CS 323

Homework # 1: due Feb 12

Problem 1 Let $f(x) = \frac{1-\cos x}{x^2}$.
1a) Show that (Hint: L'hospital rule)

$$\lim_{h \to 0} f(h) = 1/2$$

1b) Find the rate of convergence of

$$f(h) - 1/2$$

as $h \to 0$. That is, find the constant α such that

$$\lim_{h \to 0} \frac{|f(h) - 1/2|}{h^\alpha} \quad \text{is a nonzero constant.}$$

Hint: You may use the Taylor expansion of $\cos x$ at $x = 0$.
1c) Write a Matlab program to calculate the function value $f(x)$ and error $|f(x) - 1/2|$ at $x = 0.1, 0.01, \cdots, 10^{-8}$. Fill the results in a table in the following format (by hand)

x	$f(x)$	$\lvert f(x) - 1/2\rvert$
10^{-1}	4.9958*e*-01	4.1653*e*-04
10^{-2}		
...	...	...

use the command

```
format short e
```

to show the result in scientific notation
1d) Does the error in c) converges to 0 as $x \to 0$? If yes, does it convergence at the rate shown in step c)? If no, explain why the result does not converge?

Problem 2. Find an approximation to a root of $x^3 + x - 4 = 0$.
2a) Show that $x^3 + x - 4$ has at least one root in the interval $[1, 4]$.
2b) Show that derivative of $x^3 + x - 4$ is strictly positive for all $x \in \mathbb{R}$. Use

this observation to show $x^3 + x - 4$ has only one real root.

2c) Write Matlab programs for the two methods (Bisection method and Newton method) based on the model program given below. For Newton's methods, use the initial guess $x_0 = 1$. Print out the approximation given after each iteration and the total number of iterations required by each method. Note that for these methods, we will assume we have satisfied the given accuracy requirement if two successive iterates agree to the given error tolerance, i.e., $|x_{n+1} - x_n| \leq 10^{-3}$.

```
% Bisection
% a = left end point of interval containing the root
% b = right end point of interval containing the root
% tolx = error tolerance in x
% tolf = error tolerance in the function value
% N = the current iteration number
% Nmax = maximum number of iterations
% fcn.m is the name of the file containing the function
format long
a=1;
b=4;
N=1;
Nmax = 50;
tolx = .001;
tolf = 0.000001;
fa = feval('fcn', a);
fb = feval('fcn', b);
m =(a+b)/2;
fm = feval(fcn, m);
while (abs(b-a) > tolx) & (abs(fm) > tolf) & (N < Nmax)
[N,a,b,m,fm]
if fa*fm <=0;
b = m;
fb = fm;
else a = m;
fa= fm;
end
m = (a+b)/2;
fm = feval('fcn', m);
```

```
N= N+1;
end
```

To use this program, first create a *Matlab m-file* with the name `fcn`.m. Note that such a file must have the extension *.m* and must be placed in the directory from which you are running *Matlab*. For example, for the function $f(x) = x - cosx$, the contents of the file `fcn`.m would be:

```
function f = fcn(x)
f = x - cos(x);
```

Problem 3. The iteration $x_{n+1} = 2 - (1+c)x_n + cx_n^3$ will converge to $s = 1$ for some values of c (provided x_0 is chosen sufficiently close to s). Find the values of c for which this is true. For what values of c will the convergence be quadratic?

Problem 4. Write a Matlab code for Newton's method to find a (nonzero) root of the system

$$x - x^2 - y^2 = 0, \quad y - x^2 + y^2 = 0$$

starting with $x_0 = 0.5, y_0 = 0.5$. Use `format long` and stop when both components of two successive iterates agree to 14 decimal places. Display the approximations given and the size of the functions at each iteration. To solve the ensuing linear system, e.g. $Ax = b$, use the command

```
x = A \ b;
```

Problem 5. Solve the same problem using *Matlab's* **fsolve** routine. To run this program, type

```
options = optimset('Display', 'iter');
x0 = [0.5,0.5]
[x,fval] = fsolve(@fcnns,x0,options)
```

where `fcnns` is the name of the M-file defining the functions. Note that *Matlab* expects the functions to be defined as a vector. See the *Matlab* help page for further instructions.