MATH 3940-1 Numerical Analysis for Computer Scientists
Assignment 2
Due on Monday, October 19, 2020 at 1:00 pm

- You have to provide inputs and the outputs from Matlab/Octave for all questions. Also provide programs for power method and inverse power method. Hand written programs will not be accepted.
- Show all your work to receive full credit.
- You can discuss assignments with each other but do not copy them. Identical or nearly identical assignments will not be accepted.

1. Consider the system of linear equations

$$\begin{array}{rcrcrcr} 2x_1 & & & + & 3x_3 & = & 1 \\ & & x_2 & & & = & 3 \\ 3x_1 & & & + & 5x_3 & = & 1 \end{array}$$

(a) (10 marks) Use hand calculations to find the Cholesky decomposition of the coefficient matrix A and then solve the resulting triangular system.

(b) (2 marks) Use Matlab built in command to find the Cholesky decomposition of the coefficient matrix A.

Note: In Question 2 and 3 solve as it is, do not interchange equations.

2. Consider the linear system

$$\begin{array}{rcrcrcr} x & - & 5y & - & z & = & -8 \\ 4x & + & y & - & z & = & 13 \\ 2x & - & y & - & 6z & = & -2 \end{array}$$

(a) (4 marks) Starting with the zero vector, use hand calculations to perform two iterations of Jacobi method.
(b) (3 marks) Starting with the zero vector and tolerance of 10^{-5}, use Matlab to perform a maximum of 15 iterations of Jacobi method. Does it converge? If yes, how many iterations does it take to converge?

3. Consider the linear system

$$\begin{array}{rcrcrcr} 4x & + & y & - & z & = & 13 \\ x & - & 5y & - & z & = & -8 \\ 2x & - & y & - & 6z & = & -2 \end{array}$$

(a) (4 marks) Starting with the zero vector, use hand calculations to perform two iterations of Gauss-Seidel method.
(b) (3 marks) Starting with the zero vector and tolerance of 10^{-5}, use Matlab to perform a maximum of 15 iterations of Gauss-Seidel method. Does it converge? If yes, how many iterations does it take to converge?

4. Let $A = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 3 & -1 \end{bmatrix}$, and the initial approximation be $X_0 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$

(a) (9 marks) Using hand calculations, find all eigenvalues and eigenvectors of A.
(b) (2 marks) Use the Matlab built-in function to find all eigenvalues and eigenvectors of the matrix A.
(c) (4 marks) Using hand calculations, perform two iterations of the power method for matrix A starting with X_0.
(d) (3 marks) Use Matlab to find the dominant eigenvalue of A and the associated eigenvector using the power method with a tolerance of 10^{-5}, starting with X_0.
(e) (5 marks) Use Matlab to find all eigenvalues and eigenvectors of the matrix A using the inverse power method with a tolerance of 10^{-5}, starting with X_0. (Hint: take $\alpha = 0, 2.5$, and -2.5).

5. Let $A = \begin{bmatrix} 2 & 1 & -3 \\ 0 & 1 & 4 \\ 0 & 0 & -2 \end{bmatrix}$, and the initial approximation is $X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

(a) (2 marks) Use Matlab to find the dominant eigenvalue and the associated eigenvector of A using the power method with a tolerance of 10^{-5}, starting with X_0. Does it converge?
(b) (2 marks) Use Matlab to find all eigenvalues and eigenvectors of A using the inverse power method with a tolerance of 10^{-5}, starting with X_0. Does it converge? (take $\alpha = -1.5, 0.5$, and 2.5).
(c) (2 marks) Compare the performance of two methods based on your results in parts (a) and (b)? Explain the reason for their convergence/divergence.

6. Let $g(x) = -4 + 4x - \dfrac{x^2}{2}$.

(a) (4 marks) Using hand calculations, solve $x = g(x)$.
(b) (2 marks) Use Matlab to plot the functions $y = x$ and $y = g(x)$ in the same window. Your graph should show both points of intersections.
(c) (3 marks) Using hand calculations, perform 3 iterations of the fixed point method starting with $p_0 = 2.5$.
(d) (3 marks) Do you expect fixed point method to converge with an initial approximation $p_0 = 2.5$? Justify your answer using the condition of convergence.

7. Given the equation $x^3 + x^2 - 3x - 3 = 0$.

(a) (2 marks) Use Matlab built-in function to find all roots of the above equation.
(b) (6 marks) Use Matlab to perform 15 iterations of the fixed point method for the following functions, starting with $p_0 = 1$, and a tolerance of 10^{-5}. In the case of convergence, mention the number of iterations when the convergence is achieved.

(i) $g_1(x) = \dfrac{x^3 + x^2 - 3}{3}$
(ii) $g_2(x) = \sqrt[3]{3 + 3x - x^2}$
(iii) $g_3(x) = \dfrac{3x + 3}{x^2} - 1$.

MATH 3940 Assignment 2 Solutions - Matlab   Fall 2020

Question 1:

(b) >> A=[2 0 3; 0 1 0 ; 3 0 5];

>> LT=chol(A)

LT = 1.4142	0	2.1213 1
0	1.0000	0
0	0	0.7071

If the transpose of LT is L then the Cholesky decomposition will be A=(L)(LT)

Question 2:

(b) Using Jacobi method from M file, we have

>> A=[1 -5 -1; 4 1 -1; 2 -1 -6];

>> B=[-8; 13; -2];

>> [X,k]=jacobi2(A,B,[0 0 0]',10^(-5),15)

>> X = 1.0e+010 * 1.0385

-1.5282

0.0891

k = 15

Here the iterations diverge.

Question 3:

(b) Using Gauss-Siedel method from M file, we have

>> A=[4 1 -1 ;1 -5 -1 ; 2 -1 -6];

>> B=[13; -8; -2];

>> [X, k]=gauseid(A,B,[0 0 0]',10^(-5),15)

X = 3.0000

2.0000

1.0000

k = 7

The iterations converge in 7 iterations.

Question 4: (b) >> A=[-1 1 0; 1 2 1; 0 3 -1];

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

V = -0.1961 0.7071 0.3015

-0.7845 0.0000 -0.3015

-0.5883 -0.7071 0.9045

D = 3.0000 0 0

0 -1.0000 0

0 0 -2.0000

(d)
```
function [lambda, V]=power2(A,X,tol,max1)
lambda=0;
cnt=0;
err=1;
state=1;
while ((cnt<=max1)&(state==1))
   Y=A*X;
   [m j]=max(abs(Y));   %normalize 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>tol)
      state=1;
   end
```

```
    cnt=cnt+1;
end
V=X;
>> X=[1 1 2]';
>> [lambda, V]=power2(A,X,10^(-5), 35)
lambda =    3.0000
V =    0.2500
       1.0000
       0.7500
```

(e)

```
function [lambda, V]=invpower(A,X,alpha,tol, maxite )
[n n]=size(A);
A=A-alpha*eye(n);
lambda=0;
cnt=0;
err=1;
state=1;
while ((cnt<=maxite)&(state==1))
    Y=A\X;
    [m j]=max(abs(Y));
    c1=Y(j);
    dc=abs(lambda-c1);
    Y=(1/c1)*Y;
    dv=norm(X-Y);
    err=max(dc,dv);
    X=Y;
    lambda=c1;
    state=0;
    if(err>tol)
```

```
        state=1;
    end
    cnt=cnt+1;
end
lambda=alpha+1/c1;
V=X;
>> [lambda, V]=invpower(A,X,0,10^(-5),20)
lambda =   -1.0000
V =    1.0000
       -0.0000
       -1.0000
>> [lambda,V]=invpower(A,X,2.5,10^(-5),10)
lambda =    3.0000
V =  0.2500
      1.0000
      0.7500
>> [lambda,V]=invpower(A,X,-2.5,10^(-5),15)
lambda =   -2.0000
V =    0.3333
      -0.3333
       1.0000
```

Question 5: (a) >> A=[2 1 -3; 0 1 4; 0 0 -2];

```
>> [lambda V]=power2(A,[1 1 1]',10^(-5),20)
lambda =  0.8000
V =  0.8750
      1.0000
     -0.7500
>> [lambda V]=power2(A,[1 1 1]',10^(-5),21)
```

```
lambda =  5.0000

V =   1.0000

     -0.4000

      0.3000

>> [lambda V]=power2(A,[1 1 1]',10^(-5),22)

lambda =  0.8000

V =  0.8750

      1.0000

     -0.7500

>> [lambda V]=power2(A,[1 1 1]',10^(-5),23)

lambda =  5.0000

V =   1.0000

     -0.4000

      0.3000
```

We see that the values are going back and forth between 0.8000 and 5.0000, and we are not getting any convergence.

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

>> [lambda V]=invpower(A,[1 1 1]',-1.5,10^(-5),15)

lambda =  -2.0000

V =    -0.8125

        1.0000

       -0.7500

>> [lambda V]=invpower(A,[1 1 1]',0.5,10^(-5),15)

lambda =   1.0000

V =  -1.0000

      1.0000

     -0.0000

>> [lambda V]=invpower(A,[1 1 1]',2.5,10^(-5),15)

lambda =  2.0000
```

V = 1.0000

0.0000

0.0000

(c) We note that the power method fails while the inverse power method gives all the eigenvalues and eigenvectors. The reason of the failure of the power method is that A does not have a single dominant eigenvalue, both 2 and -2 have largest magnitude. Thus the power method works good in the case of a single dominant eigenvalue, however the inverse power methods works in any case with the good values of alpha.

Question 7:

(a) M- file for the function is

```
function y = f1(x)
y = x^3+x^2-3*x-3;
```

Using Matlab when we plot the graph we see that there are 3 real roots.

```
>> fzero('f1',0)
ans =  -1.0000
>> fzero('f1',1)
ans =   1.7321
>> fzero('f1',-2)
ans =  -1.7321
```

It is easy to use roots command in Matlab for polynomials.

```
>> p=[1 1 -3 -3]; % the coefficients of the polynomial
>> roots(p)
ans =    1.7321
        -1.7321
        -1.0000
```

(b) Using the fixed point program from M file.

(i) M-File for the function is

```
function y=g1(x)
y=(x^3+x^2-3)/3;
>> [k p err P]=fixpt('g1',1,10^-5,15)
```

k =   11

p =   -1.0000

err =   6.9942e-006

P = 1.0000

-0.3333

-0.9753

-0.9922

-0.9974

-0.9991

-0.9997

-0.9999

-1.0000

-1.0000

-1.0000

The iterations converge to -1 in 10 iterations.

(iii) M-File for the function is

```
function y = g2(x)
y = (3+3*x-x^2)^(1/3);
```

```
>> [k p err P]=fixpt('g2',1,10^-5,15)
```

k = 6

p = 1.7321

err = 3.0366e-006

P = 1.0000

1.7100

1.7331

1.7320

1.7321

1.7321

The iterations converge to 1.732 in 5 iterations.

(iii) M-File for the function is

```
function y = g3(x)
y = ((3*x+3)/(x^2))-1;
```

```
>> [k p err P]=fixpt('g3',1,10^-5,15)
maximum number of iterations exceeded
k =    15
p =   2.8748
err =  1.5264
P =    1.0000
     5.0000
    -0.2800
    26.5510
    -0.8828
    -0.5486
     3.4989
     0.1025
   314.0797
    -0.9904
    -0.9707
    -0.9067
    -0.6595
     1.3484
     2.8748
```

The iterations do not converge.