%AUTHOR: DANIEL CRISP

%DESC: FINAL TEST

%filename: final.m

%% Question 1

% Write a program to animate and trace the function:

y = -2t + 1.5 for t=[-10, 10];

n = 100;

%t = -10:n:10;

t = linspace(-10, 10, n);

y = -2\*t(1:n) + 1.5;

axis([-10 10 -25 25]);

for it = 1:n

plot(t(1:it), y(1:it));

axis([-10 10 -25 25]);

title('Question 1a')

drawnow

end

%% Question 1b

Find int(y,x) from x=[0 4] using composite simpson 1/3

syms x;

fun = @(x)x\*exp(2\*x);

n = 10;

a = 0; x0 = a;

b = 4;

h = (b-a)/n;

for i=1:n

val = val + simpson\_comp(fun, x0, h);

x0 = x0 + h;

end

% using this defined in simpson\_comp.m

% function [Y1] = simpson\_comp(Y, x0, h)

% syms x;

%

% A = h\*(1/3);

% x1 = x0;

% x2 = x0+0.5\*h;

% x3 = x0+h;

%

% k1 = Y(x1);

% k2 = Y(x2);

% k3 = Y(x3);

%

% Y1 = A\*(k1 + 2\*k2 + k3);

% end

%OUTPUT: 1.1332e+04

%% Question 1c

% Find the roots, max and min for the following function using newton

% rraphson method

syms x;

fun = @(x) -2\*x^3 - 3\*x^2 + 0.5;

tolerance = 0.01;

figure(1);

ezplot(fun); title('fun = -2\*x^3 - 3\*x^2 + 0.5');

axis = ([-200 20 -20 20]);

fdiffx = diff(fun,x);

x0 = 2;

OK=1; i = 1; error = 100;

while(OK)

x1 = x0 - (subs(fun,x,x0)/subs(fdiffx,x,x0));

error = abs((x0-x1)/x1)\*100;

if(error<tolerance) OK = 0; end

i = i+1;

x0 = x1;

end

fprintf('\nGreat! Your root is %f, and has an error of %f\n\n', x0, error);

%OUTPUT:

%Great! Your root is 0.366025, and has an error of 0.03%

% NOT DONE EDITING FOR MAX/MIN

% MATLAB FUNCTION FREEZING ON 'tolerance = 0.5'...

tolerance = 0.5;

start = 2;

x0 = start;

OK=1; i = 1; error = 100;

while(OK)

x1 = x0 - (subs(df,x,x0)/subs(ddf,x,x0));

error = abs((x0-x1)/x1)\*100;

if(error<tolerance) OK = 0; end

i = i+1;

x0 = x1;

end

fprintf('\nGreat! Your max/min is %f, and has an error of %f\n\n', x0, error);

%% Question 2a

% function WCF = WindChill(T, V)

%

% %WCF = 35.7 + 0.6\*T' - 35.7\*V.^0.16 + 0.43\*T'\*V.^0.16;

% [X,Y] = meshgrid(V,T);

% xn = length(X(1,:));

% yn = length(Y(:,1));

% for xi = 1:xn

% for yi = 1:yn

% WCF(xi,yi) = 35.7 + 0.6\*Y(yi) - 35.7\*X(xi).^0.16 + 0.43\*Y(yi)\*X(xi).^0.16;

% end

% end

%

% WCF = [[0,V];T',WCF];

% disp(WCF);

% end

T = -20:5:55;

V = 0:5:55;

WCF = WindChill(T,V);

%OUTPUT:

disp(WCF)

0 0 5.0000 10.0000 15.0000 20.0000 25.0000 30.0000 35.0000 40.0000 45.0000 50.0000 55.0000

-20.0000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000 23.7000

-15.0000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000 26.7000

-10.0000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000 29.7000

-5.0000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000 32.7000

0 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000 35.7000

5.0000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000 38.7000

10.0000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000 41.7000

15.0000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000 44.7000

20.0000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000 47.7000

25.0000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000 50.7000

30.0000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000 53.7000

35.0000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000 56.7000

40.0000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000 59.7000

45.0000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000 62.7000

50.0000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000 65.7000

55.0000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000 68.7000