MATH 3940-1 Numerical Analysis for Computer Scientists
Assignment 3

Due on Monday, November 2, 2020 at 1:00 pm

- You have to provide inputs and the outputs from Matlab/Octave for all questions. Also provide programs when asked. Hand written programs, inputs and outputs 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 equation: $x^7 + 8x^6 + 16x^5 = 0$.

   (a) (4 marks) Solve the equation using hand calculations. What is the order of each root of the equation?

   (b) (4 marks) Find an interval $[a, b]$ such that the bisection method can be used to find a solution of the equation. Justify your answer.

2. Consider the equation: $3x^3 - x^2 + 2 = 0$.

   (a) (5 marks) Using hand calculations, perform 3 iterations of the bisection method starting with the interval $[-1, 0]$.

   (b) (4 marks) Using hand calculations, perform 2 iterations of the regula falsi method starting with the interval $[-1, 0]$.

   (c) (4 marks) Using hand calculations, perform 3 iterations of the secant method starting with the interval $[-1, 0]$.

3. Consider the equation: $e^x + 2x - 2 = 0$

   (a) (2 marks) Use the Matlab built-in function to find the root near 0.

   (b) (2 marks) Use Matlab to perform 20 iterations of the bisection method with initial values $a = 0$ and $b = 1$ and tolerance 10^{-5}.

   (c) (2 marks) Use Matlab to perform 20 iterations of the regula falsi method with initial values $a = 0$ and $b = 1$, tolerance 10^{-5} and epsilon= 10^{-7}.

   (d) (2 marks) Use Matlab to perform 20 iterations of the secant method starting with the initial values $p_0 = 0$, and $p_1 = 1$, tolerance= 10^{-5} and epsilon= 10^{-7}.

   (e) (2 marks) Use Matlab to perform 20 iterations of Newton's method with the initial approximation $p_0 = 0$, tolerance= 10^{-5} and epsilon= 10^{-7}.

   (f) (2 marks) Based on your results from parts (b) - (e), which method is more successful. Explain your answer using the convergence rates.

4. Consider the equation: $\tan x = x$

(a) (3 marks) Use the Matlab built-in function to find the root near -2. Do you get the correct solution?

(b) (4 marks) Find the order of the root $x = 0$? Justify your answer.

(c) (5 marks) Using hand calculations, perform 2 iterations of Newton's method starting with the initial approximation $p_0 = -1$.

(d) (3 marks) Use Matlab to perform 30 iterations of Newton's method with the initial approximation $p_0 = -1$, tolerance$= 10^{-7}$ and epsilon$= 10^{-15}$. Is the convergence linear or quadratic?

(e) (4 marks) Using the method discussed in class modify Newton's iterations to accelerate the convergence. Repeat part (d) with the modified iterations using Matlab. Do you get faster convergence? (Provide program for modified Newton's method)

5. Consider the system of nonlinear equations

$$\begin{aligned} 2x &+ 2y &= 3 \\ 3x^2 &+ 2y &= 4 \end{aligned}$$

(a) (6 marks) Using hand calculations find the exact solutions.

(b) (4 marks) Using hand calculations, perform 2 iterations of Gauss-Seidel method starting with the initial values $x_0 = 0$ and $y_0 = 0$.

(c) (4 marks) Use Jacobi method with $x_0 = 0$ and $y_0 = 0$, tolerance$= 10^{-5}$, perform 10 iterations using Matlab. Does it converge? If yes, how many iterations does it take to converge? (Provide program for Jacobi method).

(d) (3 marks) Use Gauss-Seidel method with $x_0 = 0$ and $y_0 = 0$, tolerance$= 10^{-5}$, perform 10 iterations using Matlab. Does it converge? If yes, how many iterations does it take to converge?

(e) (6 marks) Explain the reason for the convergence/divergence in parts (c) and (d) using the conditions of convergence.

MATH 3940 Assignment 3 Solutions - Matlab Fall 2020

Question 3: (a) M-file for the given function is:

```
function y = fq211x)

y = exp(x)+2*x-2;
```

Using Matlab built-in function fzero, we find:

```
>> fzero('fq211',0)

ans =    0.31492305784541
```

(b)
```
>> [c k err yc]=bisect('fq211',0,1,10^(-5),20)

c =    0.3149

k =    17

err =  7.6294e-006

yc = -2.4436e-006
```

(c)
```
>> [c,k,err,yc]=regula('fq211',0,1,10^(-5),10^(-7),20)

c =    0.3149

k =    7

err =    0.3425

yc = -1.8012e-006
```

(d)
```
>> [p1, err, k, y]=secant('fq211',0,1,10^(-5),10^(-7),20)

p1 =    0.3149

err =  8.9661e-008

k =    5

y = -3.9320e-012
```

(e) M- file for the function and its derivative are:

```
function y = fq211(x)

y = exp(x)+2*x-2;
```

```
function y = dfq211(x)

y = exp(x)+2;

>> [p err k y]=newton('fq211', 'dfq211', 0, 10^(-5), 10^(-7),20)

p =   0.3149

err =  6.9226e-005

k =    3

y =  3.2832e-009
```

(f) We see that Bisection method converges in 17 iterations, Regula Falsi method converges in 7 iterations, Secant method converges in 5 iterations, and Newton's method converges in 3 iterations. Thus the Newton's method is more successful because it converges faster than other methods. It converges quadratically as the root is a simple root because f'(0.3149) is not equal to zero.

<u>**Question 4:**</u> (a) M-file for the function is

```
function y = fq311(x)

y = tan(x)-x;

>> fzero('fq311',-2)

ans =   -1.5708
```

Substituting x=-1.5708 into the equation tan x –x=0, we see that it is not a solution.

(d) M- file for the function and its derivative are:

```
function y = fq311(x)

y = tan(x)-x;

function y = dfq311(x)

y = (sec(x))^2-1;

>> [p, err, k, y]=newton('fq311', 'dfq311', -1, 10^(-7), 10^(-15), 30)

p =  -1.0633e-05

err = 5.3166e-06

k =  29

y =  -4.0074e-16
```

By looking at the values and error, we see that we obtain convergence in 26 iterations and the convergence rate is linear. Note that here 0 is a multiple root.

(e) M- file for the accelerated Newton Method is

```
function [p, err, k, y]=newtonacc(f, df, p0, M, tol, epsilon, max1)

for k=1 :max1

            p1=p0-M*feval(f,p0)/feval(df,p0);

            err=abs(p1-p0);

   relerr=err/abs(p1);

            p0=p1 ;

            y=feval(f,p0);

            if (err<tol) | (relerr<tol) | (abs(y)<epsilon) , break, end

end

p=p1;

>> [p, err, k, y]=newtonacc('fq311', 'dfq311', -1, 3, 10^(-7), 10^(-15), 30)

p =  -1.4171e-07

err = 0.0081

k =    3

y =  -9.5291e-22
```

Convergence is much faster now since here the convergence rate would be quadratic.

<u>**Question 5:**</u> **(c)** M-file for Jacobi method for nonlinear system is

```
function [P, k] = jacobinl(G,P0,tol, maxite)

N=length(P0) ;

for k=1 :maxite

   for j=1:N

    X=feval(G,P0);

   end

err=norm(X-P0);

relerr=err/norm(X);

P0=X;

k=k;
```

```
if(err<tol)|(relerr<tol)
   break
end
end
P=P0';
```

M-file for the function is:

```
function Z=GQ5(X)
x=X(1); y=X(2);
Z=zeros(1,2);
Z(1)=(-2*y+3)/2;
Z(2)=(-3*x^2+4)/2;
```

```
>> [P iter] = jacobinl('GQ5',[0 0],10^-5, 10)
P =  1.0e+009 *
  -0.00000000046192  -6.79918443878832
k =    10
```

The iterations are diverging because the values are very large after 10 iterations.

(d)
```
>> [P iter] = seidel('GQ5',[0 0],10^-5, 10)
P =  1.0e+159 *
   9.13435831329221  -Inf
k =    10
```

The iterations are diverging because the values are very large after 10 iterations.