Mathematica Commands To Know And Love
Calculus II

There are a number of things to bear in mind when using Mathematica:

- Mathematica is case-sensitive: `plot` (not okay) is not the same as `Plot`. For example, you must use `Pi` for π, not `pi`.

- For reasons unknown to me, brackets [ ] are *generally* preferred to parentheses.

- Use parentheses for grouping terms together as needed.

- Enter `3*x` to signify 3 mulitiplied by *x*. Similarly, enter `3*4` to signify 3 times 4.

- The symbol `^` is used to raise bases to powers. Examples: `x^2` is x^2 and `2^x` is 2^x.

- However, `e^x` is meaningless. (!!!) For the function e^x, enter `Exp[x]`.

- To enter a command, or anything new, you must get a horizontal flashing cursor. At that point, if you click the + sign to the left of the cursor, you may choose Mathematica inputs (Wolfram Language input), Free-form input (not sure what this is, but it sounds nifty), Wolfram | Alpha query, or Plain text—which you may use to annotate a Mathematica notebook.

- To get Mathematica to execute, you must either hit <return+shift> simultaneously or hit <enter> on an extended keyboard.

- After you’ve executed a command, an options bar may appear below the output. Some may be useful.

- To make output disappear, you may either add a semicolon after the command, or double-click the closest of the two brackets to the far right of the input line.

- If you’re reopening a Mathematica notebook, you MUST execute again any command you want ‘live’ in the notebook. It won’t ‘remember’ that you’d clicked it before you quit.

- The best way to figure out an appropriate Mathematica command is by searching online something analogous to ‘Mathematica differentiate.’

Command	Description
`Plot[Sin[x] + x^2, {x, 0, 2*Pi}]`	Plots the function sin(x) + x^2 from 0 to 2π
`Plot[{Cos[x],Cos[x]/2},{x,0,2*Pi}]`	Plots the functions cos(t) and cos(t)/2 together.
`Solve[a*Cos[x]-Pi*x+2*a == 0, a]`	Solves the equation a*cos(x) – π*x +2*a = 0 for a. NOTICE the "==" instead of "=" !!
`Simplify[Sin[x]^2 + Cos[x]^2]`	Simplifies the expression. Here, the result is 1.
`N[2*Sqrt[2]]`	Returns a decimal approximation of 2√2. On my computer, I get 5 significant digits.
`N[2*Sqrt[2], n]`	Returns a decimal approximation of 2√2 using *n* significant digits, where *n* is a chosen integer.
`f[x_] := x^3 +2*Sin[x]`	Defines a function. Example: f(x) = x^3 + 2sin(x). Once you've named a function, it's named.
`f'''[x_]`	For f(x) defined, this returns the third derivative. It works for any number of derivatives.
`D[x^3 + 2*Sin[x], x]`	Returns $3x^2$ + 2cos(x), the derivative of x^3 + 2sin(x).
`D[x^3 + 2*Sin[x], {x,n}]`	Takes the n^{th} derivative, where *n* is a chosen integer.
`Integrate[4*x^3 + 2*Sin[x],x]`	Returns x^4 - 2cos(x), an antiderivative of the expression $4x^3$ + 2sin(x).
`Integrate[4*x^3 + 2*Sin[x],{x,a,b}]`	This returns the value for the definite integral of the integrand, evaluated between *a* and *b*.
`NIntegrate[4*x^3+2*Sin[x],{x,a,b}]`	NOTICE that both the 'N' and the 'I' are capitalized! This returns the *numeric approximation* of the integrand between *a* and *b*. Mathematica automatically optimizes the answer. Nice.
`Abs[expression]`	Gives the absolute value of any expression.
`Sum[f[k],{k,1,n}]`	Gives the sum of a series, going from $k = 1$ to $k = n$.
`Limit[f[x], x -> infinity]`	Gives the limit of f(x) as x goes to infinity.
`Limit[f[x], x -> a]`	Gives the limit of f(x) as x goes to $a \in \mathbb{R}$
`RevolutionPlot3D[x^4-x^2,{x,0,1}]`	Rotates the graph of $x^4 - x^2$ around the z-axis.
`RevolutionPlot3D[x^4-x^2,{x,0,1}, RevolutionAxis -> {1,0,0}]`	Rotates the graph of $x^4 - x^2$ around the x-axis. Any 3-vector {a, b, c} can be the axis of rotation!
`Series[f[x],{x, a, n}]`	Gives the n^{th} order Taylor polynomial expansion to f(x), centered at x = a.