The first of teaching aids for Calculus with the HP-49G and the TI-89. This topic will cover the Newton-Raphson recursive algorithm, commonly known as Newton's Method. Given the fact that we alredy know how to calculate derivatives and find the equations of tangent lines to curves at a given point, we will move on. To set the stage for a problem, suppose a used car dealer offers to sell you a car for $18,000 or for payments of $375 per month for five years. You would like to know what monthly interest rate the dealer is, in effect, charging you. To find the answer, you have to solve the equation (for our time here we will not discuss how we come up with the equation. Accept it right now for no better reason than authority):

48x(1+x)^60 - (1+x)^60 + 1 = 0

How would you approach solving it? (Note for a quadratic ax^2 + bx +c = 0, there is a well known formula to find the roots called the quadratic equation, 3rd and 4th degree equations get much more complicated and there are formulas to find the roots there, but if f(x) is a polynomial of degree 5 or higher, there is no such formula to find exact roots.) We could graph the function in our HP-49 or TI-89, set the viewing rectangle to x->{0,.012} y->{-.05,.15}, and then use the trace function to approximate the root between .007 and .008. Zooming in repeatedly, we could find correct to nine decimal places that the root is .007628603. But this is tiresome, redundant, and takes a great deal of time. We could use the Solve( ) command in our calculators to find the approx solution as well. But how does the calulator find the root? They use a variety of methods, but the most commonly used method is Newton's Method. Now what is Newton's Method? Lets discuss it detail:

Suppose you have a curve that has a root R and suppose R is not known. To find R, we take a known value close to R and call it x1. Then we locate the y-value on the curve so that we have a point (x1, f(x1)). Then calculate the tangent line at that point and sketch it such that the tangent line crosses the x axis. That root where the tangent line crossed we will call x2. Then find the y-value of x2 and repeat the process. What you will find is in effect x2, x3, etc will get closer and closer to your R root (there are cases where this will fail, we will discuss these later). In fact, you only need to find about x5 or x6 to be correct to 6-8 decimal places! To find a formula for x2 in terms of x1, we use the fact that the slope of L is f'(x), so its equation is:

y - f(x1) = f'(x1)*(x-x1); where f'(x1) is the derivative of f(x1).

Since the x-intercept of L is x2, we set y=0 and obtain;

0 - f(x1) = f'(x1)*(x2-x1).

If f'(x1) != (does not equal) 0, we can solve this equation for x2:

x2 = x1 - f(x1) / f'(x1)

We use x2 as a second approximation to R:

x3 = x2 - f(x2) / f'(x2)

If we keep repeating this process, we obtain a sequence of approximations x1,x2,x3,x4,....... In general, if the nth approximation is xn and f'(xn) != 0, then the next approximation is given by:

xn+1 = xn - f(xn) / f'(xn).

If the numbers xn become closer and closer to R as n becomes large, then we say that the sequence converges to R and we write:

lim xn = R
n->oo

Although the sequence fo successive approximations converges to the desired root for some functions, in other circumstances the sequence may not converge. This is likely to be the case when f'(x1) is close to 0. It might even happen that an approximation falls outside the domain of f. THEN NEWTONS METHOD FAILS AND A BETTER INITIAL APPROXIMATION X1 SHOULD BE CHOSEN.

Now how would you use this with the HP-49 or the TI-89? Well, each calculator goes about it differently but the idea is the same. Suppose we want to find the root of x^3 + x + 1 = 0. Let's make our first guess (x1) be -1. f'(x) = 3x^2 + 1, so our equation would be:

(-1) - ((-1)^3 + (-1) + 1) / (3(-1)^2 + 1).

Which yields -.75. Putting .75 in for x2 and re-evaluating gives us-.686046511628. Put our answer now in for x3 and evaluate again, and we get -.682339582597. One more time yields -.682327803947, and a last evaluation gives us -.682327803828. (Note: Notice how on our 2nd and 3rd evaluations .68 is repeated, and on our 3rd and 4th .6823 is repeated and on our 4th and 5th .682327803 is repeated? Newton brought to light something interesting when coming up with his recursive formula. For each evaluation after x2, your accuracy will double. Notice we have two decimal places of accuracy by our 3rd evaluation, 4 by our fourth, and 9 by our fifth. Newton's Method is a GREAT way to get accurate in a hurry. Chances are by our 6th evaluation, we would be accurate to 18 decimal places!)

Now those of you with the HP-49, you can program it so each time you hit enter, your answer will be displayed. To program the algorithm, do the follwing (assuming you are in RPN mode): Place your first equation (f(x)) on the stack. Press alpha Y, then STO. Now put the derivative of your first equation (f'(x)) on the stack. Press alpha Z then STO. Press your first initial guess on the stack (in our previous example, it would be -1), press alpha X then STO. Now for the program. Key in the following, then press enter to place it on the stack:

```
<< X Y Z / - ->NUM { X } PURGE X STO X >>
```

Now that it is on the stack, press alpha twice, type NEWT, then STO. If you press the VAR button you will notice your variables X, Y, Z, and NEWT above their respective soft keys. Now every time you press NEWT, you will get a numerical value closer and closer to your root. On the TI-89, the idea is the same. Press the green diamond then f1. This brings you to the y= screen. Type your first equation in y1, and the derivative of your second equation in y2. Then press the home button. For your program here, press your first inital guess (ours was -1) the press STO-> , press X, then ENTER. Here you assigned a value to x, Then in this sequence, press the following:

```
X - Y1(X) / Y2(X) STO-> X ENTER
```

Each time you press ENTER, you will get closer and closer to the root you are seeking.

I know this was a lengthy post, so your comments are appreciated. Any questions, feel free to email me, or respond to the post. Thanks!
-Aaron