From previous page

```
(DOTGRAPH :FUNC :L + :D :H :D)
END
```

Now that there was more than one actual graphing procedure, it was necessary to enable a program to receive inputs consisting of a graphing procedure (either GRAPH or DOTGRAPH), a function, an interval on which to graph the function, and the optional increment. (If nothing is inputted for this, the computer will automatically use the .2 that was mentioned earlier.) The program was named **M&M&M** because it took **M&M** one step further.

On occasion, when more than one graph was sketched on the screen at the same time, they intersected. The coordinates of the intersection are often interesting, so a program to find them became necessary. The result was a program called **COLL** (collect coordinates) which enables the user to see a display of such coordinates when he/she moved the cursor arrow to the desired location and clicked the button on the mouse.

Polynomial functions were no longer a problem to graph. All that was necessary for them to be graphed on the screen was that they be defined in Logo (a process that only takes a few seconds.)

To aid in our calculus assignments, we decided to write a program that would determine the derivative of all functions that we had already defined.

Definition of a derivative: Suppose F is a function defined on a closed interval $[a, b]$, and $x_o \in [a,b]$. Then the derivative of F evaluated at x_o is equal to

$$\lim_{h \to 0} \frac{F(x_o + h) - F(x_o)}{h}$$

provided the limit exists.

The fraction in this equation is called the differential quotient. We wrote such a program that approximated the limit of the differential quotient as h went to 0 (by letting $h = .01$), and we incorporated that into another graphing program known as GRAPHD, for *graph derivative.* This now enabled us to not only graph the function that we wanted to see, but also its derivative on the same screen (an invaluable aid in a calculus course.) Here are **DERIV** and **GRAPHD:**

```
TO DERIV :F :X
OP ((APPLY :F ( :X + .01)) - (APPLY :F :X )) / .01
END
```

```
TO GRAPHD :F :L :H [ :D .2]
IF :L > :H [PD STOP]
PD HT
SETPOS SE (250/ :SCALE) * :L —>
(250/ :SCALE) * (RUN (SE "DERIV QUOTE :F :L))
(GRAPHD :F :L + :D :H :D)
END
```

The graphing of trigonometric functions was the next project we undertook. The computer already has the basic functions *sin, cos,* and *tan*. These formulas, however, were designed to take inputs measured in degrees instead of radians. After this was completed, the functions for *sec, csc* and *cot* were added along with all six inverses (one for each function.) Since the derivative function was already defined, we had the graphs of all six trigonometric functions, and their inverses at our disposal.

The only problem with the derivative that we defined was that it only approximated the actual derivative. Therefore, we also input the actual derivatives (modeled as Logo functions.) This enabled us to see how close the approximation is to the exact function. The approximations were very close indeed.

The nest program we designed enabled us to graph a function F over a closed interval and sketch a trapezoid at specified increments. The program, named **GRAPHTRAP**, is as follows:

```
TO GRAPHTRAP :FUNC :L :H :D
IF :L > :H [PD STOP]
PD HT
SETPOS SE (250/ :SCALE) * :L (250/ :SCALE) * (APPLY
:FUNC :L)
SETPOS SE (250/ :SCALE) * :L 0
SETPOS SE (250/ :SCALE) * :L (250/ :SCALE) * (APPLY
:FUNC :L)
(GRAPHTRAP :FUNC :L + :D :H :D)
END
```

Our latest addition to this project included a program which approximates the area under a curve using the trapezoid rule from Calculus. Recall that the trapezoid rule is defined to take a function F on a closed interval $[a, b]$ and it approximates $\int F(x)\,dx$. It does this by performing the following steps:

1. It divides [a,b] into n parts giving equally spaced points $x_o, x_1, ..., x_n$ where x_o equals a and x_n equals b
2. It adds the $F(x_i$'s)
3. It subtracts $.5((F(a) + F(b))$ from the sum
4. It multiplies this result by $(b-a)/n$

In mathematics, this algorithm can be stated as:

*See **Calculus**...page 14*