APM 153 LECTURE EIGHT – Numerical Methods, Newton-Raphson Method, Do-While Loops, Discriminant of a Cubic Equation, Algorithm for root3.m

Numerical Methods

(1) We have already discussed the fact that many solutions to problems in mathematics are not exact, but are instead an **approximation**.

(2) One reason we often rely on approximations is that **no analytical solution exists** for the problem being solved. Where an exact, analytical solution does not exist or is too difficult, we use a **numerical method** to approximate the solution.

(3) Numerical methods typically use **repetition** to solve a problem. For example, when dividing a small number into a very large number we continue dividing the remainder until we reach an acceptable solution.

Newton-Raphson Method

(4) For a more sophisticated example of a numerical method, we can use the Newton-Raphson method for computing the roots of a polynomial equation such as,…

$$f(x) = ax^3 + bx^2 + cx + d$$

where the following equation is repeated over and over until an acceptable level of precision is achieved.

$$x_{n+1} = x_n - \left[\frac{f(xn)}{f'(xn)} \right]$$

(5) In this formula, we start with an estimate, (x_n), for one of the roots of the polynomial equation and we calculate an "improved estimate" (x_{n+1}), that is closer to the true value of the root.

(6) In other words, we enter a **guess** (x_n), for what the true value of the root is, we pass our guesstimate through the equation, and we come up with an answer that is closer to the actual value of the root.

(7) Also, in this formula $f(x_n)$ represents the function at x_n and $f'(x_n)$ represents the first derivative of f.

(8) Given our original equation above from (1), what would be the first derivative of *f*?

$f(x) = ax^3 + bx^2 + cx + d$ $f'(x) =$ ____

(9) After passing our initial estimate through the equation, we repeat the process with the new estimate (x_{n+1}) as our starting point.

(10) So, the first time through the equation, we use our initial guess (x_n) in the equation and we get the result (x_{n+1}). Next, we take x_{n+1} and pass it through the equation to yield x_{n+2} and so on.

(11) We repeat the formula until an acceptable level of precision is achieved. But, how do we know when an "acceptable" level of precision has been achieved?

(12) Well, we set the precision ourselves and **we test our results**. Let's say I would like to know the value of the root to within 0.000001 of the actual value.

(13) Every time I pass a new estimate through the formula, we compare the value of x_{n+1} with x_n.

(14) As long as the difference between the two estimates is greater than 0.000001, we keep on repeating the formula.

Do-While Loops

(15) To repeat the formula in Matlab, we need to use what is called a "do-while" loop.

(16) A do-while loop is a command that tells the program to repeat a process as long as some condition is true.

(17) In pseudocode, we might write the following.

```
while abs(xnplus1  - xn) >  0.000001
        xn = xnplus1
        xnplus1 = xn – f(xn)/f'(xn)
end
```

(18) The lines of pseudocode above say,

> *"As long as the difference between xn+1 and xn is greater than 0.000001, continue to replace the old estimate xn with the improved estimate and pass the new estimate through the Newton-Raphson equation."*

(19) When the difference between the current values of xn and xnplus1 becomes less than 0.000001, it means that the value of f(xn) **must be very close to 0**.

(20) The value of xn where *f(xn)* = 0 is said to be the **root** of the equation.

(21) A cubic equation of the form $f(x) = ax^3 + bx^2 + cx + d$ will always have three roots. As in the program calc_roots.m, it is possible to calculate whether the roots are real or complex using the **discriminant**.

Discriminant of a Cubic Equation

(22) Given a cubic equation of the form $ax^3 + bx^2 + cx + d = 0$, you can calculate the discriminant as follows.

First calculate $Q = \dfrac{3c - b^2}{9}$ and $R = \dfrac{9cb - 27d - 2b^3}{54}$

Next calculate discriminant = $Q^3 + R^2$

(23) If discriminant > 0 — there is one real root and two complex roots
discriminant = 0 — all roots are real and at least two are equal
discriminant < 0 — all roots are real and unequal

Algorithm for root3.m

(24) On Friday in lab we will begin working on a Matlab function called root3.m which will use the Newton-Raphson method to solve for the roots of a cubic equation.

(25) Before we start writing code however, we need to at least begin our algorithm. Based upon your experience with calc_roots.m, what do you think should be the first thing we do in the algorithm?

Pseudocode Algorithm for root3.m