Mathematica Workshop (Introduction to Mathematica 9)

By Jessica Craig

Getting Started

1. **Click on the Mathematica icon**. You have now opened the front end of Mathematica. You will see the welcome screen. On the welcome screen, note all of the information and options there. For an overview of Mathematica, click on the video screen.

   a. **Find Your Learning Path**: This will take you to the Wolfram website where you can get instructions on using Mathematica. (Later, try following the links: under Education go to Higher Education; then under Videos and Screencasts; then go to Hands-on Start to Mathematica 8 and look at the screencast. The features in Mathematica 8 carry over to Mathematica 9)
   b. **Get Demonstrations and Source Code**: This will take you to the demonstrations in at Wolfram website. These are demonstrations that have been created by Mathematica users. (Later, you may want to explore some of these demonstrations. You can even use these demonstrations without Mathematica if you download the free Mathematica Player.)
   c. **Search Complete Documentation**: This will also take you to the Wolfram website where you can get more help with Mathematica. (We will talk more about this later.)

   To begin using Mathematica, open a **Mathematica Notebook** by clicking on **NOTEBOOK** (under **Create a New**) from the welcome screen. This Mathematica Notebook is where you will enter all of the commands for Mathematica to process, and this is where Mathematica will send all results. (This does not take you to the web.)

2. **Start typing**. (Try 3 + 4.)
   a. Mathematica automatically puts you into **Input mode**. This means that anything you type and enter is sent to Mathematica's kernel to be processed or calculated.
   b. To **enter** a line of input to be sent it to the kernel for processing, use **enter on the number keypad** or use **shift-enter** (shift-return) on the regular keypad. Note that enter (return) on the regular typing keyboard will move you to the next line on the input screen. It will not send the typed information to the kernel. (You can also use the **Enter** button on the **Basic Math Assistant** or **Basic Classroom Assistant** palettes.)
   c. Note the **In** and **Out** numbers that Mathematica has attached to your first line of type and the result. These are the numbers that Mathematica uses to stores all commands and results in its kernel.

d. Note that the cell bracket "] " at the far right. (It will be yellow while Mathematica's kernel is working. Wait until this symbol becomes black before entering a new line of input.) Each cell is connected with this symbol.

e. It is a good idea to start each new command in a new cell. If a horizontal line appears on the screen, typing a command will automatically put this command in a new cell below this line. To make a horizontal line appear at the end of a list of commands or between commands, move the cursor until it is horizontal and click the mouse. (You can also use the down arrow.)

f. Note the  on the new cell horizontal line. If you click on this + you will see options for the inputs. We will talk more about these options later in this handout.

g. You can delete a cell by highlighting the cell symbol and then hit the Delete key on the keyboard. (This deletes a command from the screen, but it is still in Mathematica's kernel.)

h. If the cell bracket stays yellow for a very long time, this may mean that you have Mathematica's kernel working in a continuous loop. If this is the case use "**Alt .**" to stop the evaluation.

i. Note the Suggestion Bar that appears below the output line. This is a new feature in Mathematica 9. It give options for commands that are related to your results. (Try "binary form" and base 2. Or try the "prime?" option.

   The  option send out to the Wolfram/Alpha website to get more information about the output.)

3. **This handout contains** some basic techniques for performing simple operations in Mathematica. It also contains a small list of some commands that you may find useful in your math courses. It is definitely not a complete list of all of the commands in Mathematica.

4. Open the **Basic Math Assistant** (**BMA**) palette. It is the first option in **Palettes** at the top of the screen. This palette is used often in this handout.

Basic Techniques

1. **Palettes**: All of the built-in commands and functions in Mathematics can be typed in directly. There are also some palettes that can be used for shortcuts to some of these commands and functions. The **Basic Math Assistant** (**BMA**) palette is a very useful one, but you may want to examine some of the other palettes as well. The best way to learn what options are available on the palettes is to experiment with them. Also note the circles with the question marks. These symbols give links to the documentation about the buttons listed in these palettes.

The **Basic Math Assistant** palette is divided into four areas: **Calculator**, **Basic Commands**, **Typesetting**, and **Help and Settings**. Under **Help and Settings**, note the **Screencast link**. It will take you through some of the useful features of this palette.

a. **Calculator – Basic:** Try the exponent palette key under **Calculator - Basic**. We will find 3^2. Get a new cell. Then click on the button with the filled-in box and an empty superscript box . This set of two boxes will appear on your notebook screen. Type 3. This number will be placed in the highlighted box. To move to the box on top (the exponent location) use the TAB key on the computer keyboard. Type the exponent 2. Then press Enter (**shift enter** on the regular keyboard, **enter** on the number keypad, or **enter** on the **BMA** palette). Mathematica will return 9. This is the basic method that you will be used for all of the operation shortcuts in this palette. (Click on the correct palette key. Fill in the highlighted box. Tab to the next box and fill it in.) **Note that you can fill in the slots of a template by tabbing or using the mouse.**

Under **Calculator – Basic** note the **Command Complete** button (which gives a list of commands that start with the entry you have typed) and the **Make Template** button (which takes a Mathematica command that you have typed and makes a template of the syntax that this command uses). Also note the **Documentation** button which will take you to the documentation in Help for any command that you have highlighted.

b. **Calculator – Advanced** contains shortcuts to the trigonometric function, inverse trigonometric functions and logs. Note that Log in Mathematica is the natural log (the base e log). It also contains other buttons which are frequently used. A button that aids in defining functions (and piecewise functions), the **Define Function** button, is also given here. Another button to note under **Calculator – Advanced** is the **TraditionalForm** button which changes the output of a command into the form you see in most textbooks.

c. Another area of the **BMA** is **Basic Commands**. This is broken up into tabs for Mathematica Functions; Algebra Manipulations and Equations Solving; Calculus and Optimization; Matrices and Linear Algebra; Lists and Tables; Two and Three Dimensional Graphing.

Let's try the **Plot** command under the **2D** tab. To graph $y = x^2$, click on **Plot** and fill x^2 into the function slot. Tab to fill in the variable *x*, its minimum and its maximum values. Note that when plotting $y = f(x)$, you will fill in $f(x)$ in the function slot (not the y = part). Then tab to fill in the variable and its minimum and its maximum values. Mathematica will set the *y* range. (You can change this range using the option key **Range**.)

Note all of the options in the suggestion bar. Note the **PlotStyle** options. Under **More**, note the **Drawing Tools**.

Look at some of the other options in the **2D** tab. Under **More** next to **Plot**, you can graph 2 or more functions on the same screen. There is also an option that will graph a parametric equation as well as several other options. Be sure to note the **Manipulate** tab at the bottom of this screen. We will talk more about this later.

d. The **Typesetting** area gives symbols that are used in text mode. Some also work in the input mode.

e. Under **Help and Settings**, note the screen casts about this palette and Mathematica Learning Center link.

2. **Basic arithmetic**: The following is a list of the symbols used for basic arithmetic: add (+), subtract (–), multiply (*, or a space), divide (/), exponents (^).

a. **You must be careful with multiplication**. Mathematica will know that an expression like $x(3x + 5)$ means to multiply x and $(3x + 5)$. It inserts a space. However, *xy* in Mathematica is a single variable, not a product. You must insert a space or * or × from the **Basic Math Assistant** (**BMA**) palette if you mean this to be a product. This also means that commands should not contain spaces unless you mean to multiply. (To expand the expression $x(3x + 5)$, highlight it and go to the **BMA** under **Basic Commands** and **y = x** and click on **Expand**.)

b. **Examples**: 1. Type 2+5^2 and Enter. Mathematica returns the value of 27.

2. Type 3*x**7*x* (or put a space between the 3*x* and the 7*x*) and Enter. Mathematica returns $21x^2$. Note that Mathematica makes the input variables a different color from the numbers. Watch for these color cues. They help you check your input lines. (Also note the options on the Computation Suggestion Bar. You could take a derivative or an integral of this result.)

3. Type Cos[Pi/2]. Note that all commands and built in functions must start with a capital letter and the arguments for all commands and function must be in square brackets. Note the Command Completion options that appear as you start typing the function.

3. **Free-Form Input**: You can also enter commands using English and letting Mathematica look for a command to carry out your request. There are 3 ways to enter a line of type using free-form input. To access this option, start a line in input mode with an equal sign. Then type in words the operation you wish

Mathematica to carry out and enter. (Be sure to use the shift-enter from the regular key board or enter from the number pad.) This will send the request to the Wolfram website and there Mathematica will search for a command that it thinks you want and execute this command. If you click on the small plus sign on the top right, you will get related commands that Mathematica thinks you may be interested in. Try typing "= sine of 30 degrees".

Another way to get free-form input is under Insert tab at the top of the screen. You can also use free form input by clicking on the small  on the left of the new cell horizontal line. Also note the other options here. Particularly note the **Wolfram/Alpha query** option. This option goes out to the web to go to the Wolfram/Alpha site where there is a great deal of calculable data. (Click on this option and try typing "Atlanta population".) Also note the **Other styles of text option**. You can use this to change from the default input mode to other options such as text mode or title mode which lets you add text to your notebook. (When you get a new cell, which is indicated by a new horizontal line, you will be back in input mode.)

4. **Universal Constants**: 1. The natural base e has symbol **E**. 2. The square root of -1 ($\sqrt{-1}$ or i) has symbol **I**. 3. The ratio of the circumference to the diameter of a circle (or π) has symbol **Pi**. 4. The symbol **Infinity** (or ∞) can be used. (You can also access these constants from the **Basic Calculator** option on the **BMA** palette.)

5. **Built-In Functions and Commands**: When typing in built-in functions or commands in Mathematica there are some important things to keep in mind. (Note that the templates given in the palettes will help with this syntax. You will probably want to use these palette options until you become more familiar with the command syntax.) Try the Sine function from palette.
   a. **Capitalize** the first letter of all built-in commands and functions! Use **square brackets** (that is "[" and "]") for the arguments of all functions and commands!
   b. **Examples**: The sin 30° in Mathematica is **Sin[30 Degree]**. (The Degree command is also on the Basic Math Assistant palette as a small circle under Pi.) The square root of 4^2 is **Sqrt[4^2]**. (The square root function is also on the Basic Math Assistant palette.) To obtain the prime factorization the integer 36 use **FactorInteger[36]**. (Note that the output uses curly brackets which list, first the factor then the number of times it is repeated in the factorization.)
   c. Be sure to **use the correct spelling** of the built-in function names and commands. Mathematica has a **command completion feature**. Type a partial function name and a list of functions that start this way appears. (Be sure to start with a capital letter.) Select the desired command, (Also note that this option is listed in the **Basic Math Assistant** palette under **Calculator**.)

d. Also note that a command or function will **show on the screen in blue** until it is typed as a command or function that Mathematica recognizes. It then turns black. (This is also true for user defined functions.)

e. If you have entered a command or function improperly, Mathematica will give you an **error message** or the repeat of the command. (Be sure to read these error messages. They can sometimes be very helpful.)

6. **Grouping symbols**: Use square brackets ("[ ]") only for arguments of functions and commands. Use curly brackets ("{ }") only for lists of objects. The only symbol that can be used for grouping is a set of parentheses ("( )").

7. **Equal signs**: Mathematica uses one equal sign to denote a definition. Equations require two equal signs. On the **BMA** palette go to the **y = x** tab under **Basic Commands**. Go to **Solve** and enter the equation $x^2 + 6x - 7 == 0$. Then tab to var and enter x. Then enter to get the solution.

8. **User defined function and constants** can be useful. You can define constants and functions with letters or words. But, it is a good idea to start all user defined constants and functions with lower case letters so that they will not be confused with any built-in functions or commands. It is also a good idea to use the command **Clear[ ]** before using any letters or words for names.

   a. To **define a constant**, use **=** . **Example**: If you want a to have the value of 5, type **Clear[a]** and enter. Then type **a = 5** and enter. After that, every time you use the letter a, Mathematica will give it the value of 5. (Mathematica assumes that letters not previously defined are variables. Thus, if you have not defined b to have a value and you type 3 b + 2 b and enter, Mathematica will return the value 5 b.)

   b. To **define a function**, use **f[x_]:=** . **Examples**: To give g the formula x^2, type **Clear[g]** and enter. Then type **g[x_]:=x^2** and enter. After this, anytime you use g[x] Mathematica uses the formula x^2. (For example g[2] and enter returns the value of 4.) You can also use the **Define Function** button on the **BMA** palette under Advanced Calculator to define a function.

   c. **Piecewise Defined Functions** can be defined using the **BMA** palette. Use the **Define Function** button under **Calculator –Advanced**. Fill in the name of the function. Tab to the equation space and use the button

   $$\begin{cases} \square & \square \\ \square & \square \end{cases}$$

   to fill in the equations and the conditions. More rows can be added using the **Row+** key on this same palette location.

9. **Decimal Approximations**: Whenever possible, Mathematica gives **exact answers**. If you want a decimal approximation for an exact answer use **N[ ]**. For

example **N[11/9]** gives 1.22222. You will also obtain approximate answers when an input number contains a decimal.

Mathematica uses 16 significant figures in its calculations. You can use the command **N[Pi, n]** to obtain n digits of accuracy, where n is an integer.

10. The **Help** option at the top of the screen is very useful. It contains all of the information in the Mathematica user manual. (You can also access help from the BMA palette under **Documentation**.) Also, as you start typing a command, the command completion box appears. On the top right of this box, there is a symbol (a box with 4 lines). If you click on this symbol, an arrow will appear that it will take to you the documentation for the highlighted command.

    Under **Help**, the **Documentation Center** and **Function Navigator** sections are useful in finding commands and information about these commands. You can use the categories in these sections to find the commands or functions that you need for various operations. Or you can type a command or function that you want more information on in the window at the top of these screens.

    Examine the categories in the **Documentation Center**. Note the category called "**COMPUTABLE DATA**". In this section, Mathematica takes you to the Wolfram/Alpha site on the web where there is a great deal of calculable data. Let's try "Scientific & Technical Data", **ChemicalData**. (Recall that you can also go to the Wolfram/Alpha site by using the + on a new cell line.)

    Anlther method for finding information about a command or function without going to the help screen is to type a question mark before the command and **Enter**. For example: **?Plot** brings up information about the plot command. If you then click on the << symbol it takes you to the documentation on the Plot command. The command **Options[Plot]** will give you the list of the options associated with the plot command and their default values.

11. **Modes**:
    When you first open a Mathematica notebook and enter a line of type, you are in **Input mode**. Any new cell is also automatically in Input mode. This is the mode you will use to input functions and commands. When you type in this mode and **Enter**, everything you type is sent to Mathematica's kernel to be processed. After it is processed in the kernel, the result is returned to you as **Output**. (Note the line numbers that Mathematica automatically puts on the screen. These numbers keep track of the order in which calculations were performed. Note also that your input is in bold type and the output is not.)

    Sometimes you will want to include text with your calculations. This can be done by using **Text Mode**. You can change to Text Mode using the **Create Text Cell** on the **Calculator** area from the **BMA** palette. You can also access text mode (and other modes) from **Format** on the tool bar or using the  on the new cell line. Under **Format** go to **Style** and pick the style you desire. The choices include Text, Title, Subtitle, Input among others. You can also open a tool bar that shows a window that indicates the mode of any given cell. To open this toolbar go to the **Window** option at the top of the screen and go down to **Show ToolBar**. Then to change the mode of a cell, highlight the cell symbol on the far right of the line of type you want to change. Then choose the style option from the choices given in the style window. Text can also be inserted in the same line as a command by using the format (* your text *).

12. **Saving and printing** can be done under the file menu.
    Saving: Before you save your document it is a good idea to use the **Delete All Output** command in the **Cell** menu at the top of the screen. This will save memory space because only the commands will remain. (This is particularly important if your document includes three-dimensional graphs which can require a great deal of memory.) To have the input re-evaluated, use the **Evaluate Notebook** option in the **Evaluation** menu.

13. **If all else fails**: Mathematica keeps all inputs from a session in its memory, even if they are deleted from the screen. If you find that commands are not working even though you have properly typed them, it could be because you have unknowingly defined something in an earlier step that is confusing Mathematica. If this happens, delete all incorrect commands. (You can use the **Delete All Output** - in the **Cell** menu - to see just a list of current commands.) Then, use **Quit Kernel** under the **Evaluate** menu and go over to **Local**. This will end your session and erase the commands from the memory in the kernel without removing the current commands from your screen. To restart the session, use **Evaluate Notebook** command in the **Evaluation** menu. This will evaluate only the commands that are on the screen and any incorrect commands that were erased are now no longer in the memory.

14. **To make typing easier**:
    a. You can use the **Copy** and **Paste** commands under **Edit** in the tool bar.
    b. You can position a new line of input under a previous line and use the **Input from Above** or **Output from Above** buttons on the **BMA** palette. This will copy the input or output information from the above line.
    c. You can make the output of one command the input of another in one step using **//**. For example: **Together[2/5 + ¾]** returns 23/20 but **Together[2/5 + ¾]//N** returns 1.15.
    d. The ReplaceAll command has the short cut /.
       Example: x^2*y/.{x->3,y->-1} replaces *x* with 3 and *y* with -1.

15. **Packages**: Mathematica does so many different operations that they cannot all be called up when you first open the software. Some of them are stored in packages that you can access during a session. Under "**Function Navigator**" in the **Help** menu, note the list of available **Packages**.

To load a package, uses the symbol **<<** and the name of the package. (Another way to load a package is to use the command **Needs**.)

For example: Open the package **<<Units`** (Note the use of the backwards single quote in the name of this package. This is the key in the top left of the keyboard.)

<<Units`
ConvertTemperature[30,Centigrade,Fahrenheit]

This returns the Fahrenheit temperature for 30^0 Centigrade.

One caution: **don't use any commands defined in a package before you load the package.** Mathematica will assume that you have defined the command and so it will not use it's built in definition. (If you do use a command before you have opened the package, quit the kernel then place call the package before applying the command.)

16. We have been creating a Notebook in Mathematica. Note that you can create this notebook in several formats including a slide show format. (To create a slide show go to **File – New – Slide Show**.)

17. This is just a small sample of things that you can do with the Mathematica software.

Commands:

Built-In Functions

Built-In Functions: Frequently encountered functions include the following: **Exp[x]** (gives e^x); **Log[x]** (gives ln *x*); **Log[10,x]** (gives the base ten log of *x*); **Abs[x]** (gives the absolute value of x); **N[x]** give the decimal approximation of *x*; **Sqrt[x]** (gives the square root of *x*). **Floor[x]**: Is the greatest integer function f(x) = [[x]]. **Factorial[n]** : gives n! = 1·2·3·4·…·n .

Trig functions were *x* is in radians **Sin[x]**, **Cos[x]**, **Tan[x]**, **Cot[x]**, **Sec[x]**, **Csc[x]**; (For angles in degrees, use the function **Degree**. For example **60 Degree** for 60^o. Also note that $\sin^2 x$ is **Sin[x]^2** in Mathematica.

Inverse trig functions **ArcSin[x]**, **ArcCos[x]**, **ArcTan[x]**, **ArcSec[x]**, **ArcCsc[x]**, **ArcCot[x]**;

Hyperbolic function and their inverses: **Sinh[x]**, **Cosh[x]**, **Tanh[x]**, **ArcSinh[x]**, **ArcCosh[x]**, **ArcTanh[x]**.

Graphing

Graphing: (The coordinate axes are positioned in a readable location. Thus, the axes do not always intersect at the origin. Be careful to note the position of the intersection of the axes.) Many of these commands are listed in the **Basic Commands** section of the **Basic Math Assistant** palette under the 2D or 3D tabs.

1. **Plot[f[x], {x, xmin, xmax}]** graphs the function $y = f(x)$ on the x interval [$x$min,$x$max].
2. **Plot[{f[x], g[x]}, {x, xmin, xmax}]** graphs the two functions $y = f(x)$ and $y = g(x)$ on the same coordinate axes over the x interval [$x$min, $x$max].
   A nice command to use with a set of functions is **Tooltip**. For example, **Plot[Tooltip[{f[x], g[x]}], {x, xmin, xmax}]** not only graphs the two functions but also displays the function formula for each graph when you move the cursor on top of the graph.
3. There are many ways to customize a plot. Right click on the graph and call up the **Drawing Tools** option. Or use the **More** option on the suggestion bar. (You can also find this tool box under the **Graphics** menu or the **Drawing Tools** button at the very bottom of the **2D** tab on the **Basic Commands** section of the **BMA** palette.) You can use this tool to add lines, arrows, circles, boxes, and dots to the graph. The symbol ∑ ≡ is used to place equations on the graph and the A≡ symbol is used to place text on the graph. You can also change the look of the graph by double clicking on the portion of the graph that you want to change. Then use the options in drawing tools to change the graph.
4. There are a lot of options that work with the plot command. To include these option is a graph, list the option inside the square brackets of the plot command after the limits for x. Also note that many of these options are listed on the **BMA** palette under **Basic Commands** in the **2D** and **3D** tabs. (To use these templates, position the curser at the end of the Plot command and click on the option from the palette. The template for the option will then be position properly in the Plot command.) Some useful options include the following.
   a. **AxesLabel-> {x,y}** places the labels x and y on the axes. (You can also place words or equations on the axes with this command.)
   b. **PlotLabel->"Name"** puts Name at the top of the graph.
   c. **PlotStyle-> Red** or **Thickness[.01]** or **Dashing[Large]** can be used to change the color of the graph to red, or thickness of the graph to make the curve thicker or to draw the graph as a dotted curve. (Other colors, thicknesses could also be used. You could also use Dashing[Tiny or a number like 0.1].)
   d. **AspectRatio -> Automatic** makes the tick marks on the graph the same for the x- and y-axes.
   e. **PlotRange->{ymin,*y*max}** sets the minimum and maximum values of the y-axis.
   f. **Ticks->{{-Pi, -Pi/2, 0, Pi/2, Pi},Automatic}** is a option that labels the tick marks on the x axis with $-\pi, -\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi$ and the y axis is labeled automatically. (Ticks option specifies the labels on the axes.)
   g. **AxesOrigin->{0, 0}** will cross the axes at the point (0, 0).

5. **ListPlot[{ {a,b}, {c,d} }]** Plots the list of ordered pairs on numbers in two dimensions. A useful option with this command is **PlotStyle->PointSize[0.025]**. This will change the size of the dots. (Other numbers besides 0.025 could be used.)
6. **To graph odd roots** you must use the Abs and Sign[x] commands. Otherwise, Mathematica will default to complex numbers and show only the portion of the graph found using positive *x* values. For example: To graph $y = \sqrt[3]{x}$ use **Plot[Sign[x]$\sqrt[3]{\text{Abs}[x]}$, {x,-3,3}]**
7. **To graph implicitly defined functions** (equations) use the command **ContourPlot**. (Don't forget to use two equal signs for the equation.) For example: To graph $x^2 + y^2 = 4$ use the command **ContourPlot[$x^2 + y^2 == 4$, {x,-3,3}, {y,-3,3}] (ContourPlot3D** graphs equations in three variables.)
8. **ParametricPlot[{f[t], g[t]}, {t, tmin, tmax}]**
   Graphs the parametric equations $x = f(t)$, $y = g(t)$
9. **Show[plot1, plot2]** will paste the two graphs in plot1 and plot2 together. (Many of the options that apply to the Plot command also apply to the Show command.) For example:
   Show[Plot[x^2,{x,-3,3}], ListPlot[{1,4,9}, PlotStyle->PointSize[.02]]
   will graph the curve $y = x^2$ and dots at the points (1,1), (4,2) and (3,9).
   (You can also define each graph and use the definitions in the Show command. For example:
   p1 = Plot[x^2,{x,-3,3}]
   p2 = ListPlot[{1,4,9}, PlotStyle->PointSize[.02]]
   Show[p1, p2]
10. Three dimensional graphs can be graphed using **Plot3D** or **ParametricPlot3D**. Note that you can **rotate the graph** by clicking on the graph and dragging the mouse.

Manipulate

1. **Manipulate**: The Manipulate command creates an interactive display that varies with a parameter. It is located on the **BMA** palette under **Basic Commands** using the **2D** or **3D** tab (at the bottom of the palette). Also note the **Manipulate Control** button that formats the control for this command. This command allows you to manipulate variables along a slider. For example:
   Manipulate[Plot[Sin[a*x], {x, -2Pi, 2Pi}],{a,.5,5}] creates a slider graph that changes the period of the sine function starting at the value 0.5 and ending at 5.
2. **Manipulate[Plot[Sin[a*x], {x, -2Pi, 2Pi}],{{a, 2, "Period"},0.5, 5}]** Labels the slider as Period and starts the slider with the initial point 2. The slider values (a values) also begin and end with the values of .05 to 5.
3. You can click on the + next to the slider to see other options for changing the display.
4. The Manipulate command works with more than just graphs. It also works with more than one parameter.

Algebra

1. **Expand[*expression*]** : Multiplies the *expression*.
2. **PowerExpand[*expression*]** : expands all powers and products in the *expression* (assuming all variables are positive when inside roots with even indices).
3. **ExpandAll[*rational expressions*]** : Expands all products and integer exponents in any part of the expression (even in denominators).
4. **Factor[*expression*]** : Factors the algebraic expression over the real numbers. (Use **FactorInteger[ ]** to factor integers.)
5. **Simplify[*expression*]** : Performs certain algorithms to try to simplify the expression.
6. **FullSimplify[*expression*]** : Performs more algorithms and so it can simplify more complicated expressions but it takes longer.
7. **Together[*sum of rational expressions*]** : Adds the rational expressions and cancels common factors from the result.
8. **Apart[*rational expression*]** : Performs partial fraction decomposition on a rational expression. (That is it "unadds" fractions.)
9. **Cancel[*expression*]**: factors the numerator and denominator of an expression and returns the expression in lowest terms.
10. Solve equations (if possible): (Note: In Mathematica, an e*quation* is indicated with a double equal sign. For example: 2x^2 – 3 == 5x. So == means equality and = means assignment.)
    a. **Solve[** *equation*, *variable*] : Solves the equation for the variable. This works best with polynomial or rational equations. (**Solve[**{eq_1, eq_2},{x,y}**]** solves the system of the two equations in terms of the variables *x* and *y*.)
    b. **NSolve[** *equation*, *variable* **]** : Gives the numerical approximations for the solutions of the equation. This command can also be used for a system of equations the same way the Solve command can be used to solve a system.
    c. **FindRoot[** *equation*, {x, x_0} **]** : Approximates the solution to the equation using a variant of Newton's method starting at the value x_0. (You want the starting value to be close to the solution. You may want to graph the equation to find this starting value.) This command often works better than NSolve for non-algebraic equations.
11. The **replace all** command is **/.** . For example, the input **2x + 1 /. x – > -3** yields the result -5. Mathematic has replaced every occurrence of *x* in the expression on the left with the value of -3. (Note also the button **/.** on the **Basic Math Assistant** palette under **Calculator-Basic**.)
12. The **Basic Math Assistant** palette under **Basic Commands** in the ***y = x*** tab includes many of these commands. To use a button from this palette, highlight the expression you want to work with and then click on the command you want to use in the palette.
13. Mathematica does not automatically return the real cube root of a negative number. To get the real cube root, use the **Abs** and **Sign** commands. For example: To get $\sqrt[3]{-27} = -3$ type **Sign[-27]***$\sqrt[3]{\boldsymbol{Abs}[-27]}$.

Tables

1. **Table[f(n), {n, nmin, nmax}]** : Creates a table of values by replacing n in the function with successive values starting at nmin and ending at nmax with a step size of one.
2. **Table[f(n), {n, nmin, nmax, #}]** : Creates a table of values by replacing n in the function with successive values starting at nmin and ending at nmax with a step size of #.
3. Tables of lists of functions can also be created. Also the function could contain more than one variable which increments according to the step size.
4. The command **TableForm** puts the list in the form of a table instead of a list in curly brackets. **TableHeadings** and **TableSpacing** are nice options to include in tables that are in table form. For example:
   TableForm[Table[n^2 - j, {n, -2, 2},{j, -2, 2}],TableSpacing->{1,4}]
   Puts the table in a rectangular array with one line spacing between rows and 4 spaces between columns.
5. The command **MatrixForm** puts the list in the form of a matrix.
6. Also notice that many of these commands are also listed on the **Basic Math Assistant** palette under **Basic Commands – List**. (You can even create a regression equation from the data under **Statistics** on this palette.)

Trigonometry

1. Note the trigonometric functions and their inverses which are listed in the **Basic Math Assistant** palette under the $\sqrt{x}$ button in **Basic Commands**. (The hyperbolic functions are also listed here.)
2. Trigonometric functions: **Sin[x]**, **Cos[x]**, etc.
3. Inverse trigonometric functions: **ArcSin[x]**, **ArcTan[x]**, etc.
4. To change from radians to degrees: Multiply by **Degree** or use **°** in the Basic Math Assistant palette. (For example: **Sin[30 Degree]**.)
5. To change from degrees to radians: Divide by **Degree** or by **°**. For example: **N[(π/2)/Degree]** yields 90**.**
6. The ordinary **Plot** command will also graph trigonometric functions. If you would like to have the *x*-axis labeled in multiples of π, use the option **Ticks**. For example:
   Plot[Sin[x],{x, -π, π},Ticks–>{{ -π , -π/2, 0, π/2, π}, Automatic}] , gives a nice graph of the sin function.
7. Many of the functions used to simplify and manipulate algebraic functions also work for trigonometric expressions. For example: **ExpandAll[*rational expressions*]**, **Simplify[*expression*]**, and **FullSimplify[*expression*]** work as they do in algebraic expressions. (These are listed in the Basic Math Assistant palette under Basic Commands in the *y* = *x* tab. Also note the Algebraic Trig button.)
8. **TrigExpand[*expression*]** and **TrigFactor[*expression*]** use more of the trigonometric identities to perform their manipulations. **TrigReduce[*expression*]** if possible, rewrites the expression in an equivalent form without exponents. (You can find these options on the **Basic Math Assistant** palette in the **y = x** tab.)

9. Mathematic does not **prove trigonometric identities**. But you could try subtracting the left-hand side of the equation from the right-hand side and simplifying the resulting expression. If Mathematica returns the value of 0, then the sides were equivalent.

Calculus I

1. **ParametricPlot[{f[t], g[t]},{t, tmin, tmax}]** : Graphs the parametric curve (f(t), g(t)). You can graph more than one curve also by using a list of the parametric functions. (Note that you can set the limits on the x and y using **PlotRange->{{x1,x2},{y1,y2}}**.)
2. **Limit[*function*, x –>a]** : Takes the limit of the function as x approaches a. Note that a could be a real number or infinity.
3. **Limit[*function*, x –>a, Direction –> 1]** : Takes the one-sided limit as x approaches a from the left. **Limit[*function*, x –>a, Direction –> – 1]** : Takes the one-sided limit as x approaches a from the right.
4. **f '[x]** or **D[f[x], x]** or **∂_x[f[x]]** on the palette: Differentiates the previously defined function *f(x)* with respect to *x*. You may also use **D[function, x]** or $\partial_x(function)$ to differentiate a function that has not been defined.
5. **D[f[x], {x, n}]** : Finds the n^{th} derivative of the function f[x].
6. **Dt[equation,x]** : Gives the total derivative of the equation. This can be used to implicitly differentiate the equation to find dy/dx (which has the form Dt[y,x] here) by using the **Solve** command. For example:

   Dt[x^2+y^2==4,x]

   Solve[%,Dt[y,x]]. (Recall that % means the previous output by number.)

   You can also implicitly differentiate by finding the derivative with respect to *x* using the command **D[equation, x]** were the equation is equation in *x* and *y*[*x*] (replace *y*'s with *y*[*x*]). Then solve the result for y'[x].
7. **Integrate[*function*, x]** or **∫*function* dx** (using **∫□ d□** from the palette): Finds the indefinite integral of the function with respect to *x*. (Note that the answer omits the constant of integration. You must include this constant.)
8. **Integrate[*function*, {x, xmin, xmax}]** (or use the definite integral from the palette): Evaluates the definite integral with lower limit xmin and upper limit xmax. Mathematica will try to integrate symbolically first. If there is not a nice antiderivative for Mathematica to use, it will return the integral as its answer.
9. **NIntegrate[function, {x, xmin, xmax}]**: Evaluates a definite integral using numerical methods. If possible, it returns a decimal approximation of the value of this integral.
10. Note that many of these commands are given in the $d\int\sum$ tab.

Calculus II

1. To find the surface of revolution when the function $y = f(x)$ is revolved about the y-axis, use **RevolutionPlot3D[f(x), { x, xmin, xmax}]** or **ParametricPlot3D[{x*Cos[t],x*Sin[t],f[x]},{x, xmin, xmax},{t,0,2Pi}]**.

   To revolve about the x-axis, use **RevolutionPlot3D[f(x), {x, x, xmin, xmax}, RevolutionAxis->{1, 0, 0}]** or **ParametricPlot3D[{x, f[x]*Cos[t], f[x]*Sin[t]},{x, xmin, xmax},{t,0,2Pi}]**.

   These commands can be found on the palette under **Basic Commands** in the **3D** tab.

2. Mathematica can often evaluate convergent **improper integrals**. You can use ∞ or –∞ in the upper or lower limits of an integral. For example: $\int_1^{\infty} \frac{Log[x]}{x^2}\, dx$ entered from $d \int \sum$ tab, yields a value of 1. Mathematica will also handle limits of integration at points were the integrand is discontinuous.
3. Series (even an infinite series, if it converges) can often be evaluated by Mathematica. This can be done by using the summation symbol $\sum_{nmin}^{nmax}[]$ or the command **Sum[f[n],{n, nmin, nmax, step}]** both in the $d \int \sum$ tab on the **Basic Math Assistant** palette. (The step amount default value is also one.) The limits of summation could be ∞ or –∞. For example: **Sum[(1/2)(1/10)^n, {n, 0, ∞}]** yields 5/9. **NSum** gives a numerical approximation of the sum.
4. **Series[*f*[*x*], {*x*, *a*, *n*}]** finds the first n terms of the Taylor's series expansion for $f[x]$ centered about a.
5. **PolarPlot[f[θ],{ θ, θmin, θmax}]** Graphs the polar equation $r = f(\theta)$.

Calculus III

1. **ParametricPlot3D[{f[t], g[t], h[t]},{t, tmin, tmax}]** : Graphs the parametric curve $(f(t), g(t), h(t))$ in three dimensions.
2. **Plot3D[f[x,y], {x, xmin, xmax}, {y, ymin, ymax}]** : Graphs the curve $z = f(x, y)$ in three dimensions. A nice option to include is **AxesLabel -> {x, y, z}**. You can rotate the graph by clicking on the graph and rotating with the mouse.
3. **ContourPlot3D[f[x,y,z]==0,{x,xmin,xmax},{y,ymin,ymax},{z,zmin,zmax}]** can be used to graph the implicitly defined function in the variables x, y, and z given by $f(x,y,z) = 0$.
4. The derivative command D works for functions of more than one variable. For example **D[f[x,y],{x,2},y]** give $\frac{\partial^3 f}{\partial x^2 \partial y}$.
5. Multiple integrals can be evaluated in Mathematica by nesting integrals.

Differential Equations

1. To plot a slope field for the differential equation $\frac{dy}{dx} = f(x, y)$ use the command **VectorPlot[{1,f[x,y]},{*x*, *x*min, *x*max},{*y*, *y*min, *y*max}]**. (This command is in the palette in the 2D tab.)
2. To solve a differential equation $\frac{dy}{dx} = f(x, y)$, use the command **DSolve[*y*'[*x*] == *f* [*x*, *y*[*x*]], *y*[*x*], *x*]**. (Be sure that you use two equal signs in the equation. Also, be sure that you use *y*[*x*] every time there is a *y* in the equation.)
3. To solve an initial value problem $\frac{dy}{dx} = f(x, y)$, $y(x_0) = y_0$, use the command **DSolve[{*y*'[*x*] == *f* [*x*, *y*[*x*]], y[x_0] == y_0}, *y*[*x*], *x*]**. (Be sure that you use two equal signs in the equation and the initial condition. Also, be sure that you use *y*[*x*] every time there is a *y* in the equation.)

   Or, if you want a decimal approximation for solution to the initial value problem on the interval $a < x < b$, use the command **NDSolve[{*y*'[*x*] == *f* [*x*, *y*[*x*]], y[x_0] == y_0}, *y*[*x*], {*x*, *a*, *b*}]**. (Of course, x_0 should be in the interval (*a*, *b*).)

   These commands are on the palette under the $d \int \sum$ tab.

4. To plot a solution to a differential equation, use the **Plot** command. It often helps to also include the **Evaluate** command. For example:

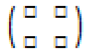
   sol = NDSolve[{*y*'[*x*] == *f* [*x*, *y*[*x*]], y[x_0] == y_0}, *y*[*x*], {*x*, *a*, *b*}].
   Plot[Evaluate[y[x] /. sol], {x, a, b}, PlotRange $\rightarrow$ All]

Linear Algebra by Barrett Walls

1. To row reduce a matrix use the command **RowReduce[A]**. Your matrix A can be entered as a list of lists of the rows. Thus, **{{1,2,3},{4,5,6}}** is the matrix whose first row is 1,2,3 and whose second row is 4,5,6.

   Alternatively, it can be entered a matrix using the $\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$ button from the **Basic Commands** section of the **Basic Math Assistant** palette. Mathematica will convert this matrix to the list of lists format. (Note that more rows and columns can be added using the **Add Row** or **Add Column** keys.) You can change from a list of a list notation to matrix form using the command **MatrixForm**.

2. To take the determinant of matrix A use the command **Det[A]**.
3. To find the inverse of a matrix use the command **Inverse[A]**.
4. To find the transpose of a matrix use the command **Transpose[A]**.
5. To multiply matrices A and B use a period, **A . B**. To take a matrix A to the power n use the command **MatrixPower[A,n]**.
6. To find the null space of a matrix use the command **NullSpace[A]**. Note that Mathematica returns a basis for the null space.

7. To find the eigenvalues of a matrix use the command **Eigenvalues[A]**. To find the eigenvectors of a matrix use the command **Eigenvectors[A]**. Alternatively, you can use **Eigensystem[A]** to get both the eigenvalues and eigenvectors.
8. To find the LU Factorization of A use the command **LUDecomposition[A]**. Note that your output will be a matrix and a list. The matrix is the combination of the L and U matrices and the list specifies the rows used for pivoting.
9. To take the dot product of two vectors **u** and **v** use **Dot[u,v]**. Your vectors are entered as lists, e.g. **{1,2,3}** is the vector (1,2,3).
10. To take the cross product of two vectors **u** and **v** use **Cross[u,v]**.
11. These commands can be found on the **Basic Math Assistant** palette under **Basic Commands** at the  or the **Lis**t tabs.

References: *Mathematica by Example*, 4th Edition by Abell and Braseton; The Help documentation in Mathematica and Mathematica's screencasts.