APM 153 Assignment Four – Solving for the Roots of a Cubic Equation

root3.m

(1) This week in lab we will be writing a **function** in Matlab named **root3.m.** Every function begins with a line of code that includes the word "function", the name of the function, and a list of **input arguments**.

```
function  root3(coef, xn)
```

(2) In the case of our root3.m function, the input arguments consist of a vector of the coefficients (coef) and an initial estimate of the root of the cubic equation (xn).

(3) After the function statement, the next lines of code should be the **help file** starting in the H1 Comment Line which was **previously discussed in class**. (Hint: DO IT).

(4) Next, we assign values from the input variable coef to the variables a, b, c, and d. The line of code below assigns the first **element** in the vector coef to the variable **a**. How would you assign the other elements of coef to b, c, and d?

```
a = coef(1)
```

Calculate the Discriminant

(5) Once you have assigned the elements from the vector coef to the variables a,b,c, and d, you are ready to calculate the discriminant as discussed in class. Use the value of the discriminant to write out a message telling the user what type of roots to expect.

Perform the Newton-Raphson Equation

(6) Write a line of code that will calculate the new, improved estimate, **xnplus1**, using the Newton-Raphson equation. Once you have calculated value of xnplus1 for the first time you can now set up your do-while loop to repeat the calculation.

(7) A do-while loop begins with word **while** and ends with the word **end**.

```
while   (some condition)
        do this
        then do this
end
```

(8) For our do-while loop, we want the do-while to continue as long as the absolute value of (xnplus1 – xn) is greater than 0.000001. (What Matlab command yields the absolute value?)

(9) Finally, when your function has calculated the root nearest to the estimate xn, have your function display a message and the value of xnplus1 as in the lines of pseudocode,..

```
display  "The root nearest to the estimate x is"
display  xnplus1
```

Testing root3

(10) Test your function for the following cubic equations. Find all three roots.

$-2000x^3 - 54x^2 + 100x - 2 = 0$	three real roots, all unequal
$8x^3 + 6x^2 + 4x + 2 = 0$	one real and two complex roots
$x^3 – 7x^2 + 16x - 12 = 0$	three real roots, two equal (3, 2, 2)

Check Your Results

(11) Matlab has a built-in root finder named **roots.m** You can check your results by typing in **roots(c)**, where c is the vector of coefficients. You will notice **roots.m** finds **all three roots** at once and does not need an estimate

What to Hand In

(12) Assignment Three will be **due in lab on Friday, February 17th**. You will hand in a printout of your MS Word document with,...

Part One	Your **algorithm** as pseudocode and as a flowchart
Part Two	A copy of your root3.m **mfile**.
Part Three	Your **diary** file showing that your root3.m found **all three roots** for each of the cubic equations **and** that you **checked your results** with roots.m
Part Four	**Answer the questions**: (1)How would you change your root3.m to solve for the roots of a fourth-order polynomial using the Newton-Raphson method? and (2) How many roots would a fourth-order polynomial have?