ENEL102, fall term 2017

Assignment 4

Solving nonlinear functions, Chapter 9

Due date: Oct 30

Donald Robert Dunn

10152653

This assignment is based on the material in section 9.1. Suggest you read through this first before attempting the assignment questions. As usual, fill in this template with your matlab code, output and analysis. Then submit your Word document on D2L.

………

Q1. Consider the single variable equation of

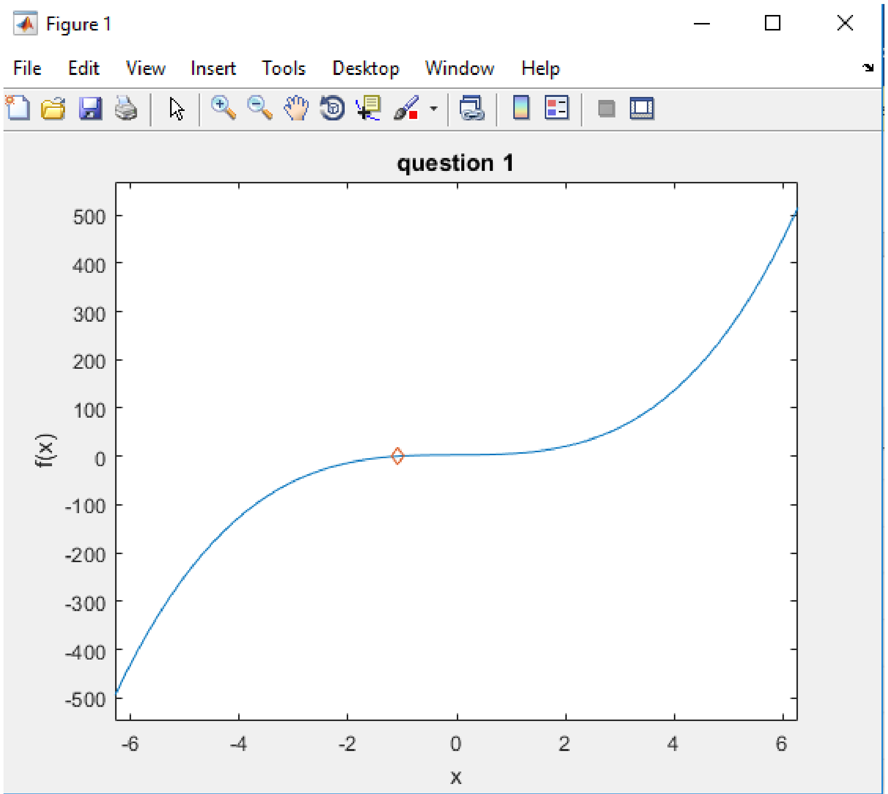
$$f(x) = \exp\left(0.45x\right) + 2x^3 + 2$$

This function is known to have a single real solution of f(x)=0. Find this solution by writing an anonymous function for f(x) and then using this in fzero() to find the solution of f(x)=0. Generate a plot of f(x) showing the root location and thereby verifying the answer generated by fzero(). List the Matlab statements used and the answer

(Matlab input)

```
f=@(x) exp(0.45*x)+2*(x^3)+2;
zero=fzero(f,0);
ezplot(f);
hold
plot(zero,0,'d')
xlabel('x')
ylabel('f(x)')
```

(Matlab Response)



……..

Q2. Consider the single variable equation of

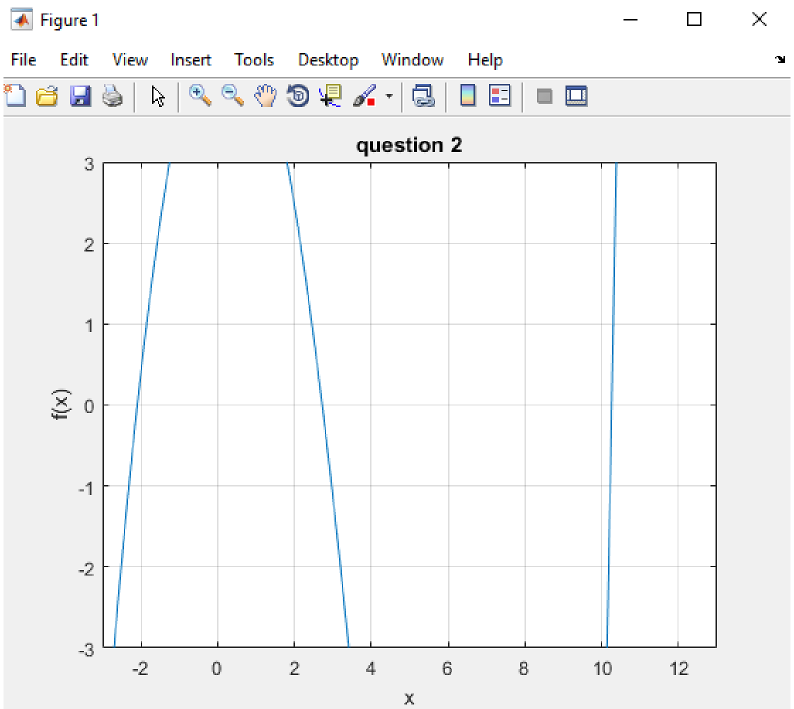
$$f(x) = \exp(0.45x) - x^2 + 4$$

This function is known to have three solutions of x that satisfy f(x)=0. That is three values of x satisfy this relation. Generate a plot such that these solutions are easily observed.

(Matlab input)

```
f=@(x) exp(0.45*x)-x^2+4;
ezplot(f,[-3,13,-3,3])
grid on
xlabel('x')
ylabel('f(x)')
title('question 2')
```

(Matlab Response)



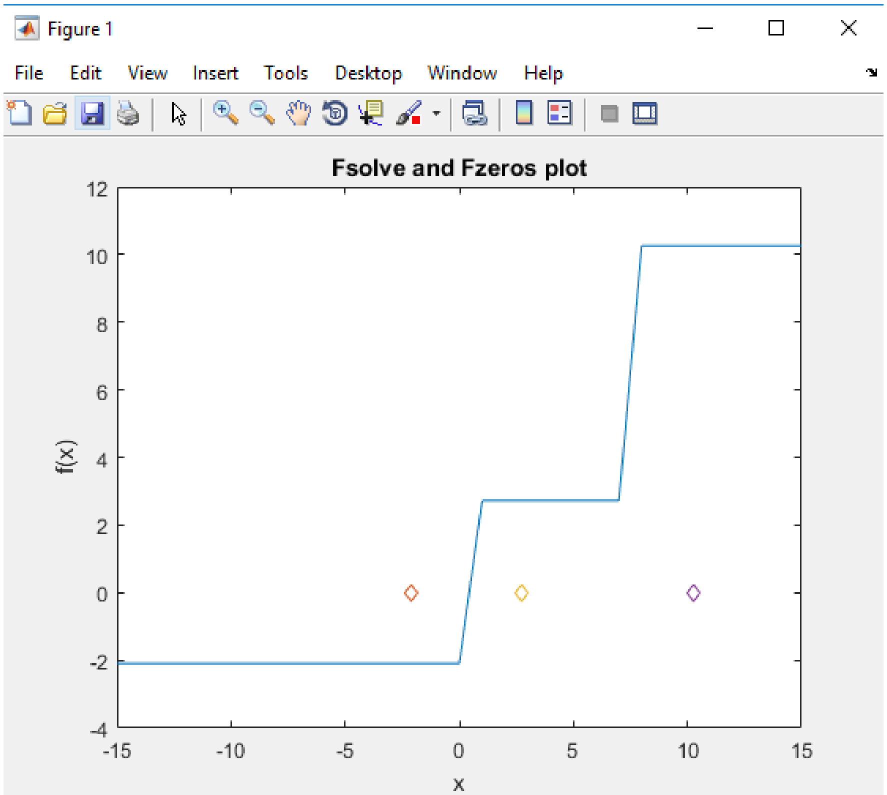
………

Q3. For the function given by **Q2** above find the three values of x that satisfy f(x)=0 using fzero(). Note that fzero() can only find one solution for a given initial guess for x. One way to ensure that we can find the three solutions is to use a set of initial conditions. Take the start points as x = -15,-14,…15 and use fsolve on each start point individually and store the point to which it converges. Show a plot of the output of fsolve as a function of the input guesses. Next superimpose the solutions of x onto a plot of f(x) verifying correctness of fzero().

(Matlab input)

```
f=@(x) exp(0.45.*x)-x.^2+4;
vector=-15:1:15;
x=fsolve(f,vector);
plot (vector,x)
hold on
x1=fzero(f,-2);
x2=fzero(f,3);
x3=fzero(f,10);
plot(x1,0,'d', x2,0,'d', x3,0,'d')
hold off
ylabel('f(x)')
xlabel('x')
title('Fsolve and Fzeros plot')
```

(Matlab Response)

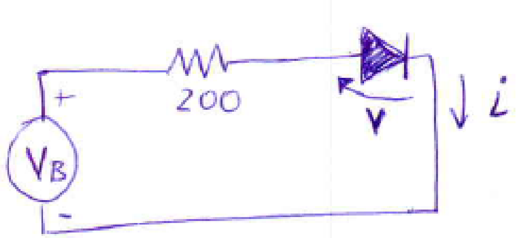


………

Q4 Consider a series circuit of a battery of voltage V_b, connected to a series circuit consisting of a diode with a current-voltage relation of

$$i = 0.001(\exp(6v) - 1)$$

and a resistor of 200 ohms. The diode is connected such that it is forward biased as shown in the diagram.



Plot the current through the diode (i) as a function of the voltage across it (v).

Now using fzero(), find the voltage across the diode when the supply battery is one volt.

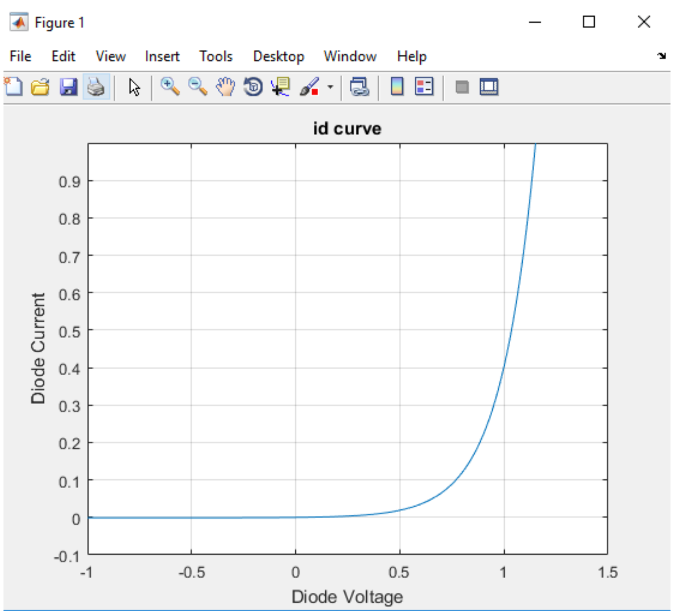
(Matlab input)

```
voltage_diode = @(v) 1-v-200*0.001*(exp(6*v)-1);
current = @(v) 0.001*(exp(6*v)-1);
ezplot(current,[-1,1.5,-0.1,1])
grid on
xlabel('Diode Voltage')
ylabel('Diode Current')
title('id curve')
v_diode = fzero(voltage_diode, 0)
```

(Matlab Response)

v_diode =

0.2582



…….

Q5 You can get information regarding the convergence of fzero() by setting the optimset() to display the output. Use this to **a)** Determine the number of function calls required to find the solution in **Q4**. **b)** Show the command for optimset that changes the function residual tolerance from the default of 1e-4 to 1e-8. **c)** Run the diode current problem again in **Q4** with this tighter tolerance and explain why no additional function calls are required for this particular problem when the function tolerance is changed.

(Matlab input)

```
fzero('200*0.001*(exp(6*x)-1) + x - 1', 0, optimset('display', 'iter',
'TolFun', 1e-8))
```

(explanation)

a) **The solution in question 4 requires 25 function calls**
b) **Adding a comma and a 1e-8 at the end of the optimset call changes the default value from 1e-4 to 1e-8**
c) **This is because the function takes 25 calls to evaluate the zeros, no matter what tolerance you give it.**

…….

Q6 Consider the intersection of a circle and line given by the functions

$$x^2 + y^2 = 1$$
$$x = y$$

By inspection, determine the two points of intersection

(sol)

My solution by hand results in $\pm(\frac{1}{2})$^(1/2)

…….

Q7 Use fsolve() to determine the two points of intersection. Show the Matlab code to do this.

(Matlab input)

```
f = @(x) [x(1).^2+x(2).^2-1; (x(1)-x(2))];
p_neg = [-2,-2];
Intersect_one = fsolve(f,p_neg)
p_pos = [2,2];
Intersect_two = fsolve(f,p_pos)
```

(Matlab Response)

Intersect_one =

-0.7071 -0.7071

Intersect_two =

0.7071 0.7071

…….

Q8 Consider a matrix with unknown coefficients of a,b,c that is given by the model of

$$A(a,b,c)=\begin{bmatrix} ab & ac \\ -ac & ab+c \end{bmatrix}$$

Lab measurements show that A has the values of

$$A=\begin{bmatrix} 4 & 3 \\ -3 & 5 \end{bmatrix}$$

Find the values of a b and c using fsolve().

(Matlab input)

```
% in function file
function F=a4q8(x)
F(1) = x(1).*x(2)-4;
F(2) = x(1).*x(3)-3;
F(3) = x(2).*x(1)+x(3)-5;
end
% in command window
returnval=@a4q8;
r=[1,2,3];
abc=fsolve(returnval, r)
```

(Matlab Response)

abc =

3.0000 1.3333 1.0000

…….

Q9 Consider the spiral function generated by the parametric equations

$$x(t)=t\cos(2t)$$
$$y(t)=t\sin(2t)$$

for t>0. Also there is the parabolic function of

$$y=2+x^2$$

The objective is to determine the points of intersection of the spiral and parabola in the x-y plane that have a magnitude of less than 10. Use whatever numerical means to find these and list them out. List your program with outputs and sufficient comments that the TA can understand what you are doing.

(Matlab input)

```
%creating a function with x and t
f= @(t) t.*sin(2*t) - 2 - (t.*cos(2*t)).^2;
```

```
% i will use ez plot to visualize the approximate location of roots
ezplot(f)
grid on

% roots seem to be at approx 7.5, 7, 4.2, 3.9, so i will start there
root(2) = fzero(f,4);
root(3) = fzero(f,7);
root(4) = fzero(f,7.25);
for i=1:1:4
    x(i) = root(i)*cos(2*root(i));
    y(i) = root(i)*sin(2*root(i));
end
% this loop allows me to solve for the roots
```

(Matlab Response)

```
x =

   1.2436   -1.3648    2.1383   -2.2086

y =

   3.5466    3.8628    6.5722    6.8780
```