eigenvector
%Initialize the matrix A-alphaI and parameters
[n n]=size(A);
A=A-alpha*eye(n);
lambda=0;
cnt=0;
err=1;
state=1;
while ((cnt<=max1)&(state==1))
   %Solve system AY=X
   Y=A\X;
   %Normalize Y
   [m j]=max(abs(Y));
   c1=Y(j);
   dc=abs(lambda-c1);
   Y=(1/c1)*Y;
   %Update X and lambda and check for convergence
   dv=norm(X-Y);
   err=max(dc,dv);
   X=Y;
   lambda=c1;
   state=0;
   if (err>epsilon)
     state=1;
   end
   cnt=cnt+1;
end
lambda=alpha+1/c1;
V=X;
```



```
function [lambda,V]=power1(A,X,epsilon,max1)
%Input - A is an nxn matrix
%- X is the nx1 starting vector
%- epsilon is the tolerance
%- max1 is the maximum number of iterations
%Output - lambda is the dominant eigenvalue
%- V is the dominant eigenvector
%Initialize parameters
lambda=0;
cnt=0;
err=1;
state=1;
while ((cnt<=max1)&(state==1))
  Y=A*X;
  %Normalize Y
  [m j]=max(abs(Y));
  c1=Y(j);
  dc=abs(lambda-c1);
  Y=(1/c1)*Y;
  %Update X and lambda and check for convergence
  dv=norm(X-Y);
  err=max(dc,dv);
  X=Y;
  lambda=c1;
  state=0;
  if(err>epsilon)
    state=1;
  end
  cnt=cnt+1;
end
V=X;
```