COMPUTER SCIENCE 349A, SPRING 2018
ASSIGNMENT #3 - 20 MARKS

DUE THURSDAY FEBRUARY 8, 2018 (11:30 p.m. PST)

This is a really large class and the logistics of grading assignments are challenging. Me and the markers require your help in making this process go smoothly. Please ensure that your assignments conform to the following requirements - any violation will result in getting a zero for the particular assignment.

- All assignments should be submitted electronically through the ConneX course website and shoud be **SINGLE PDF FILES**. No other formats will be accepted.. Handwritten answers are ok but they will need to be scanned and merged into a single pdf file together with any code examples and associated plots.
- The assignment number, student name and student number should be clearly visible on the top of every page of your assignment submission.
- **PLEASE DO NOT COPY THE ASSIGNMENT DESCRIPTION IN YOUR SUBMISSION**
- The asnwers to the questions should be in the same order as in the assignment specification.
- Some of the questions of the assignments are recycled from previous years but typically with small changes in either the description or the numbers. Any submission that contains numbers from previous years in any questions will be immediately graded with zero.
- Any assignment related email questions should have a subject line of the form CSC349A Assignment X, where X is the number of the corresponding assignment.
- The total number of points for this assignment is 20.

Question #1 - 8 marks.

(a) Determine the third order ($n = 3$) Taylor polynomial approximation for $f(x) = \ln(x+2)$ expanded about $a = -1$ and its remainder term. Leave your answer in terms of factors $(x+1)$ (that is, do not simplify). Show all your work.

(b) Use the polynomial approximation in (a) (without the remainder term) to approximate $f(-1.06) = \ln(0.94)$. Use either hand computation, your calculator or MATLAB. Give an exact answer to 6 significant digits.

(c) To 6 correct significant digits, the exact value of $f(-1.06) = \ln(0.94)$ is -0.0618754. Use this value to compute the absolute error of your computed approximation in (b).

(d) Determine a good upper bound for the truncation error of the Taylor polynomial approximation in (b) by bounding the remainder term in (a). Give at least 2 correct significant digits.

Question #2 - 6 marks.

The evaluation of

$$g(x) = \begin{cases} \frac{(1+\cos x)}{(x-\pi)^2} & x \neq \pi \\ 0.5 & x = \pi \end{cases}$$

is inaccurate in floating-point arithmetic when x is approximately equal to π (radians). For example, if $x = 3.16$, then

$$fl(g(3.16)) = 0.5908$$

using 4 decimal digit, idealized, chopping floating-point arithmetic. Note that the correct value of $g(3.16)$ is 0.499985..., so this computed approximation has a relative error of approximately 18%.

The fourth order ($n = 4$) Taylor polynomial approximation for $f(x) = \cos x$ expanded about $a = \pi$ is

$$\cos x \approx -1 + \frac{(x-\pi)^2}{2} - \frac{(x-\pi)^4}{24}$$

.

(a) Substitute the above Taylor polynomial approximation for $\cos x$ into the formula for $g(x)$, and simplify in order to obtain a polynomial approximation for $g(x)$ when $x \neq \pi$. (This polynomial approximates $g(x)$ very well when x is close to π since the Taylor polynomial approximation is very accurate when x is close to π.)

(b) Show that the above floating-point computation of $g(3.16)$ is unstable. Use the notation and the definition of stability given in Handout 7 to show this.

Hint: consider a perturbation of the data $\hat{x} = 3.16 + \varepsilon$, where $\left|\frac{\varepsilon}{3.16}\right|$ is small. Use the polynomial approximation to $g(x)$ in (a) to determine a very accurate approximation to the exact value of $\hat{r} = \dfrac{1+\cos\hat{x}}{(\hat{x}-\pi)^2}$, and show that for all small values of ε, the exact value of $\hat{r}$ is not close to the computed floating-point approximation of 0.5908.

(c) If $x = 1.41$ (radians), then

$$fl(g(1.41)) = 0.3871$$

using 4 decimal digit, idealized, chopping floating-point arithmetic. Show that this floating-point computation is stable (using the notation and the definition of stability given in Handout 7).

Question #3 - 6 Marks

(a) Write a MATLAB function M-file with header

```
function  root = Bisect ( xl , xu , eps , imax, f )
```

corresponding to the pseudocode given in Handout #8 for the Bisection method (in `xl` it is an "ell" not a "one").

The only differences from that given algorithm are the following:

- print a caption for your computed approximations by inserting the following statement just before the while statement:

```
fprintf ( ' iteration     approximation \n')
```

- print each successive computed approximation by inserting the following statement after the computation of x_r at the beginning of the while loop:

```
fprintf ( ' %6.0f %18.8f \n', i, xr )
```

- print a message to indicate that the algorithm has failed to converge in `imax` steps by replacing the last statement in the pseudocode by the following:

```
fprintf ( ' failed to converge in %g iterations\n', imax )
```

Use the function M-file Bisect to solve the following two problems (parts (b) and (c). In each case, you will need to write another MATLAB function M-file with header

```
function  y = f(x)
```

corresponding to the function of which you are computing a zero.

DELIVARABLES: A copy of your MATLAB M-file Bisect.m

(b) Use Bisect to solve the following problem, you are designing a spherical tank to hold water for a small village in a developing country. The volume that it holds can be computed as

$$V = \pi h^2 \frac{3R - h}{3}$$

where V is the volume in m^3, h is the depth of the water in the tank in meters, and R is the tank radius in meters. If $R = 4.1$, determine the depth h that the tank must be filled to in order that the tank holds 45 m^3 of water. In Bisect, use an initial interval of $[0, 4.1]$, eps $= 10^{-4}$ and imax $= 20$ (as was done in Handout 8). In MATLAB, 10^{-4} can be written as 1e-4.

DELIVARABLES: MATLAB M-file for the function f, call and output of Bisect.

(c) Use Bisect to solve the following problem, The velocity v of a falling parachutist of mass m kg is given by

$$v = \frac{gm}{c}(1 - e^{-ct/m})$$

, where $g = 9.81$ meters per second squared. For a parachutist with drag coefficient $c = 13.5$ kg/s, compute the mass m so that the velocity is $v = 40$ meters per second at time $t = 10$ seconds. In Bisect, use an initial interval of $[1, 100]$, eps $= 10^{-4}$ and imax $= 20$.

DELIVARABLES: MATLAB M-file for the function f, call and output of Bisect.