EE254-2F Exam #3 Due: Sunday 4/26 Spring 2020 Name: ___Connor McGarty___

Instructions:

Open book, open notes and open reference with the following exceptions:

1) No cell phones.
2) No email or messaging applications.
3) No direct copying from online resources. If online resources are used, they must be identified and documented. Your contribution must be easily identifiable. If in doubt, ask.

I am looking for certain keyword/concepts that demonstrate your understanding of C, programming and software development in general. You have 75 minutes to complete the exam and may only leave the room when you turn in the exam. Write clearly!

Question{01}: Briefly explain in your own words, the following terms and concepts (8):

a. What is your understanding of the purpose/use of numerical differentiation?

The purpose of numerical differentiation is to estimate derivatives (differences) in data in order to quantify and examine the way relationships between variables change in relation to one another. Numerical differentiation is useful when either an analytical solution is sufficiently difficult or even impossible to determine, or when dealing with sampled data in which there is no apparent formula/model to describe it.

b. What is your understanding of the difference between numerical differentiation of a function vs. data (ex: from a sensor, scope, etc…)?

Numerical differentiation of a function relies on choosing a domain to differentiate over and providing points of the variable you are differentiating with respect to, and then evaluating the function at those points using some numerical method for differentiation.

When differentiating data, the numerical differentiation techniques can be applied directly to the sampled data, since the underlying function driving the data may be unknown.

c. What are practical application examples of a single and double numerical derivative (beyond the traditional Calculus II/III answers).

Single: Current across a capacitor: $i = C\frac{dv}{dt}$

Double: 2nd-order RLC circuit: $L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C}i = 0$

d. From a prior Electrical and Computer Engineering course you have taken, give an example of where you have either a) used numerical differentiation or b) could have used it if you had the tools we have covered.

Question{02}: In the context of numerical differentiation, we have discussed the concept of error and specifically how it relates to h (step size). Answer the following questions using O(h) and the terminology used in the numerical differentiation chapter. (8)

a. In general, what happens as h gets smaller?

In general, as h (step size) decreases, error also decreases.

b. What does $O(h^2)$ mean?

$O(h^2)$ means that as h decreases, error decreases quadratically. For example, if the step-size is halved (h/2), error is quartered ($h/2^2$, or h/4).

c. Using O(h), $O(h^2)$, …, $O(h^n)$ notation, compare the first methods of the forward, backwards and center finite-difference formulas for the first derivative, explaining which is a better approximation and why.

1^{st} Derivative, forward method, v1 error: O(h)
1^{st} Derivative, backward method, v1 error: O(h)
1^{st} Derivative, center method, v1 error: $O(h^2)$

Due to the forward and backward methods having truncation error of O(h) while center method has error of $O(h^2)$, for a given step size h, the center method will give a better estimate due to less error.

d. Using O(h), $O(h^2)$, …, $O(h^n)$ notation, compare the first and second method of the center finite-difference formulas for the first derivative, explaining which is a better approximation and why.

1^{st} Derivative, center method, v1 error: $O(h^2)$
1^{st} Derivative, center method, v2 error: $O(h^4)$

Since the second method has truncation error of $O(h^4)$, it is a better approximation given the same step size as the first method, since the error decreases with quartic behavior (step size is halved, error is reduced by a factor of 1/16).

Question{03}: Regarding data driven finite-difference differentiation and built in Matlab functions, given x and y vectors of length 1 x n (9):

a. Complete the following table:

Method	n_{start}	n_{end}	length of f'
Forward, first derivative, method one	1	end-1	n-1
Backward, first derivative, method one	2	end	n-1
Center, first derivative, method one	2	end-1	n-2
Forward, first derivative, method two	1	end-2	n-2
Backward, first derivative, method two	3	end	n-2
Center, first derivative, method two	3	end-2	n-4

b. Complete the following table:

Method	n_{start}	n_{end}	length of f'
diff (symbolic)	1	end	n
diff	2	end	n-1
gradient	1	end	n

Question{04}: Use P3 to solve Problem 21.19 from Chapra.(12)

The objective of this problem is to compare second order accurate forward, backward, and centered finite-difference approximations of the first derivative of a function to the actual value of the derivative. This will be done for

$$f(x) = e^{-2x} - x$$

(a) Use calculus to determine the correct value of the derivative at x = 2.

$$\frac{d}{dx}(e^{-2x} - x) = \frac{d}{dx}(e^{-2x}) - \frac{d}{dx}(x)$$

$$\frac{d}{dx}(e^{-2x}) = -2e^{-2x}$$

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(e^{-2x} - x) = -2e^{-2x} - 1$$

Value of $\frac{d}{dx}(e^{-2x} - x)$ at $x = 2$, is -1.9817.

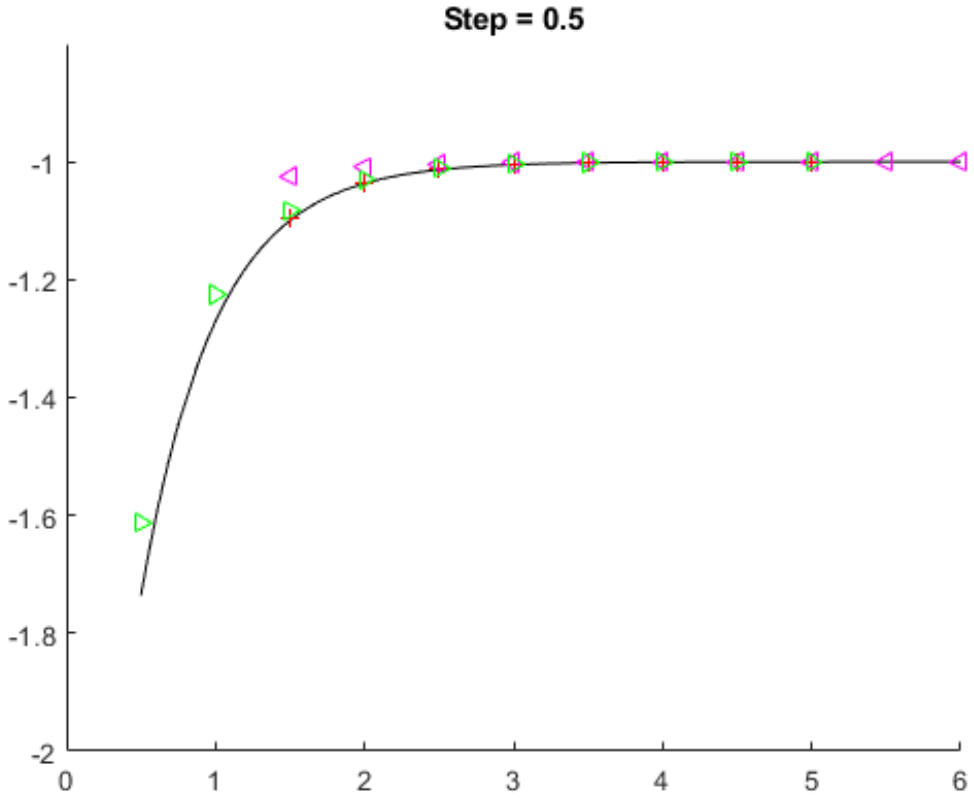
(b) Develop an M-file function to evaluate the centered finite-difference approximations, starting with x = 0.5. Thus, for the first evaluation, the x values for the centered difference approximation will be x = 2 ± 0.5 or x = 1.5 and 2.5. Then, decrease in increments of 0.1 down to a minimum value of Δx = 0.01.

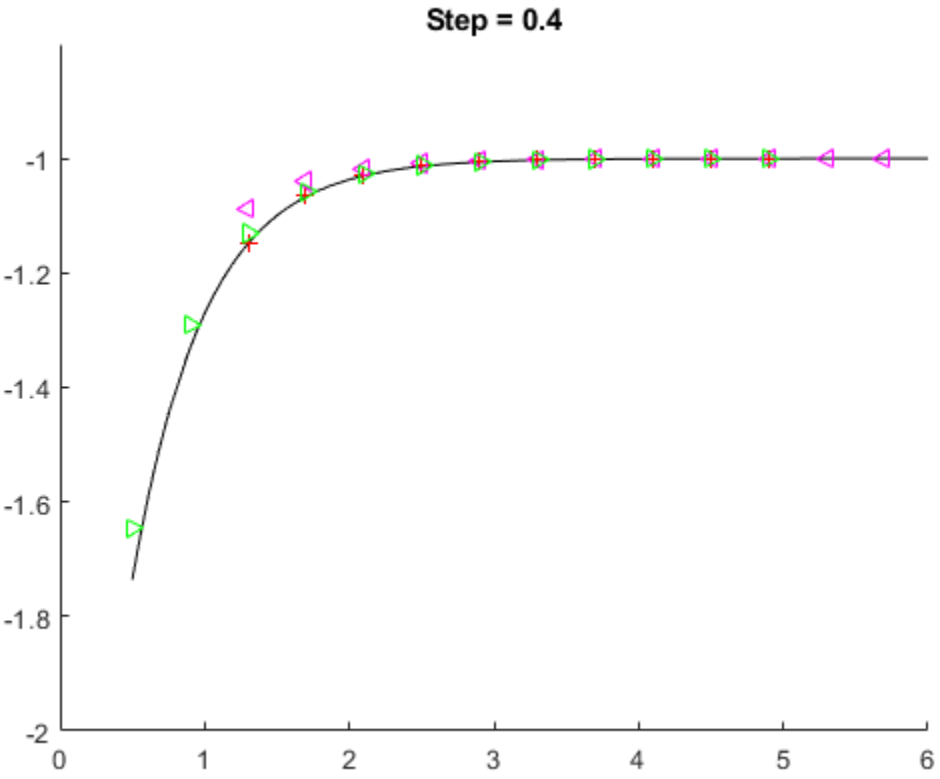
(c) Repeat part (b) for the second-order forward and backward differences. (Note that these can be done at the same time that the centered difference is computed in the loop.)

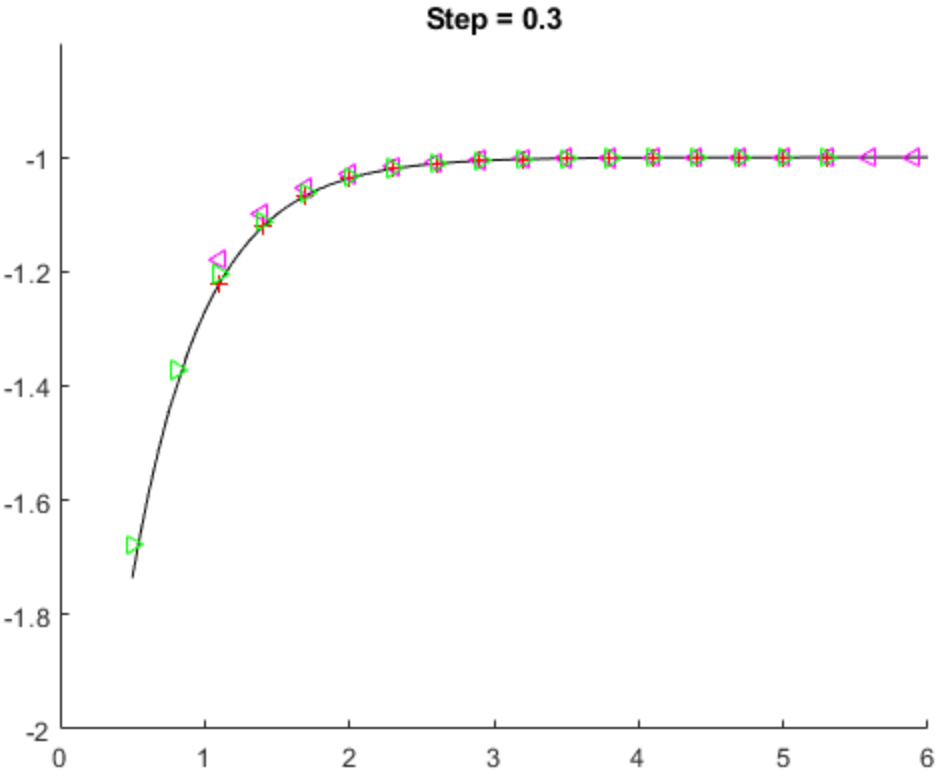
(d) Plot the results of (b) and (c) versus x. Include the exact result on the plot for comparison.

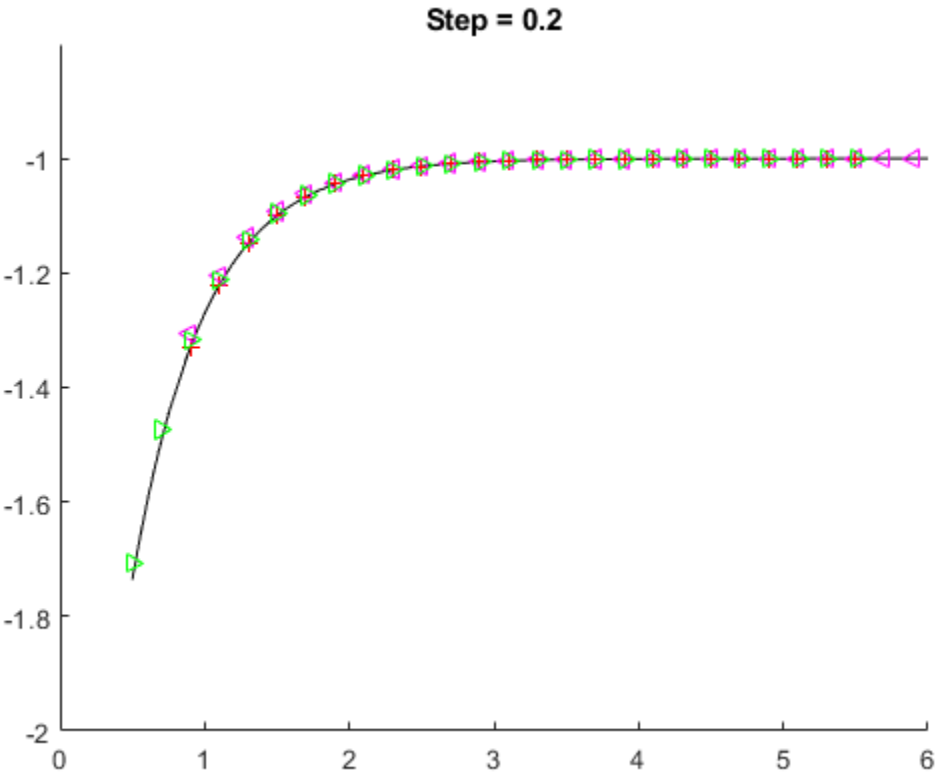
```
close all; clear; clc;
syms x;
y = exp(-2*x) - x;



%fplot(y)
xmin = 0.5;
xmax = 6;
xdelta = [.5 .4 .3 .2 .1 .01];
index = 0;

for step = xdelta
    index = index + 1;

    back{index} = backward(y, xmin, xmax, step);
    cent{index} = center(y, xmin, xmax, step);
    forw{index} = forward(y, xmin, xmax, step);

    figure;
    ylim([-2 -.8]);
    hold on;
    fplot(diff(y),[xmin xmax], 'k');
    if (index == length(xdelta))
        plot([xmin+(2*step):step:xmax-(2*step)], cent{index}, 'r');
        plot([xmin+(2*step):step:xmax], back{index}), 'o';
        plot([xmin:step:xmax-(2*step)], forw{index}), 'g';
    else
        plot([xmin+(2*step):step:xmax-(2*step)], cent{index}, 'r+');
        plot([xmin+(2*step):step:xmax], back{index}, 'm<');
        plot([xmin:step:xmax-(2*step)], forw{index}, 'g>');
    end

    title(['Step = ', num2str(step)]);
end

% subplot(6,1,1) % xdelta(1), step == .5
% fplot(diff(y),[xmin xmax]); % actual 1st deriv
% hold on;
% plot([xmin:step(1):xmax], results{1}{1,10}, 'r+'); %center method 1
% plot([xmin:step(1):xmax-step(1)], results{1}{1,4}, 'o+');



function res = center(y, xmin, xmax, step)
    y_h = matlabFunction(y);
    x_v = [xmin:step:xmax];
```

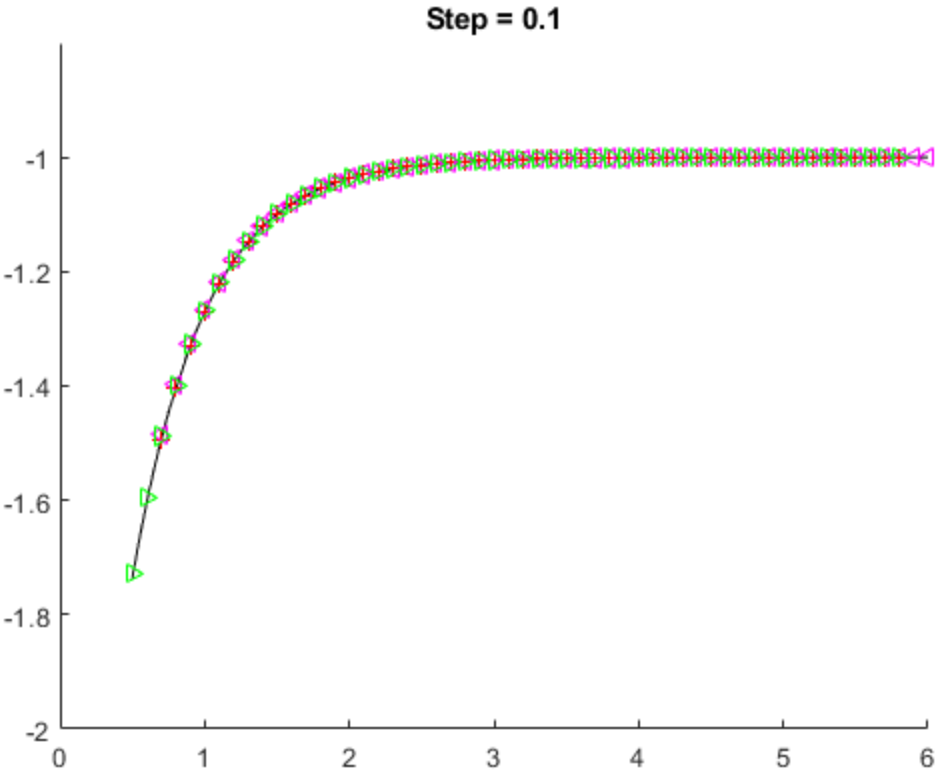
```
    res = (-1.*(y_h(x_v(5:end))) + 8.*(y_h(x_v(4:end-1))) ...
        - 8.*(y_h(x_v(2:end-3))) + (y_h(x_v(1:end-4)))) ./ (12*step);
end

function res = backward(y, xmin, xmax, step)
    y_h = matlabFunction(y);
    x_v = [xmin:step:xmax];

    res = (3.*y_h(x_v(3:end)) - 4.*y_h(x_v(2:end-1)) + y_h(x_v(1:end-2))) ./
(2*step);
end

function res = forward(y, xmin, xmax, step)
    y_h = matlabFunction(y);
    x_v = [xmin:step:xmax];

    res = (-y_h(x_v(3:end)) + 4.*(y_h(x_v(2:end-1))) - 3.*(y_h(x_v(1:end-
2)))) ./ (step*2);
end
```

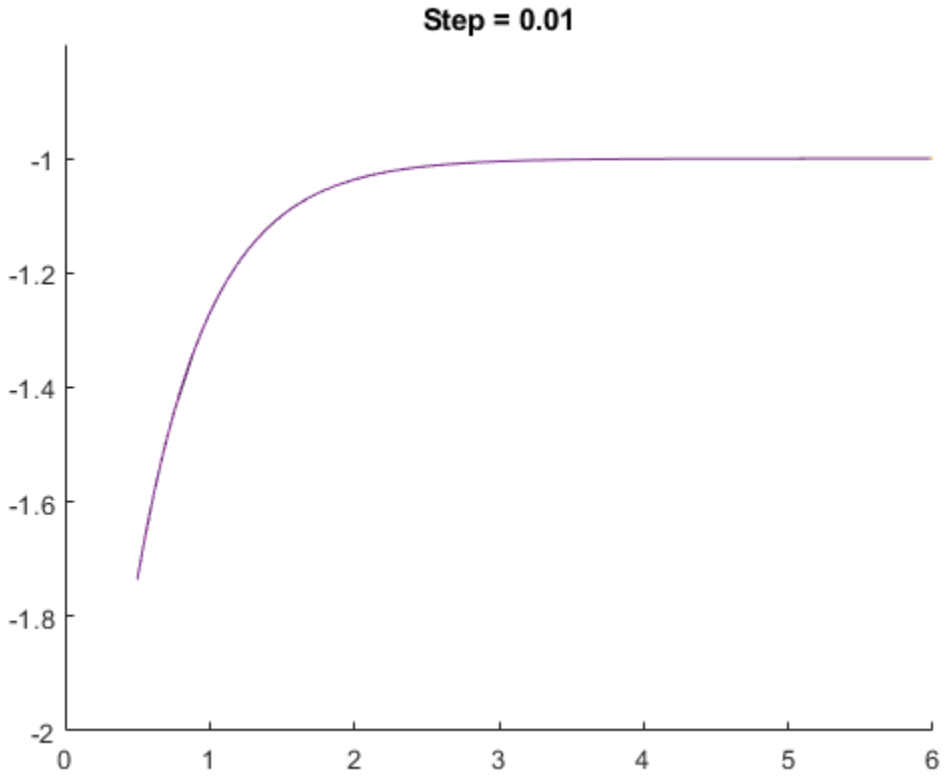



Step = 0.4
-1
-1.2
-1.4
-1.6
-1.8
-2
0
1
2
3
4
5
6


Step = 0.3
-1
-1.2
-1.4
-1.6
-1.8
-2
0
1
2
3
4
5
6


Step = 0.2
-1
-1.2
-1.4
-1.6
-1.8
-2
0
1
2
3
4
5
6


Step = 0.1
-1
-1.2
-1.4
-1.6
-1.8
-2
0
1
2
3
4
5
6


Step = 0.01
-1
-1.2
-1.4
-1.6
-1.8
-2
0
1
2
3
4
5
6