A DEEPER LOOK AT USING FUNCTIONS IN MATHEMATICA

In "Getting Started with Mathematica" we learned the basics of plotting and doing computations in this platform. In this document, I will delve a little more deeply into some areas, and introduce some new and very useful functions.

Let's begin by recalling some important fundamentals:

• All functions in Mathematica begin with a capital letter, like Cos, Sin, Plot. Some functions have multiple capital letters, like ArcCos, LegendreP (for finding Legendre Polynomials), or SphericalHankelH1.

• All function calls make use of square brackets, [], as in:

\begin{verbatim}
In[1]:= Cos[30 Degree]
Out[1]= 3/2

In[3]:= ArcSin[1]
Out[3]= \pi/2
\end{verbatim}

• Lists use braces, or curly brackets, { }, as in:

\begin{verbatim}
In[4]:= Plot[x, {x, 0, 2}]
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=0.8\textwidth]{plot.png}
\caption{Plot of \( y = x \) from 0 to 2.}
\end{figure}
Notice that we have two sets of lists in the plot directly above, :  the first set enumerates the list of functions to be plotted, the second list delineates the plotting parameters (i.e., plotting with respect to x from a lower limit of 0 to an upper limit of 2)

•We use parentheses to indicate clustering of numbers or variables.

Nested Function Calls and some new Functions

Not only does Mathematica have thousands of possible functions we can use, but the structure of its programming language makes it very easy to nest, or cluster, several functions in one command line (i.e., one line of programming code).

Suppose we want to evaluate the value of the indefinite integral \(\int \frac{x}{1+x^3}dx\) and plot the resulting function between \(x =0\) and \(x=3\). We have a few ways we can proceed. First, we can evaluate the integral:

\[
\text{In}[14]:= \text{Integrate}[x / (1 + x^3), x]
\]

\[
\text{Out}[14]= \frac{\text{ArcTan}\left[\frac{-1 + 2 x}{\sqrt{3}}\right]}{\sqrt{3}} - \frac{1}{3} \log[1 + x] + \frac{1}{6} \log[1 - x + x^2]
\]

The output line shows the function that results from integrating this particular integrand. We can plot this function on the interval \(0 < x < 3\):
This introduces us to the "%" symbol; "%" gives the last result generated output line, (%% gives the penultimate result, %% ... % n times gives the nth previous result). We can also nest several functions into one command as:

\[
\text{In}[16]= \text{Plot}[\text{Evaluate}[\text{Integrate}[x/(1+x^3), x]], \{x, 0, 3\}]
\]

As you use Mathematica more extensively, you will find that you will need to exert great care to ensure that all parentheses, braces and brackets are properly balanced.

Let's explore some of these concepts introducing the Sum function. Suppose we want to explore the properties of Taylor series, and try to convince ourselves that summing polynomials in the proper way can reproduce a wide array of functions. In particular, remember from basic calculus that the Taylor series for \(\cos x\) is:

\[
\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots
\]

Let's plot both \(\cos\) and its Taylor series on the same set of axes and see if they are equal:
In[17]:= Plot[{Cos[x], 1 - x^2/2 + x^4/24}, {x, 0, \[Pi]}]

Out[17]=

Well, they match OK through about x = 1.5, so let's see how many more terms I have to add to get a better fit:

In[18]:= Plot[{Cos[x], 1 - x^2/2 + x^4/4! - x^6/6! + x^8/8!}, {x, 0, \[Pi]}]

Out[18]=

Much better, but we can see that the graphs are beginning to diverge at x > 3, so that if we extend our region of comparison:

In[19]:= Plot[{Cos[x], 1 - x^2/2 + x^4/4! - x^6/6! + x^8/8!}, {x, 0, 2\[Pi]}]
We can always add more terms in our Plot function, but we realize immediately that this can become tedious if we need to consider too many terms.

Enter the Sum function. You should remember from intro calc that the Taylor series for cos can be written in the closed form:

\[
\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}
\]

We can express this in Mathematica as:

\[
\text{Sum}[(\text{-1})^n x^{2n} / (2n)!], \{n, 0, \infty\}
\]

"Sum" followed by brackets tells the program to do the sum of the indicated function, indexing over \(n\), with a lower limit of 0 and an upper limit of infinity. Now, let's exploit the ability of Mathematica to nest functions and explore the convergence properties of the cos series. We plot over the interval \(0 < x < 3\pi\):

\[
\text{Plot}[(\cos x), \text{Sum}[(\text{-1})^n x^{2n} / (2n)!], \{n, 0, 10\}], \{x, 0, 3\pi\}]
\]

The curve for \(\cos x\) is in blue; red shows the curve for the summation. Notice that we use braces to define the list of functions to be plotted, and also braces inside the Sum function describing, in order, the index, the lower limit of summation, and the upper limit of summation. The example above computed the Taylor series for \(\cos\) using the first ten terms of the series; let's see how well the Taylor series approximates \(\cos x\) if we use the first 20 terms of the series:
And we obtain the expected result:

using more terms in the series expansion yields closer approximation to the function.

Our First Brush with Fourier Series and the Manipulate Function

We will spend several weeks investigating the properties and uses of Fourier series in mathematical physics. Fourier series allow us to take a wide array of functions and express them as sums of trig functions (remember that Taylor series allow us to express functions as sums of polynomials). Fourier series are particularly useful in describing piecewise functions such as:

This graph shows a repeating (i.e., periodic) function known as a square wave. Let's investigate whether we really can reproduce this function using a series of sinusoids properly chosen. Later in the course we will learn how to compute the terms for a Fourier series, but now let's just focus on the results of those computations. I will try to approximate this graph via:
(Don't worry about the factor $\frac{4}{\pi}$, I include this just to make sure the function has a max of 1 and min of -1; you will see how to compute these when we begin our formal study of Fourier Series).

Not too bad. Our choice of trig functions reproduces the periodicity of the square wave, and apart from some wiggles in the graph, we see the outline of a square wave taking shape. As before, let’s see what happens if we sum the first 100 terms of this series. Now, we have to be a little careful in finding an expression for the nth term of the series:

$$\text{Plot}\left[\frac{4}{\pi} \sin(\frac{x}{1}) / (2n+1), \{x, 0, 100\}\right]$$

Much better.

Let’s take a look at our expression for the nth term in the series, namely $\sin((2n+1)x)/(2n+1)$. Notice that we want only the odd terms in our series; in order to omit the even terms, we need an expression that will "skip" them. So by using $2n+1$, we get $\sin x/1$ when $n=0$, $\sin 3x/3$ when $n=1$, $\sin 5x/5$ when $n=2$ and so on. We have two other ways we could write this sum; we could start our summation at $n=1$ in which case we would write:
Verify that using \(2n - 1\) and starting from \(n = 1\) is equivalent. Mathematica allows us yet another way to write this sum:

\[
\text{In}[33]:= \quad \text{Plot[}(4/\pi) \text{Sum[Sin[}(2n - 1)x]/(2n - 1), \{n, 1, 201\}], \{x, -3\pi, 3\pi\}] \\
\text{Out}[33]=
\]

Notice that we simply have \(\sin(nx)/n\) in our summation. How can we obtain only the odd terms? Notice the list of indexing parameters: \(\{n, 1, 201, 2\}\). The first three elements \(\{n, 1, 201\}\) retain their familiar meaning (we index over \(n\) starting from \(n = 1\) and going to \(n = 201\)); the new element means that we take every other term in the sequence. In other words, the list \(\{n, 1, 201, 2\}\) means we index over \(n\), starting at \(n = 1\), ending at \(n = 201\) using a step size of 2 (use terms 1, 3, 5, 7 ...). You can read more about the Sum function simply by typing in "sum" into the search line of Mathematica's documentation center.

It is clear that taking the first 100 terms of this series yields a very nice approximation to our original square wave. Suppose now we approach our study of Fourier series from a slightly different angle and ask how close the approximation is after a certain number of terms? We can do this by making use of the Manipulate command. Our nest grows a bit larger with:
Let's examine how this output derives from the input line[34]. In line [34], notice that we have made a subtle change to the list of sum elements in the Sum command. Instead of \{n, 1, 100\}, we now have \{n, 1, terms\}. This means we are treating the upper limit of our sum as a free parameter labeled "terms" whose value can change; if terms = 1, we have only one term in our Fourier series, if terms = 50, we are plotting the first 50 terms, and so on. The final brace in the command, \{terms, 1, 100\} means that Fourier series will be computed and plotted that have anywhere from 1 to 100 terms in the series. Since the .pdf you are reading is not interactive, you will have to type the input line into a live session of Mathematica to see how this works. Notice the slide bar above the graph. Moving the slide bar will allow you change the value of terms in the computation, and the graph will change as you add more and more terms to the Fourier series. Notice also that at the right end of the slide bar is a small, white cross. If you click on that cross, you will get a panel with animation controls. It should not take you long to explore how they work and what they show you.