Return to List of Lessons

Calculator Lesson 11

Newton's Method

We will try to solve the same problem we solved using the bisection method in Lesson 7, namely, find $\sqrt[3]{5}$, but this time we will use Newton's Method. As before, we will be seeking a zero of the function F(X) = X^3 - 5, so we define this function as we did before. We also need the function F1(X) = 3X^2 for the derivative of F. This function is quite simple and the easiest thing would be to simply type it in, but let us see how we could take the derivative and turn it into a function if F were much more complicated. Be sure the calculator is in Exact mode, put 'F(X)' on the stack and press LS CALC F5-DERVX to find the derivative. Before we can use the DEF command to define this as a function, we need to put F1(X)= in front of it. Press DA F1-EDIT RA to get this into the command line with the cursor inside the apostrophe, then type F1(X)= ENTER ENTER. We can now press LS DEF to define this as a function. Press VAR and we should now have F1 as the first item in our menu and F as the second.

Recall that Newton's Method requires us to make an initial estimate, x_0, then to apply the recursive formula $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ until the desired level of accuracy is achieved.

We know the zero we are seeking is between 1 and 2. When we have our zero trapped in such an interval, it is customary to use the middle of the interval as the starting point for Newton's Method, so put 1.5 on the stack. The recursion formula will need to use this value three times, and we would also like to keep a list of our successive approximations on the stack for comparison purposes. Thus, we want four copies of this on the stack so our algorithm will be

ENTER ENTER ENTER F2-F RA F1-F1 ÷ -

The RA in the above sequence serves as a SWAP command to interchange the values in levels 1 and 2 of the stack, putting X on level 1 so we can compute F1(X). Note that F and F1 are now in the right position for the division. We now see 1.74074074074 on level 1 of the stack and our first approximation, 1.5 is on level 2. Go through the algorithm again and our next approximation is 1.71051646184. Repeat the above algorithm three more times and we see that the last two approximations are both 1.70997594668, which is the correct value of the cube root of 5 to twelve decimal places. Notice the tremendous difference in the speed of Newton's Method compared to the Bisection Method. This one gave us an eleven decimal answer in 5 iterations, while the Bisection Method required 8 iterations just to get a 2 decimal answer. Unfortunately, as pointed out by our text, there are situations were Newton's Method can go very wrong, while the Bisection Method is always sure to converge, at least within the computational limitations of the computing devise we are using.

In our numerical analysis course we prove a theorem that under the right conditions Newton's Method is also sure to converge if our initial estimate is close enough to the actual root. Unfortunately, these issues are beyond the scope of an introductory calculus course.

To get a better appreciation for the power of functions, try defining the following function:

$$NM(X) = X - \frac{F(X)}{F1(X)}.$$

Now, after entering the initial approximation, 1.5, onto the stack, the algorithm becomes:

ENTER F1-NM

repeated five times.

We can even improve on this if we are willing to engage in a little be of programming. Press the following sequence of key strokes:

‘ F1-NM ENETER LS DA RA LS PRG F1-STACK F1-DUP ENTER.

Now all that is needed it to put the initial approximation on the stack and press F1-NM repeatedly until the same value shows up on the stack twice in a row.

If you really want to get fancy, click on the introduction to programming link at the bottom of the list of lessons page and learn about WHILE LOOPS. You will then be able to write a program that will allow you to press only one key after entering the initial approximation. The program will automatically repeat the algorithm as many time as needed until the desired degree of accuracy is achieved. That is just a small example of the power of programming.

Return to List of Lessons