An Introduction to Maple (Calculus II style)

Maple has two main modes: worksheet mode and document mode. In document mode, you can work like you would in Microsoft Word (editing text, inserting pictures, etc.) but you can also insert equations and have Maple compute for you. Personally, I don't like using document mode because it is difficult to distinguish between text and maple commands. Everything gets mixed up.

This document is a "worksheet". In worksheet mode, you have text boxes and prompts.

This is written in a text box. The prompts: "[>" are Maple command prompts. You can put Maple commands there. When you hit enter, Maple will execute them.

When typing (at a prompt or in a textbox) Maple has two modes: "Text" and "Math". When writing in text mode, you will see exactly what you're typing. At prompts this means you'll see red text (your commands). In math mode, Maple will "pre-format" equations doing things like displaying fractions and putting exponents and subscripts where they belong. I prefer to write in text mode because I like to see what I've typed. Math mode may look prettier, but if something goes wrong it can be very difficult to track down the problem. To switch between modes you can either hit the "Text" and "Math" buttons above (top left of the second toolbar) or you can use the "F5" key. Maple defaults to text mode for textboxes and math mode for command prompts. To change this go to "Tools -> Options" menus, select the "Display" tab, and change "Input Display" from "2D math" to "Maple notation".

The following commands clear memory and then load up the "plots" and "Student[Calculus1]" packages.

Many commands are preloaded, but not all. Since we'll want to check out some advanced plot tools and see how to integrate, I've loaded these extra packages.

```
> restart;
  with(plots):
  with(Student[Calculus1]):
```

When entering Maple commands "shift-enter" gives you an extra line to type on. "enter" executes what you've already typed.

Notice that each line ends with either a ":" or ";". Lines ending in a semi-colon are executed and then results/output is displayed. Lines ending in a colon are executes with output suppressed.

Suppose we wish to find out what is in the "plots" package. Of course, we could open Maple's help or go to Google (often a great choice -- many Maple tutorials exist out on the web). However, the easiest way to access help is to use "?"

```
> ? plots
```

What does "Student[Calculus1]" have to offer?

```
> ? Student[Calculus1]
```

Assignment:

Let's assign a variable a value. This is done with ":="

```
> a := 5;
```

$$a := 5 \tag{1}$$

Now "a" has the value 5. So anytime Maple sees "a" it will substitute "5".

Note:

Maple is case sensitive.

"a" and "A" are not the same.

```
> a^2;
```

$$25 \quad (2)$$

```
> A^2;
```

$$A^2 \quad (3)$$

If I reassign the value for "a" and re-execute commands, Maple will use whatever assigned I gave "a" last.

```
> a := 2;
```

$$a := 2 \quad (4)$$

So if I go back up to "a^2", Maple will now print out 4. Things brings up an important point...

Results depend on the order in which you execute commands.

So if you want to have a readable, sensible Maple worksheet, don't jump around! Work from top to bottom.

This also brings up another important point. If you close down Maple, **it forgets everything**!

So if you close down Maple and then open it back up again, you need to re-execute all commands before continuing your work.

Arithemetic:

+, -, *, /, ^ for add, subtract, multiply, divide, and exponeniate.

```
> 0+1-2*3/4^5;
```

$$\frac{509}{512} \quad (5)$$

Equations:

"=" is used to equations (not assignments). So writing "a=5" does not give "a" the value "5". Think of it more as asking, "Is the value of 'a' equal to 5?"

```
> a=10;
```

$$2 = 10 \quad (6)$$

Notice that the above equation is false. Our command just stated the equation. To evaluate its truth requires something more.

We want to use the "evalb" function. This is a member of a family of evaluation functions. "evalb" stands for evaluate to boolean (i.e. true or false). Other functions like this are: "eval" and "evalf". "eval" is just a generic evaluate function which prods Maple to do something. "evalf" is evaluate to floating point. This makes Maple give an approximation of an exact
symbolic answer.

```
> evalb(a=10);
```

$$\mathit{false} \tag{7}$$

```
> evalf(0+1-2*3/4^5);
```

$$0.9941406250 \tag{8}$$

Solving:

To solve an equation you can enter an equation and choose a "solve" function from the context panel to the right.

Note: If the context panel is not open, "right-click" on the blue output and choose "Open Context Panel for more..."

```
> x^2+3*x-6=0;
```

$$x^2 + 3\,x - 6 = 0 \tag{9}$$

```
> solve( { (9) } );
```

$$\left\{x = -\frac{3}{2} + \frac{\sqrt{33}}{2}\right\}, \left\{x = -\frac{3}{2} - \frac{\sqrt{33}}{2}\right\} \tag{10}$$

Solve comes in several flavors. "solve" is the generic solver and will try to find all possible **exact** (symbolic) solutions. "fsolve" is Maple's numeric solver. It tries to find all solutions **approximated** in terms of floating point numbers.

Be careful, Maple's solver is not perfect. If you ask Maple to solve something and it comes back blank, that may be because there is no solution. However, it may just be the case that Maple couldn't find the solution. In the same vein, "fsolve" sometimes fails to find a solution or just finds one solution when many solutions exist.

To help "fsolve" one can choose "Numerically Solve from Point" and give it a "guess" to help it get started close to the desired solution.

```
> solve(x^2+3*x-6=0);
```

$$-\frac{3}{2} + \frac{\sqrt{33}}{2}, -\frac{3}{2} - \frac{\sqrt{33}}{2} \tag{11}$$

```
> fsolve(x^2+3*x-6=0);
```

$$-4.372281323, 1.372281323 \tag{12}$$

If you give solver an expression, it will set it equal to zero and solve.

```
> solve(x^2+3*x-6);
```

$$-\frac{3}{2} + \frac{\sqrt{33}}{2}, -\frac{3}{2} - \frac{\sqrt{33}}{2} \tag{13}$$

Important Functions:

Many functions are exactly what you think they should be. Sine, cosine, tangent, the natural logarithm are:
sin(x), cos(x), tan(x), ln(x)

The one that gives many students trouble is the exponential function. It is <u>not</u> "e^x". Instead it is "**exp(x)**".

The reason "e^x" does not do what we want it to is that Maple sees "e" as a variable. It can be assigned any value. So to Maple "e^x" behave as we might like, we'd first need the assignment command "e := exp(1);"

Here's a more complicated expression. We will look for roots of this expression (places where the corresponding

graph crosses the x-axis). Notice that "solve" fails. "fsolve" picks out the solution "-3.07" and "fsolve" with point "x= 10" picks out a
solution near "10".

```
> solve(exp(-0.1*x+ln(x^2))*sin(x^2));
> fsolve(exp(-0.1*x+ln(x^2))*sin(x^2));
```

$$-3.069980124 \quad (14)$$

```
> fsolve(exp(-0.1*x+ln(x^2))*sin(x^2),x=10);
```

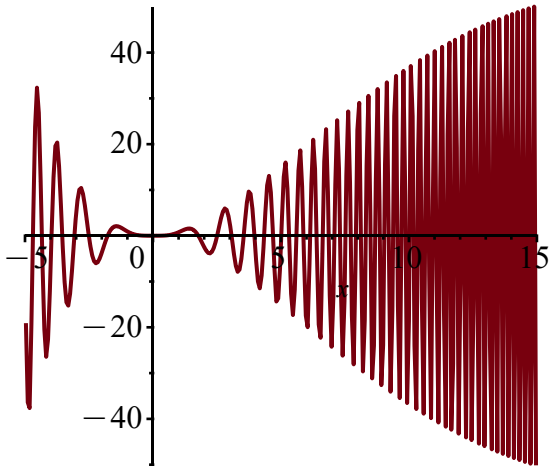
$$10.02651310 \quad (15)$$

Basic Plotting:

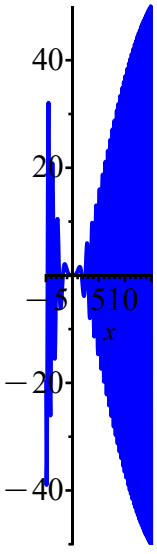
To plot an expression. Use the "plot" command. It syntax is as follows:
"plot(thing to plot, variable = left bound .. right bound);"

Plot will automatically scaling the vertical axis as necessary.

```
> plot(exp(-0.1*x+ln(x^2))*sin(x^2),x=-5..15);
```

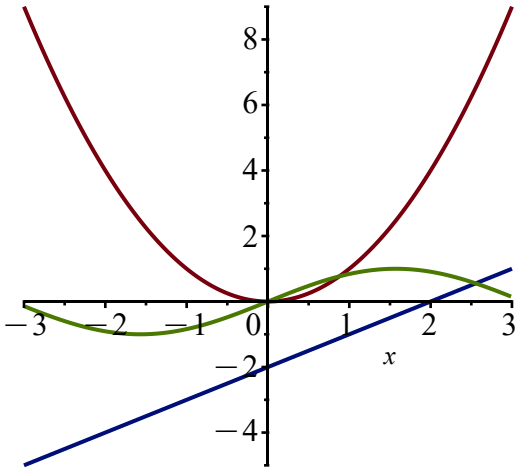



```
> ? plot options
> plot(exp(-0.1*x+ln(x^2))*sin(x^2),x=-5..15,scaling=constrained,color=blue);
```



We can plot several things at once by giving the plot command a set of functions to plot.

```
> plot({x^2,sin(x),x-2},x=-3..3);
```

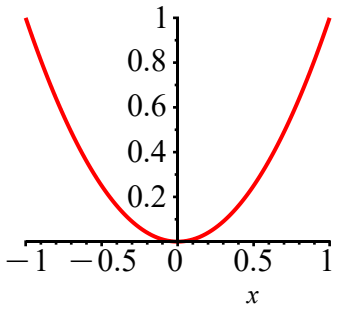


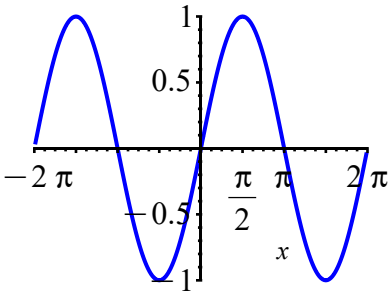
Alternatively you can use the "display" command to piece together various plots.

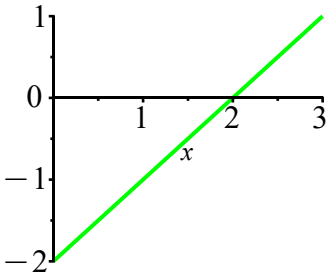
The "scaling=constrained" plot option tells Maple not to rescale coordinate axes.

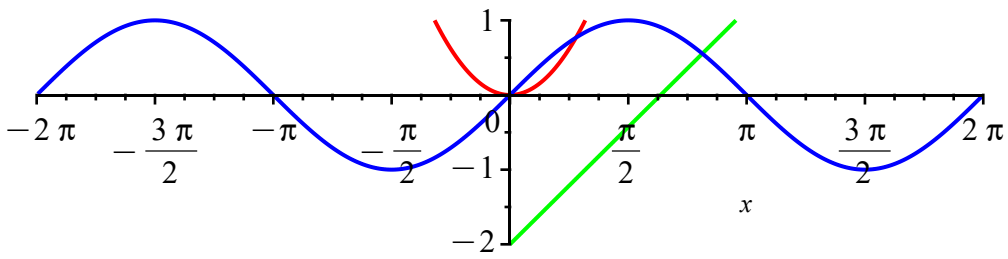
```
> firstPlot := plot(x^2,x=-1..1,color=red);
  secondPlot := plot(sin(x),x=-2*Pi..2*Pi,color=blue);
  thirdPlot := plot(x-2,x=0..3,color=green);

  display({firstPlot,secondPlot,thirdPlot},scaling=constrained);
```









Pi:

The mathematical constant pi is "Pi" not "pi"

"pi" (lowercase) is a variable which can be assigned any value.

```
> pi := 3;
```

$$\pi := 3 \quad (16)$$

```
> Pi <> pi;
```

$$\pi \neq 3 \quad (17)$$

```
> evalf(Pi,100);
```

3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706 (18)

Functions vs. Expressions:

Functions in Maple are structured as follows: "input thing -> what I map input thing to"

Here we can define the square function:

```
> f := x -> x^2;
```

$$f := x \mapsto x^2 \quad (19)$$

```
> f(6);
```

$$36 \quad (20)$$

Breaking that down "x^2" is the formula used to compute our function, "x -> x^2" says x should map to the expression "x^2", and then "f := x -> x^2" says our square function should be assigned to the name "f". In order to print out things nicely, I generally define functions as follows:

```
> f := x -> x^2:
  'f(x)' = f(x);
```

$$f(x) = x^2 \quad (21)$$

The first line "f := x - > x^2:" actually defines the function whereas the second line prints it out nicely.

We can evaluate functions as we might expect:

```
> f(-2);
  f(Bob);
```

$$4$$

$$Bob^2 \quad (22)$$

Sometimes it is more convenient to use expressions. However, these require the "subs" command to substitute in (i.e., plug in) values.

```
> g := x^2;
```

$$g := x^2 \quad (23)$$

```
> subs(x=2, g);
```

$$4 \quad (24)$$

Calculus:

differentiate with diff, integrate with int

```
> diff(sin(x),x);
```

$$\cos(x) \quad (25)$$

```
> diff(sin(x),x,x);
```

$$-\sin(x) \quad (26)$$

```
> diff(sin(x),x,x,x);
```

$$-\cos(x) \quad (27)$$

```
> diff(sin(x),x$4);
```

$$\sin(x) \quad (28)$$

```
> int(sin(x),x);
```

$$-\cos(x) \quad (29)$$

```
> int(sin(x),x=0..Pi);
```

$$2 \quad (30)$$

Capatalized commands are usually "inert".

```
> Int(sin(x),x=0..Pi) = int(sin(x),x=0..Pi);
```

$$\int_0^{\pi} \sin(x)\, dx = 2 \quad (31)$$

The "DiffTutor" command allows us to see how to differentiate step-by-step.

```
> DiffTutor(x^3*sin(5*x-2),x);
```

$$\frac{d}{dx}\left(x^3 \sin(5x-2)\right)$$

$$= \left(\frac{d}{dx}\left(x^3\right)\right)\sin(5x-2) + x^3\left(\frac{d}{dx}\sin(5x-2)\right) \quad [product]$$

$$= 3x^2\sin(5x-2) + x^3\left(\frac{d}{dx}\sin(5x-2)\right) \quad [power]$$

$$= 3x^2\sin(5x-2) + x^3\left(\left(\frac{d}{d\_X0}\sin(\_X0)\right)\Big|_{\_X0=5x-2}\right)\left(\frac{d}{dx}(5x-2)\right) \quad [chain]$$

$$= 3x^2\sin(5x-2) + x^3\left(\left(\frac{d}{d\_X0}\sin(\_X0)\right)\Big|_{\_X0=5x-2}\right)\left(\frac{d}{dx}(5x) + \frac{d}{dx}(-2)\right) \quad [sum]$$

$$= 3x^2\sin(5x-2) + x^3\left(\left(\frac{d}{d\_X0}\sin(\_X0)\right)\Big|_{\_X0=5x-2}\right)\left(\frac{d}{dx}(5x)\right) \quad [constant]$$

$$= 3x^2\sin(5x-2) + 5x^3\left(\left(\frac{d}{d\_X0}\sin(\_X0)\right)\Big|_{\_X0=5x-2}\right)\left(\frac{d}{dx}x\right) \quad [constantmultiple]$$

$$= 3x^2\sin(5x-2) + 5x^3\left(\left(\frac{d}{d\_X0}\sin(\_X0)\right)\Big|_{\_X0=5x-2}\right) \quad [identity]$$

$$= 3x^2\sin(5x-2) + 5x^3\cos(5x-2) \quad [\sin]$$

$$\frac{d}{dx}\left(x^3\sin(5x-2)\right) = 3x^2\sin(5x-2) + 5x^3\cos(5x-2) \qquad \textbf{(32)}$$

The "IntTutor" command allows us to see how to integrate step-by-step.
[Of course, there are many ways to go about finding an integral. Maple doesn't always choose a "good" way to do it!]

```
> IntTutor(x^2*exp(3*x),x);
```

$$\int x^2 e^{3x}\,dx$$

$$= \int \frac{e^u u^2}{27}\,du \qquad [change,\ u = 3x,\ u]$$

$$= \frac{\left(\int e^u u^2\,du\right)}{27} \qquad [constantmultiple]$$

$$= \frac{e^u u^2}{27} - \frac{\left(\int 2 e^u u\,du\right)}{27} \qquad [parts,\ u^2,\ e^u,\ 2u,\ e^u,\ u]$$

$$= \frac{e^u u^2}{27} - \frac{2\left(\int e^u u\,du\right)}{27} \qquad [constantmultiple]$$

$$= \frac{e^u u^2}{27} - \frac{2e^u u}{27} + \frac{2\left(\int e^u\,du\right)}{27} \qquad [parts,\ u,\ e^u,\ 1,\ e^u,\ u]$$

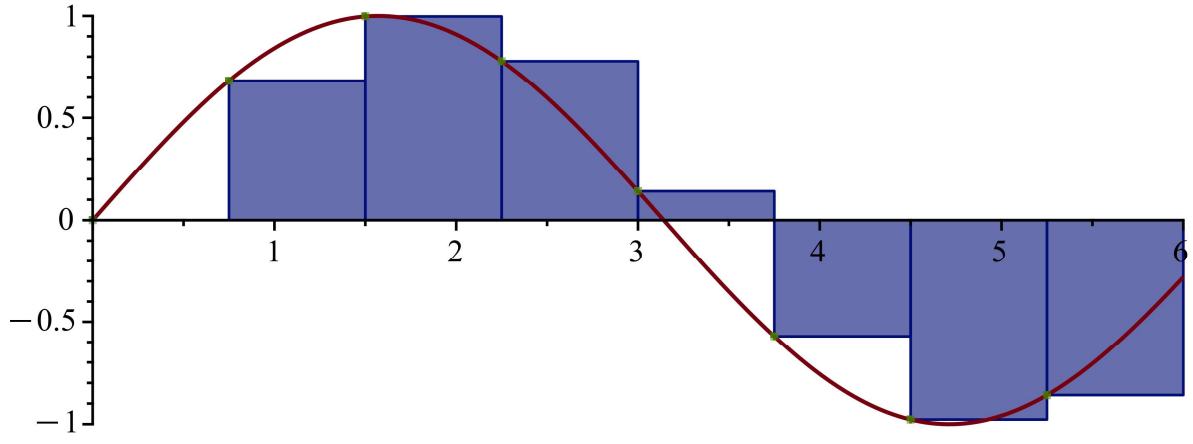
$$= \frac{e^u u^2}{27} - \frac{2e^u u}{27} + \frac{2e^u}{27} \qquad [exp]$$

$$= \frac{x^2 e^{3x}}{3} - \frac{2e^{3x}x}{9} + \frac{2e^{3x}}{27} \qquad [revert]$$

$$\int x^2 e^{3x}\,dx = \frac{x^2 e^{3x}}{3} - \frac{2e^{3x}x}{9} + \frac{2e^{3x}}{27} \qquad \textbf{(33)}$$

The "ApproximateInt" command will not only compute Riemann sums (and other approximations) but also provide plots to help us visualize approximations.

```
> ? ApproximateInt
> ApproximateInt(sin(x), x=0..6, method=left, output=plot, partition=8);
```

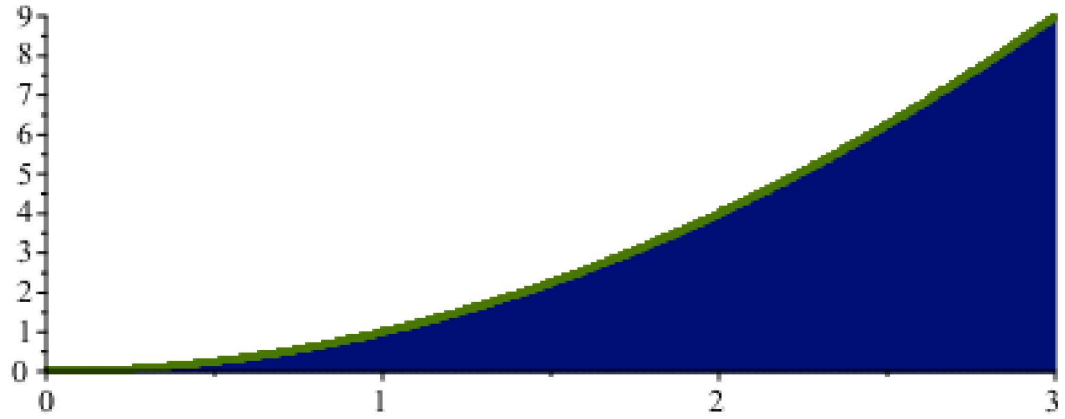


A left Riemann sum approximation of $\int_0^6 f(x)\,dx$, where $f(x) = \sin(x)$ and the partition is uni

The following command produces an animation. If you click on the plot, animation controls will appear above. Click play to see the animation.

```
> ApproximateInt(x^2, 0..3, partition=1, method=right, output=animation,
```

```
iterations=10);
```



An animated right Riemann sum approximation of $\int_0^3 f(x)\,dx$, where $f(x) = x^2$ and

Sequences, Lists, and Sets:

In Maple a collection of objects separated by commas is a "sequence".
A sequence beginning and ending with "[" and "]" is a "list".
A sequence beginning and ending with "{" and "}" is a "set".

Sets do not keep items ordered and do not allow duplication.
Lists and sequences are pretty much the same (except for notation).

```
> mySequence := 1,2,4,8,4,2,1;
```

$$mySequence := 1, 2, 4, 8, 4, 2, 1 \quad (34)$$

```
> myList := [1,2,4,8,4,2,1];
```

$$myList := [1, 2, 4, 8, 4, 2, 1] \quad (35)$$

```
> mySet := {1,2,4,8,4,2,1};
```

$$mySet := \{1, 2, 4, 8\} \quad (36)$$

To get the second item in a sequence or list use "[2]".

```
> mySequence[2];
```

$$2 \quad (37)$$

```
> myList[2];
```

$$2 \quad (38)$$

"op" gets the contents of a list and "nops" gets the number of elements in a list.

```
> op(myList);
```

$$1, 2, 4, 8, 4, 2, 1 \quad (39)$$

```
> nops(myList);
```

$$7 \quad (40)$$

```
> listOfStuff := [1,A,[1,2,3],Pi,squareFun(t)];
```

$$listOfStuff := [1, A, [1, 2, 3], \pi, squareFun(t)] \quad (41)$$

```
> listOfStuff[nops(listOfStuff)];
```

$$squareFun(t) \quad (42)$$

The "seq" command creates sequences for you.

```
> seq(i^2,i=0..5);
```

$$0, 1, 4, 9, 16, 25 \quad (43)$$

```
> listOfLists := [seq([seq(i*j,i=0..4)],j=0..4)];
```

$$listOfLists := [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8], [0, 3, 6, 9, 12], [0, 4, 8, 12, 16]] \quad (44)$$

```
> [seq(op(listOfLists[i]),i=1..nops(listOfLists))];
```

$$[0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 0, 2, 4, 6, 8, 0, 3, 6, 9, 12, 0, 4, 8, 12, 16] \quad (45)$$

Creating a procedure:

```
> hello := proc(x,y)
    local z,i;

    z := x+y;

    for i from 1 to z do
       if i=1 then
          print(cat("Howdy for the ",i, " time."));
       else
          print(cat("Howdy for the ",i, " times."));
       end if:
    end do:

    if z>5 then
       print("I'm tired.");
    else
       print("Let's go again.")
    end if:
  end:
> hello(1,2);
```

"Howdy for the 1 time."
"Howdy for the 2 times."
"Howdy for the 3 times."
"Let's go again." (46)

```
> hello(5,4);
```

"Howdy for the 1 time."
"Howdy for the 2 times."
"Howdy for the 3 times."
"Howdy for the 4 times."
"Howdy for the 5 times."
"Howdy for the 6 times."
"Howdy for the 7 times."
"Howdy for the 8 times."
"Howdy for the 9 times."
"I'm tired." (47)