Intro to Using *Mathematica* for Math 136

New Commands/ Practice using *Mathematica*

Helpful Hints

Mathematica is a very strong tool for doing many mathematical tasks as well as illustrating concepts we are learning in class. It can be frustrating at first and it takes sometime to get used to the conventions of using Mathematica. Lets start out with some basic tips that will help us out:

1. All *Mathematica* commands begin with a CAPITAL (uppercase letter) and for variables *Mathematica* is case sensitive. In other words, *Mathematica* understands Cos[2] but it doesn't know what cos[2] is. Also, the letters "A" and "a" would represent different variables.

2. *Mathematica* commands use square brackets to enclose their arguments. Curly brackets are used for lists of objects. For instance Cos[x] is used, not Cos{x} or Cos(x). If we want to write a point we use the list notation {x, y} not [x,y] or (x,y) and if we want to do operations we wrtie 3^(7-4) not 3^{7-4}.

3. Save often. As soon as you create a new document save it. Then save it frequently. Once you have saved it initially you can type Apple+S (or control+S) to save.

4. Some letters are not allowed as variables as they have permanent values assigned to them. These letters are C, D, E, I, N, and O.

5. To execute a line you must hit "Shift+Return" or "Enter", the "Return" key alone will just give you a new line, but will not execute anything.

6. There are many shortcuts in *Mathematica*. I will try to point them out as we go. (for example if you type "Esc, p, Esc" the character π will be typed. Also there are many things that can be done with Palettes, which can be found in the menu bar at the top. These can be particularly helpful when typing text in *Mathematica*.

7. Typing text in *Mathematica* requires that you tell *Mathematica* that the cell is text. The blue brackets on the right denote different "cells" each cell is a certain type (text, input, output, etc.). The default when you start to type is that the cell is an input cell. There are two ways to change a cell to a text cell.   The first is if you see a vertical line across the screen and then you type "Apple+7" or "Alt+7" then a text cell formatted correctly will be created immediately under the line.   The second way to do this is useful if you have a cell already typed that you want to convert into a text cell. Click on the cell bracket on the right side of the screen that you want to

convert to text, then click "Apple+7" or "Alt+7" to convert the cell to text. You can also "Divide Cells" and "Merge Cells". You can play around with this if you like later. There are also “levels” of text cells that you can format like an outline (“Section,” “Subsection”, etc.) You can find these options and the corresponding shortcuts by selecting “Format” and then “Style” from the menu bar at the top of the window.

8. If you type an input line followed by a ";" the output will be suppressed.

9. Lets look at some basic commands now:

Functions, Plotting and other Basics

The following commands define the functions f(x), g(t) and h(x). Notice how when defining a function you give the name followed by square brackets containing the variable **and an underscore.** You only need the underscor on the left hand side of the definition. Also notice how you can define piecewise functions like h(x) below too. Evaluate the cell below by hitting “Shift+Enter”

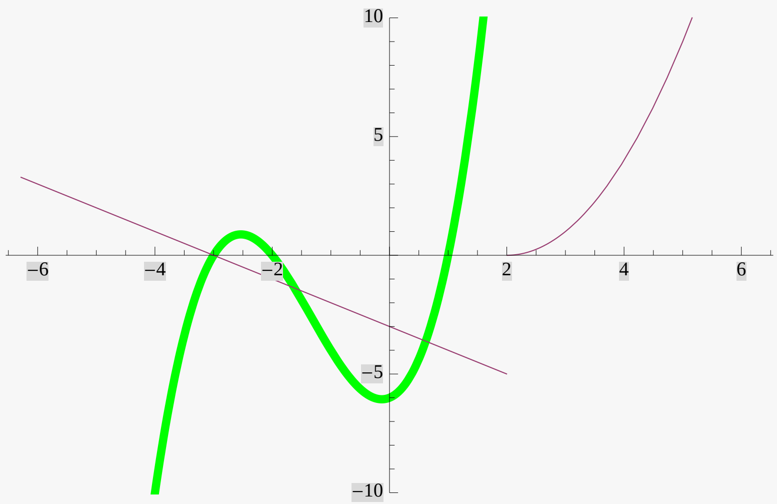
$$f[x\_] = \frac{\text{Cos}[x] - 1}{x}$$

$$g[t\_] = (t + 2)\ (t + 3)\ (t - 1)$$

$$h[x\_] = \begin{cases} -3 - x & x < 2 \\ (x - 2)^2 & x \geq 2 \end{cases}$$

The Plot command creates graphs of functions over a given domain. There are many options that can be used in the Plot command as well (different colors, labels, etc.). You can find many of them in the help section. If you look in the help menu (called “Documentation Center”) you can see how to change the color or thickness of graphs etc. Also, after a graph is displayed you can open the window “Drawing Tools” under the Graphics menu. With that you can click on the graph and make changes to it. After hitting shift enter on the graph below. Click on the blue curve and try to use the drawing tools to change it to green and make it thicker.

```
Plot[{g[x], h[x]}, {x, -2 π, 2 π}, PlotRange -> {-10, 10}]
```



Its a good idea to clear variables when you are done using them:

```
Clear[f, g, h]
Clear[x, y]
```

The N command returns a numerical approximation with a specified number of digits. If for some reason you are having a hard time displaying digits you might have to change a setting in *Mathematica* by selecting “Edit”, “Preferences”, “Appearance”, “Numbers”, “Formatting” and then chaging the “Numbers Displayed” window to a larger number.

```
N[π, 73]
```

Integration and Differentiation

Integration and Differentiation can be done rather easily in *Mathematica* as well. Here is an example of one anti-derivative, one definite integral, and two derivatives.

```
Integrate[x^2/(x^3+2)^2, x]
```

```
Integrate[x*Exp[-x], {x, 1, ∞}]
```

```
g[x_] = 3 x^2 Cos[x]
g'[x]
D[g[x], x]
```

```
l[x_] = Log[x+1]
l'[x]
```

Solving Equations

The Solve command solves given equations for given variables. The following command solves the equations x+3y=5 and -x+4y=9 for the variables x and y. Notice that you have to use double equal signs when solving equations (a single equal sign is an assignment, where a double equal sign is for equations that may or may not be true). There are other commands like FindRoot that do similar calculations and will be useful if there are multiple solutions.

```
Solve[{x + 3 y == 5, -x + 4 y == 9}, {x, y}]
```

Manipulate (and a Sequence)

The Manipulate command can be used in a variety of ways. In the following example, Manipulate is used to display the terms of a sequence $a_n = \left(1 - \frac{4}{n}\right)^n$. If you scroll the blue dot to the right it will display terms for higher values of n. To see what value of n is being displayed, click the + sign to the right of the slider bar. You can then click the plus sign or enter in which ever values you want.

```
a[n_] = (1 - 4 / n) ^n;
Manipulate[N[a[n], 10], {n, 1, 1000}]
```

You could also manipulate two different parameters in one window. The commands below plot j(x) and transformations of j(x). What kind of transformations do the parameters a and b represent? Play around with the values a and b. What happens if a is fixed at 1 and you change b?

```
j[x_] = (x - 1) ^2 (x + 1);
Manipulate[Plot[{j[x], j[x * a] - b}, {x, -3, 3},
  PlotRange → {-2, 2}], {a, 0, 5}, {b, 0, 2}]
```

Tables (usefull for series and sequences)

The Table command is used to create a list of values and can be useful to display various terms of a sequence. The following command creates a table called mytable1 that evaluates $a_n = \left(1 - \frac{4}{n}\right)^n$ (up to 10 decimal places) at several values of n starting with 1 and going to 20 by an interval of 1 (this is what is denoted in the list **{n, 1, 20, 1}**). The command TableForm just puts the values from the table in a list.

```
a[n_] = (1 - 4. / n) ^n;
mytable1 = Table[{n, N[a[n], 10]}, {n, 1, 20, 1}]
Grid[mytable1, Frame → All]
```

Series and Sequences

One can also evaluate sums (or partial sums) using the command Sum, below I have evaluated the sum of the first 30 terms in the series $\sum_{n=1}^{\infty} \frac{3n+5^n}{8^n}$, if you don't want to display it as a fraction either put N[] around the entire expression or change one of the numbers inside to a decimal.

```
Sum[(3 n + 5 ^ n) / (8 ^ n), {n, 1, 30}]
```

```
Sum[(3 n + 5. ^ n) / (8 ^ n), {n, 1, 30}]
```

You can plot a table of pairs against each other by using list plot.

```
ListPlot[mytable1]
```

One can also use the Do command to create a sequence recursively. Below is such an example.

```
a1 = 1;
Do[an+1 = 3. - 1 / an, {n, 1, 10}]
Grid[Table[{k, ak}, {k, 1, 10}], Frame → All]
```

You can also define a function that plots the nth partial sum by using the delayed evaluations. If I wanted the kth partial sum for the series above I could use the following commands. Notice that the delayed evaluation := is used instead of the =. The reason for this is that you don't want to add up things until you know how many things to add up!

```
s[k_] := Sum[(3 n + 5.^n) / (8^n), {n, 1, k}]
```

Then evaluating I could get the sum for however many terms I wish. For instance the 100th partial sum is:

```
s[100]
```

The 12th partial sum is:

```
s[12]
```

One could also do this for power series. Suppose I wanted to think about the power series given by: $\sum_{n=0}^{\infty} \frac{x^n}{n!}$. While I can't add up an infinite number of terms I can add up k. But the value will also depend on x, so our series will be a function of both x and n.

```
s[x_, k_] := Sum[x^n / n!, {n, 0, k}]
```

Now if I want to see the 5th order taylor polynomial I just have to put in k=5 and leave x as x. So $T_5(x)$ would be given by:

```
s[x, 5]
```

If I wanted to evaluate the 4th degree polynomial at x=3 then I would use:

```
s[3, 4]
```

Recall that this series is actually the power series for e^x,below I have plotted e^x in blue along with the 2 nd, 3 rd and 4 th degree Taylor Polynomials in red, yellow and green respectively.

```
Plot[{Exp[x], s[x, 2], s[x, 3], s[x, 4]}, {x, -4, 4}]
```