MATH 3940-1 Numerical Analysis for Computer Scientists
Assignment 2
Due in class on Wednesday, October 9, 2019

- You have to provide Matlab/Octave Sheets for any program used, inputs and the outputs. 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 linear system

$$\begin{array}{rcrcrcr} x & + & 2y & - & z & = & 0 \\ 2x & + & 8y & - & 4z & = & 6 \\ -x & - & 4y & + & 3z & = & -2 \end{array}$$

(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^{-6}, use Matlab to perform a maximum of 35 iterations of Jacobi method. Does it converge? If yes, how many iterations does it take to converge?

(c) (4 marks) Starting with the zero vector, use hand calculations to perform two iterations of Gauss-Seidel method.

(d) (4 marks) Starting with the zero vector and tolerance of 10^{-6}, use Matlab to perform a maximum of 35 iterations of Gauss-Seidel method. Does it converge? If yes, how many iterations does it take to converge?

2. Let $A = \begin{bmatrix} 2 & -7 & 0 \\ 5 & 10 & 4 \\ 0 & 5 & 2 \end{bmatrix}$, and the initial approximation is $X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

(a) (9 marks) Using hand calculations, find the eigenvalues and eigenvectors of A.

(b) (2 marks) Use the Matlab built-in function to find all the 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 shifted-inverse power method with a tolerance of 10^{-5}, starting with X_0. (take $\alpha = 1.5, 4.5,$ and 6.5).

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

   (a) (6 marks) Using hand calculations, find the eigenvalues and eigenvectors of A.

   (b) (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.

   (c) (3 marks) Use Matlab to find all eigenvalues and eigenvectors of A using the shifted-inverse power method with a tolerance of 10^{-5}, starting with X_0. (take $\alpha = 0, 4$, and -4).

   (d) (2 marks) What is your conclusions about the performance of power and shifted-inverse power methods. Explain the reason for their convergence/divergence.

4. Let $g(x) = \dfrac{x^2}{4} + \dfrac{5x}{4} - 3$.

   (a) (4 marks) Using hand calculations, solve $x = g(x)$.

   (b) (3 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, find 3 iterations of the fixed point method starting with $p_0 = -0.25$.

   (d) (3 marks) Do you expect fixed point method to converge with an initial approximation $p_0 = -0.25$? Justify your answer using the condition of convergence.

   (e) (3 marks) Use Matlab to perform 40 iterations of the fixed point method to solve $x = g(x)$, starting with $p_0 = -0.25$, and a tolerance of 10^{-5}.

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

   (a) (2 marks) Use the Matlab built-in function to find all roots of the above equation.

   (b) (6 marks) Use Matlab to perform 25 iterations of the fixed point method for each of 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) $g_1(x) = \sqrt{\dfrac{3 + 3x - x^2}{x}}$

   (ii) $g_2(x) = -1 + \dfrac{3x + 3}{x^2}$

   (iii) $g_3(x) = \dfrac{x^3 + x^2 - x - 3}{2}$.

MATH 3940 Assignment 2 Solutions Fall 2019

Question 1: (b) M file for Jacobi method is

```
function [X,k]=jacobi2(A,B,P,tol,maxite)
%A is  an N X N nonsingular matrix, B is  an N X 1 nonsingular matrix,
% P is  an N X 1 nonsingular matrix initial guess, tol is tolerance for P
N=length(B);
for k=1:maxite
   for j=1:N
      X(j)=(B(j)-A(j,[1:j-1,j+1:N])*P([1:j-1,j+1:N]))/A(j,j);
   end
   error=abs(norm(X'-P));
   relerr=error/norm(X);
   P=X';
    if (error<tol)|(relerr<tol)
    break
 end
end
X=X';

>> A=[1 2 -1; 2 8 -4; -1 -4 3];
>> B=[0 6 -2]';
>>[X,k]=jacobi2(A,B,[0 0 0]',10^(-6),35)

X =   1.0e+04 *
   6.5540
   2.5961
  -4.0179
k =     35
```

The iterations diverge.

(d) M file for Gauss-Siedel method is

```
function [X,k]=gauseid(A,B,P,tol,maxite)
%A is  an N X N nonsingular matrix, B is  an N X 1 nonsingular matrix,
% P is  an N X 1 nonsingular matrix initial guess, tol is tolerance for P
Digits=8;
N=length(B);
for k=1:maxite
   for j=1:N
      if j==1
         X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
      elseif j==N
         X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1))')/A(N,N);
      else
         %X contains the kth approximations and P the (k-1)st
      X(j)=(B(j)-A(j,1:j-1)*( X(1:j-1))'-A(j,j+1:N)*P(j+1:N))/A(j,j);
   end
end
   error=norm(X'-P);
   relerr=error/norm(X);
   P=X';
```

```
    if (error<tol)|(relerr<tol)
    break
 end
end
X=X';
```

```
>> A=[1 2 -1; 2 8 -4; -1 -4 3];
>> B=[0 6 -2]';
>> [X, k]=gauseid(A,B,[0 0 0]',10^(-6),35)

X =   -3.0000
       2.0000
       1.0000
k =    33
```

The iterations converge in 33 iterations.

Question 2: (b) >> A=[2 -7 0; 5 10 4; 0 5 2];

```
>> [V D]=eig(A)
V =
    0.7035    0.7683   -0.6247
   -0.5025   -0.3293   -0.0000
   -0.5025   -0.5488    0.7809
D =
    7.0000         0         0
         0    5.0000         0
         0         0    2.0000
```

(d) M-file for power method is

```
function [lambda, V]=power2(A,X,tol, max1 )
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>tol)
      state=1;
   end
   cnt=cnt+1;
end
V=X;
```

```
>> A=[2 -7 0; 5 10 4; 0 5 2];
>> [lambda V]=power2(A,[1 1 1]',10^(-5),35)
lambda =    7.0000
V =    1.0000
      -0.7143
      -0.7143
```

(e) M-file for the inverse power method is

```
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;
   %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>tol)
      state=1;
   end
   cnt=cnt+1;
end
lambda=alpha+1/c1;
V=X;
>> A=[2 -7 0; 5 10 4; 0 5 2];
>> [lambda V]=invpower(A,[1 1 1]',1.5,10^(-5),6)
lambda =     2.0000
V =   -0.8000
      -0.0000
       1.0000
>> [lambda V]=invpower(A,[1 1 1]',4.5,10^(-5),5)
lambda =  5.0000
V =   1.0000
     -0.4286
     -0.7143
>> [lambda V]=invpower(A,[1 1 1]',6.5,10^(-5),10)
lambda =    7.0000
V =   1.0000
     -0.7143
     -0.7143
```

<u>Question 3:</u> (b) Using the program for the power method from previous question in Matlab

```
>> A=[-5 1 -2; 0 1 1; 0 0 5];
>> X=[1 1 1]' ;
>> [lambda,V]=power2(A,X,10^(-5),10)
lambda =  -6.1250
V =    1.0000
      -0.2041
      -0.8163
>> [lambda, V]=power2(A,X,10^(-5), 11)
lambda =   -4.0816
V =   0.8750
      0.2500
      1.0000
>> [lambda, V]=power2(A,X,10^(-5), 100)
lambda =  -6.1250
V =   1.0000
     -0.2041
     -0.8163
>> [lambda, V]=power2(A,X,10^(-5), 101)
lambda =  -4.0816
V =  0.8750
     0.2500
     1.0000
```

Power method is not converging as the values are oscillating between -6.1250 and -4.0816.

(c) Using the program for the power method from previous question in Matlab

```
>> [lambda,V]=invpower(A,X,0,10^(-5),10)
lambda = 1.0000
V =   0.1667
      1.0000
      0.0000
>> [lambda,V]=invpower(A,X,4,10^(-5),10)
lambda =    5
V =  -0.1750
      0.2500
      1.0000
>> [lambda,V]=invpower(A,X,-4,10^(-5),10)
lambda =   -5.0000
V =    1.0000
      -0.0000
      -0.0000
```

<u>Question 4:</u> (e) M-file for the fixed point method is

```
function [k,p,err,P] =fixpt (g, p0, tol,max1)
P(1)=p0;
for k=2:max1
   P(k)=feval(g,P(k-1));
   err=abs(P(k)-P(k-1));
   relerr=err/abs(P(k));
   p=P(k);
   if (err<tol) | (relerr<tol),
      break;
   end
```

```
    end
    if k == max1
       disp('maximum number of iterations exceeded')
    end
    P=P';
```

M- file for the function is

```
function y=g416(x)
y=(x^2+5*x-12)/4;
>> [k p err P]=fixpt('g416',-0.25,10^-5,40)
k =    39
p =   -4.0000
err =  3.1016e-005
P =   -0.2500
   -3.2969
   -4.4037
   -3.6564
   -4.2282
   -3.8159
   -4.1296
   -3.8986
   -4.0735
   -3.9435
   -4.0416
   -3.9684
   -4.0234
   -3.9823
   -4.0132
   -3.9900
   -4.0074
   -3.9944
   -4.0042
   -3.9969
   -4.0024
   -3.9982
   -4.0013
   -3.9990
   -4.0007
   -3.9994
   -4.0004
   -3.9997
   -4.0002
   -3.9998
   -4.0001
   -3.9999
   -4.0001
   -3.9999
   -4.0000
   -4.0000
   -4.0000
   -4.0000
   -4.0000
```

Just a note that the convergence is achieved in 38 iterations.

Question 5:

(a) 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
```

Alternatively, using Matlab we see from the graph that there are 3 real roots. M- file for the function is

```
function y = f516(x)
y = x^3+x^2-3*x-3;
>>fzero('f516',0)
ans =  -1.0000
>> fzero('f516',1)
ans =   1.7321
>> fzero('f516',-2)
ans =  -1.7321
```

(b) (i) M file for the function is

```
function y = g5116(x)
y = ((3+3*x-x^2)/x)^(1/2);
end
>> [k p err P]=fixpt('g5116',1,10^-5,25)
k =   23
p =   1.7320
err =   1.3480e-05
P =   1.0000
   2.2361
   1.4511
   1.9017
   1.6358
   1.7883
   1.6998
   1.7508
   1.7213
   1.7383
   1.7285
   1.7341
   1.7309
   1.7327
   1.7317
   1.7323
   1.7319
   1.7321
   1.7320
   1.7321
   1.7320
   1.7321
   1.7320
```

Converge to 1.732 in 22 iterations

(ii) M fiel for the function is

```
function y = g5216(x)
y = -1+(3*x+3)/(x^2);
[k p err P]=fixpt('g5216',1,10^-5,25)
maximum number of iterations exceeded
k =    25
p =    4.8961
err =    5.3990
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
    0.4066
   24.5293
   -0.8727
   -0.4986
    5.0500
   -0.2883
   24.6873
   -0.8736
   -0.5029
    4.8961
```

The iterations do not converge.

(iii) M file for the function is

M-File for the function is

```
function y = g5316(x)
y = (x^3+x^2-x-3)/2;
>> [k p err P]=fixpt('g5316',1,10^-5,25)
k =     3
p =    -1
err =     0
P =     1
    -1
    -1
```

Converge to -1 in 2 iterations.