**ChE 310 – Computational Methods in Chemical Engineering**

**Spring 2019**

**Midterm Exam #1 KEY**

**NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

By writing my name, I certify that I have abided by all  
academic honesty policies.

* This portion of the exam is closed book, closed course notes. No additional resources may be consulted to complete this portion of the exam.
* This portion of the exam is worth 20 points (2 per problem).
* Write your answers to be graded in the space provided.
* You must turn in this portion of the exam before you will receive the free response portion, which is worth an additional 60 points.

1.1

Consider an array A in MATLAB. **Write a MATLAB command** that will store *all elements* of the *third column* of A to a new variable B.

>> B = A(:,3)

1.2

Dr. Roling is 6.3 feet tall. When he measures himself 5 times with a yardstick, he records values (in feet) of 7.3, 7.2, 7.2, 7.3, 7.2. **Which statement is most correct** about his method of measuring (**circle one**)?

1. The method is precise and accurate.
2. **The method is precise but not accurate.**
3. The method is accurate but not precise.
4. The method is neither accurate nor precise.

1.3

We discussed several types of cubic spline constraints in class. **What specific constraints are specified when constructing a *clamped cubic spline*,** compared to other cubic splines? (**Answer below**.)

Clamped cubic splines specify the first derivatives at the endpoints of the data set.

1.4

Consider a probability density function and a cumulative density function . Which of the following is ***true***? (**Circle one**)

1. For all ,
2. For all ,
3. For all ,

1.5

Consider the following MATLAB code. After running the code, **what is the stored value of a?**

a = 5; b = 10; c = 15;

if b + c >= 20 || b > c

a = 0;

elseif b < c && c > 10 a = \_\_\_0\_\_\_\_

a = 10;

else

a = 2\*a;

end

1.6

Consider the following MATLAB code. After running the code, **what is the stored value of a?**

x = 0; y = 1;

f = @(x) x + y;

x = 3; y = 4; a = \_\_\_4\_\_\_\_

a = f(x);

1.7

Consider the following function file, saved as my\_sum.m:

function sum\_out = my\_sum(a,b)

c = a + b

sum\_out = c;

end

I execute the following MATLAB script, saved as my\_script.m:

clear

x = 1; y = 2;

z = my\_sum(x,y);

**List below all variables that are stored in the MATLAB workspace environment** after running my\_script.m.

**Variables: \_\_\_\_\_\_\_\_\_x,y,z\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1.8

Consider the following code. After running, **what is the value of the array A**?

A = zeros(3,3);

for ii = 1:2

for jj = 2:3

A(ii,jj) = 1;

end

end A = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.9

We want to calculate the following series to a relative error tolerance of 10-5:

**List below two problems** with the following MATLAB code that will lead to an error or incorrect results.

clear

tol = 1e-5; err = tol;

total = 0; n = 1;

while err > tol

total\_bk = total;

total = total + 1/n.^2;

err = abs(total - total\_bk)/total;

end

**Problems:**

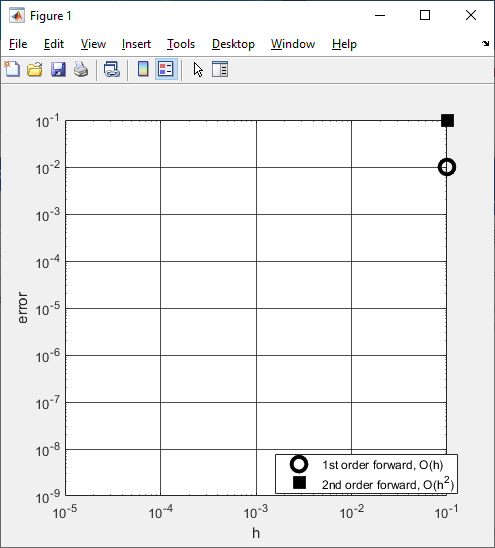
***n not updated within loop***

***setting err = tol will cause while loop to exit immediately before running***

1.10

Dr. Reuel uses a first-order forward finite difference formula (truncation error ) and a second-order finite difference formula (truncation error ) to evaluate a function’s derivative using a step size . He then plots the error vs. step size (see plot and legend below).

***Sketch on the plot***how the error for each method is expected to change as a function of step size*.* (***Draw two total lines/curves***, no need to draw symbols. Note that this is a log-log plot.)



**Free Response 1:** Consider the following data table, which samples elevation (in feet) as a function of latitude and longitude in the State of Colorado.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Longitude (degrees west) | | | | |
|  |  | 109 | 108 | 107 | 106 | 105 |
| Latitude (degrees north) | 40 | 5600 | 6300 | 9800 | 9100 | 5200 |
| 39 | 5800 | 9800 | 10200 | 9300 | 9200 |
| 38 | 7500 | 8300 | 12400 | 7600 | 7500 |
| 37 | 4700 | 6600 | 8300 | 8000 | 9100 |

1. **In the first panel of a 1 x 2 subplot, generate a 3-D surface plot of the elevation as a function of position** (latitude and longitude).
2. Based on these data, **determine the elevation of Alamosa, CO,** which is located at 37.5degrees north, 105.9 degrees west.
3. The city of Cortez, CO is located at 37.5 degrees north, 108.6 degrees west. **In the second panel of the   
   1 x 2 subplot, plot the elevation profile from Alamosa (105.9 °W) to Cortez (108.6 °W) as a function of the longitude** (noting that the latitude is constant 37.5 °N from Alamosa to Cortez).

%2019S 310 Exam 1 [m-file with header 2pts]

%Free Response #1

%Coded by Luke T. Roling on 2.21.19

clear; clf;

%Input x, y, z data (2)

x = 109:-1:105;

y = 40:-1:37;

elev = [5600 6300 9800 9100 5200;

5800 9800 10200 9300 9200;

7500 8300 12400 7600 7500;

4700 6600 8300 8000 9100];

%Set up meshgrid (2)

[X,Y] = meshgrid(x,y);

%Correct use of subplot (2)

subplot(1,2,1)

%Correct surface plot (2)

surf(X,Y,elev)

xlabel('longitude, deg W')

ylabel('latitude, deg N')

zlabel('altitude, ft')

%Use of interp2 as solution tool (2)

%Correct Alamosa elevation (2)

alt\_alamosa = interp2(X,Y,elev,105.9,37.5)

%Create lat/long input arrays (2)

xx = linspace(108.6,105.9);

yy = 37.5\*ones(size(xx));

%Apply interp2 to create height @ fixed latitude (2)

zz = interp2(X,Y,elev,xx,yy);

subplot(1,2,2)

%plotting of output data (2)

plot(xx,zz)

xlabel('longitude, deg W')

ylabel('altitude, ft')

**Free Response 2:** Your chemical process relies on a natural water source as a cooling stream. The water temperature varies. You sample the water temperature over the year and record the temperature (°C) in ‘WaterTemp.xlsx’ (on course website). Answer the following based on this data set:

1. What is the average temperature and standard deviation?
2. Plot a histogram of the water temperatures with **15** bins. Label axes.
3. Your process requires cooling water < 23°C. What percentage of the year will the natural stream be too warm to use without additional cooling?
4. What assumptions go into your prediction of part 3?

% 2019S 310 Exam 1 [Correct header usage 3 pts]

% Free Response #2

% Coded by Nigel F. Reuel on 2.21.19

%

% Read data [2 pts]

D = xlsread('WaterTemp');

% Report average and standard deviation [2 pts]

Avg = mean(D)

StDev = std(D)

% Plot histogram with 15 bins [4 pts]

nbins = 15;

histogram(D,15);

% Label the axes [2 pts]

xlabel('Temperature (degrees C)')

ylabel('Frequency')

% Determine percentage out of specification [4 pts]

% METHOD 1, random number generator

Num = normrnd(Avg,StDev,1e8,1);

Count = sum(Num>23); % number out of spec.

% Percentage out of spec:

Percentage\_above\_23 = Count/1e8\*100

% METHOD 2, cumulative distribution

pd = makedist('Normal','mu',Avg,'sigma',StDev)

y = cdf(pd,23);

Percentage\_above\_23 = (1-y)\*100

% What assumptions go into this calculation? [3 pts - full credit if they gave some of the following ideas]

disp('This assumes that the samples measured represent the true population (large enough, no sample bias, etc.)')

disp('This also assumes that the population is best fit by a normal distribution.')

disp('If method 1 is used, this assumes you have generated enough random numbers to capture these more rare events.')

**Free Response 3:** You are part of a team to determine the path of a new pipeline. Your surveyors provide topography data, altitude vs. position like the following (all in meters):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X position | 0 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 |
| Z height | 150 | 120 | 100 | 180 | 230 | 240 | 210 | 100 | 90 | 80 | 70 | 50 | 20 |

Pumping fluid up gradients > 0.6 (ΔZ/ΔX) is cost prohibitive. To help analyze the survey data, write a **function** that does the following:

1. Takes the X and Z data as inputs
2. Subplot 1 – plot Z vs. X data
3. Subplot 2 – plot the gradient for each point (use O(h2) finite difference for EACH point)
4. Print to the command line the **x-positions** of the pipe at which the gradient is above 0.6

Also provide a **script** that tests your function by using the data provided (grader should be able to run this and see the plots + output to the command prompt).

% 2019S 310 Exam 1 [Correct use of header 1 pts]

% Free Response #3

% Coded by Nigel F. Reuel on 2.21.19

%

% NOTE, this is the driver script. The function is below. [Driver 1 pts]

X = [0:12]\*100;

Z = [150 120 100 180 230 240 210 100 90 80 70 50 20]; %[Data 1 pts]

Topo(X,Z) %Calling function [1 pt]

% I will include my function here as a LOCAL function.

function Topo(X,Z) %[Correct function structure 2 pts]

% Create subplot 1 % [Subplot 1 with labels 2 pts]

subplot(1,2,1)

plot(X,Z,'o:')

xlabel('X (m)')

ylabel('Z (m)')

% Determine gradient at each point using O(h^2) formulas

% First point [2 pt]

h = X(2)-X(1);

S1 = (-Z(3)+4\*Z(2)-3\*Z(1))/(2\*h);

% Last point:

Se = (3\*Z(end)-4\*Z(end-1)+Z(end-2))/(2\*h); %[2 pt]

% First derivatives of all middle points

Sm = (Z(3:end)-Z(1:end-2))/(2\*h); %[2 pt]

% Assemble the vector of all the slopes

Slopes = [S1 Sm Se]; % [1 pt, note a for loop can also be used...7 pt total for finite diff]

% Plot the slopes

subplot(1,2,2) %[2 pt second subplot with labels]

plot(X,Slopes,'o:')

xlabel('X (m)')

ylabel('Slope (delZ/delX)')

% Identify the ones that are above tolerance and print to command

ind = find(Slopes>0.6); % [2 pt logic to find out of spec]

disp('Following x positions are too steep:') %[1 pt printing to screen]

X(ind)

end