

A Brief Guide to Technology Use in Calculus

Lynn E. Garner

2002

1 General Use of Technology

Whether it is a computer laboratory or a handheld calculator, technology can be a great boon to the calculus student. Unfortunately, if not used wisely, technology can be a bane that wastes time and even prevents learning. Hopefully, this little guide will prevent unwise use and aid you in learning calculus.

1.1 When to Use Technology

There are certain settings when technology can be an appropriate aid. You already know a calculator can help with arithmetic, reducing drugery and speeding up the process. Similarly, a graphing utility can show you the graph of an unfamiliar function very quickly.

However, if you have to use your calculator to do the arithmetic you memorized in grade school, you have become too dependent on the technology. Similarly, if you have to get your calculator to show you what the graph of $y = 2x - 1$ or $y = x^2$ looks like, you are not thinking but using the technology as a crutch.

That is, as you gain experience, you should be able to get along without the technology more and more for the familiar mathematics. The technology remains just as useful for the drugery and for unfamiliar functions, but as you gain more familiarity in general you need the technology less to remind you of what you know.

There are certain settings in which technology is indispensable. Here is a short list; we will illustrate all of these settings in calculus class this semester.

1.1.1 Intensive computation

Whenever there are long and tedious calculations, technology removes the drugery. In fact, we simply would not do the calculations under normal circumstances without the technology to aid us.

1.1.2 Visualization

The technology is adept at showing us what we want to see, so long as we know how to ask it for what we want to see. The graph of an unfamiliar function, or of the graphical relationship between two or more functions, we can get with little trouble.

Remember that in looking at the graph of a function, the technology shows us a “window” through which we look at the graph. We must take care to choose the window appropriately, or we will not see anything helpful. All graphing utilities allow the user to choose a wide variety of variables that determine the window; experience will teach you how to select variables wisely.

1.1.3 Estimation Techniques

Technology is a necessary tool for estimating some things that cannot be found otherwise. For example, it is impossible to solve the equation $\cos x = x^2$ algebraically, but there are “numerical methods” that allow us to quickly find an approximate solution to almost any degree of accuracy. Here the technology is a powerful tool indeed, and should be used without apology.

1.1.4 Time Compression

Similar to removing drugery, technology can compress the time that it takes to get results, thereby enabling us to see relationships that we would otherwise miss. Like a time-lapse film, the technology lets us see things that we cannot see in other settings.

1.2 When Not to Use Technology

As already alluded to above, some things are simple enough that you should not have to use technology to get them. If you find yourself reaching for the calculator to do one-digit arithmetic or to graph linear functions, you most likely are not being efficient. Thinking and memory will do better for you in these simple cases.

There are other cases of misuse of technology. Here are a few examples.

1.2.1 The Black Box Syndrome

Whenever a person tries to get the technology to do something that the person does not in principle understand, he is guilty of the black box syndrome. If you do not understand at least in general what the machine is doing with your input to get its output, you have no basis for judging whether the output is reasonable. It is too easy to make little mistakes to trust the output of any machine uncritically, so it is unsafe to turn over to the machine what you don’t understand well enough to judge.

1.2.2 Programming the User

Training a person to push the right buttons on a machine comes close to the black box syndrome, for it emphasizes the machine and not the mathematics, thereby circumventing understanding. When the technology changes or is not available, the person is helpless. A more appropriate aim is to get the person to understand the algorithm and program his calculator to do it, for then he must understand the process thoroughly and communicate it precisely to a dumb machine.

1.2.3 Believing the Output

Not all that we ask the machine to do can it do, for it is a finite machine and many mathematical processes are infinite. As an almost trivial example, if you ask your calculator to divide 1 by 3, it will give you either the indicated division, $1/3$, thus not dividing at all, or it will give you a finite decimal approximation like $.333333333333$, which is not exact. As a more tricky example, have your graphing calculator plot the graph of $y = \sin 100x$ in the window $[-6, 6] \times [-2, 2]$. You will not get a nice sine curve unless your calculator has about 1000 times the usual resolution. It is like trying to spell out your full name on the sidewalk with ten bricks; it can't be done.

Thus, the output of a technological device cannot always be trusted. The extent to which it can be trusted depends on what you are trying to get it to do, your understanding of the limitations of the machine, and your understanding of how it does what you ask. If a cycle of the graph is shorter than the width of a pixel, you will not get a very good graph.

Similar cautions apply in other settings where it is much more difficult to tell whether the machine is having trouble bumping up against its limitations. In other words, depend on the machine only for what you know it can do.

2 Using Graphing Calculators

In many ways, a programmable graphing calculator is the ideal technology to assist with a calculus course. As you learn the concepts of calculus, the features of the calculator become available to you and can gradually take over the computational part of the calculus. You can program the calculator to carry out processes made possible by the calculus.

2.1 Graphing Functions

One part of the calculator is dedicated to getting the graph of a function for you. Learn how to make this process quick and efficient. Learn how to select an appropriate window and how to zoom in or out on a graph in a desirable way. Learn how to plot several graphs in the same window so as to compare them. Study your user's guide to become proficient.

2.2 Newton's Method

Newton's method, also known as the Newton-Rapheson method, uses machine iteration to solve an equation of the form $f(x) = 0$. The method iterates the formula

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}.$$

Because of the different ways in which calculators do things, writing a program on a graphing calculator is done differently for different machines.

Note that most graphing calculators have something like a “Solve” command that will attempt to solve an equation for a given variable in the neighborhood of a given value. This command typically does not use Newton’s method, but more robust variations of it, often in combination with other methods. In order to get Newton’s method, you have to program it.

2.2.1 Newton’s Method on Algebraic Machines

The simplest way to visualize the program is to have $f(x)$ and $f'(x)$ entered as formulas, say as $y1$ and $y2$, put an initial guess x_0 in a variable A , put A into x , and then write the iteration formula as

$$A - y1/y2 \rightarrow A.$$

Each application of the program gives the next iterate in A .

2.2.2 Newton’s Method on RPN Machines

On an RPN machine, such as an HP48G series, you can get the machine to compute the derivative $f'(x)$ after you have entered $f(x)$. We first enter $f(x)$ as a “push-button” function by putting ‘F(X)=...’ on the stack (filling in the formula in terms of X), and pressing the DEF key. You then see a button named F on the VAR menu; by entering any number x on the stack, you can compute $f(x)$ by simply pressing the softkey F.

Once f has been defined as a push-button function, the program

$$\lll \text{DUP} \rightarrow \text{X 'X-F(X)/}\partial\text{X(F(X))}' \rightarrow \text{NUM} \ggg$$

will take a guess x_i from the stack, compute x_{i+1} , and put both x_i and x_{i+1} on the stack for inspection. Each application of the program gives the next iterate.

2.3 Numerical Integration

The numerical integration estimates $LHS(n)$, $RHS(n)$, $TRAP(n)$, $MPS(n)$, and $SIMP(2n)$ typically do not appear as pushbuttons on a calculator. In order to get them, we must program the calculator. As you can imagine, this is done quite differently on different calculators. The results should all be the same, however. For test purposes, with $\int_0^3 x^3 dx$ you should get

$$\begin{aligned} LHS(100) &= 19.847025, \\ RHS(100) &= 20.657025, \\ TRAP(100) &= 20.252025, \\ MPS(100) &= 20.2489875, \\ SIMP(200) &= 20.25. \end{aligned}$$

2.3.1 Numerical Integration on TI Calculators

Here is a program for the TI85 that computes all five of the estimates above. With small syntax changes, it should be adaptable to the later TI machines, also. Run the program after entering the integrand into $y1$.

```
Program NUMINT
:Disp "LOWER LIMIT"
:Input A
:Disp "UPPER LIMIT"
:Input B
:Lbl P2
:Disp "N"
:Input N
:(B-A)/(2N)→H
:0→L
:0→M
:1→J
:A→x
:Lbl P1
:L+y1→L
:x+H→x
:M+y1→M
:x+H→x
:IS>(J,N)
:Goto P1
:Disp "L/R/T/M/S"
:A→x
:L-y1→R
:B→x
:R+y1→R
:2*H*L→L
:2*H*R→R
:2*H*M→M
:(L+R)/2→T
:(2*M+T)/3→S
:Disp L
:Disp R
:Disp T
:Disp M
:Disp S
:Menu(1,"AGAIN",P2,5,"QUIT",P3)
:Lbl P3
```

2.3.2 Numerical Integration on Casio Machines

This program was created for Casio fx series calculators, and is very similar to the above. Run the program after entering the integrand into f_1 .

```
Program NUMINT
“L-LIM”?→A
“U-LIM”?→B
“DIVNS”?→N
(B-A)÷(2N)→H
0→L
0→M
A→X
Lbl 1
f1+L→L
X+H→X
f1+M→M
X+H→X
Dsz N
Goto 1
“LEFT,RIGHT,TRAP”
2HL→L
L▲
L+2Hf1→T
A→X
T-2Hf1→T
T▲
(L+T)÷2→T
T▲
“MID,SIMP”
2HM→M
M▲
(2M+T)÷3→M
M▲
```

2.3.3 Numerical Integration on the HP48G Series

In contrast to the algebraic machines, the HP48G series calculators best handle numerical integration as a collection of programs. We will organize them into a subdirectory on the VAR menu. In what follows, we have you create the environment for numerical integration by following a series of steps.

Enter the HOME level of the VAR menu by pressing HOME and then VAR if necessary. Create a subdirectory NUMI by typing 'NUMI' ENTER CRDIR ENTER. Then enter the subdirectory by pressing NUMI on the VAR menu.

Create a container H by typing 1 ENTER 'H' STO.

Create a container N by typing 1 ENTER 'N' STO.

Create a container B by typing 1 ENTER 'B' STO .

Create a container A by typing 1 ENTER 'A' STO .

Create a container F by typing 1 ENTER 'F' STO .

Create the program SUM by typing

$\ll \rightarrow X \Sigma(I=0, N-1, F(X+I*H))*H \gg$ ENTER 'SUM' STO .

Create the program SIMP by typing

$\ll MPS 2 * TRAP + 3 / \gg$ ENTER 'SIMP' STO .

Create the program MPS by typing

$\ll A H 2 / + SUM \gg$ ENTER 'MPS' STO .

Create the program TRAP by typing

$\ll LHS RHS + 2 / \gg$ ENTER 'TRAP' STO .

Create the program RHS by typing

$\ll A H + SUM \gg$ ENTER 'RHS' STO .

Create the program LHS by typing

$\ll A SUM \gg$ ENTER 'LHS' STO .

Create the program NSTO by typing

$\ll 'N' STO B A - N / 'H' STO \gg$ ENTER 'NSTO' STO .

Create the program ABSTO by typing

$\ll 'B' STO 'A' STO \gg$ ENTER 'ABSTO' STO .

Create the program FSTO by typing

$\ll 'F(X)' SWAP = DEFINE \gg$ ENTER 'FSTO' STO .

To estimate the integral $\int_a^b f(x) dx$ with n subintervals, first type in the expression for $f(x)$ in tic marks, ' ', with X as the variable. Press ENTER and then press FSTO on the VAR menu. Then type in a SPC b and press ABSTO on the VAR menu. Complete the entry of data by typing n and pressing NSTO on the VAR menu. Finally press LHS, RHS, TRAP, MPS, or SIMP, according to what you desire.

When you are through using the numerical integration environment, exit the sub-directory NUMI by pressing UP. This keeps the environment of NUMI from being corrupted by other containers, etc.

3 Using Computer Mathematics Applications

A number of high-powered computer applications do mathematics. The most popular ones are Maple, Mathematica, and MatLab. These applications are present on the machines in the Math Department computer lab in 150 TMCB. Each of these applications comes with handbooks and on-line helps that describe how to get the application to do what you want.

3.1 Graphing Functions

These applications graph functions (even three-dimensional graphs) with great alacrity. In order to see exactly what you want, you may have to learn quite a bit about the program in order to select the window and specify other parameters, etc.

For example, in order to plot the curve $y = \frac{x}{2} \cos x$ in Maple in the window $[-7, 7] \times [-3, 3]$, use a command like

```
plot((x/2)*cos x,x=-7..7,-3..3);
```

In order to plot two curves in Maple, say $y = 1.05^x$ and $y = x^2$ in the window $[0, 250] \times [0, 100000]$, use a command like

```
plot([(1.05)^ x,x^ 2],x=0..250,0..100000);
```

Whatever application you are using, you will have to study its syntax in order to request the picture you want.

3.2 Newton's Method

Mathematics applications all have built into them the means for solving equations. For example, in Maple the call

```
fsolve(f(x),x);
```

will give numerical solutions to the equation $f(x) = 0$. The method used is not just Newton's method; there seems to be no Newton's method procedure built into Maple, at least not by that name.

Here is a procedure that accomplishes Newton's method in Maple:

```
f:=cos(x)-x; fp:=value(diff(f,x)); z:=x-f/fp;
initialguess:=1; iterates:=6;
x:=initialguess; for i from 1 to iterates do x:=evalf(z); od; readlib(unassign); unassign('x');
```

As written, this procedure finds 6 iterates of a zero of $f(x) = \cos x - x$, starting with 1 as the initial guess. You may put in a different function f , a different initial guess, and a different number of iterates to solve the problem you face.

3.3 Numerical Integration

Mathematical applications have procedures for evaluating definite integrals in a variety of ways. In Maple, the environment called by

```
with(student);
```

has the procedural calls `leftsum`, `rightsum`, `trapezoid`, `middlesum`, and `simpson`. Each of these has the format

```
evalf(leftsum(f(x),x=a..b,n));
```

for estimating the integral $\int_a^b f(x) dx$ using n subintervals.

In addition, this environment has calls `leftbox`, `rightbox`, and `middlebox` that create graphical depictions of rectangles approximating the area under the graph. Each of these again has the format

```
leftbox(f(x),x=a..b,n);
```