Overview of Mathematica

Junior Lab Technical Staff
MIT Department of Physics, Cambridge, MA 02139

This quick-start guide contains instructions on how to use Mathematica on Athena to help you in your Junior Lab investigations. No prior knowledge of Mathematica is necessary and only minimal familiarity with Athena is assumed. This document will give you a basic description of just a few algorithms of interest to physicists. It is intended to provide a minimal introduction which will inspire you to reach further in your analysis than you might have without such a tool.

WHAT IS MATHEMATICA?

Mathematica is an calculation and analysis tool written by Wolfram Research Inc. It is very powerful because it has a large lookup table of builtin functions, integrals, derivates and other utilities as well as fitting algorithms and tools. Such tools are usually included in libraries which need to be declared at the beginning of a Mathematica **notebook**. The Mathematica notebook is where commands are written and executed by Mathematica. This notebook can be saved and the evaluated expressions will saved as well as the expressions themselves.

One of the useful aspects of Mathematica is that it is symbolic in that is it manipulates equations and expressions in the same way you write and work them out on paper. It does not turn them into numbers. For example, it can do analytic integrals and derivates. If you ask for it to take the derivate of **sin(x)**, it will give you **cos(x)** as an answer. Same way, it can simplify complicated expressions, do analytic integrals, all symbolically.

STARTING MATHEMATICA

Mathematica, version 4.1 is available to all MIT students through Athena. It is easily accessible in the **math** locker:

```
>> add math}
>> mathematica &
```

This will open a Mathematica notebook and also a **palette** which contains several standard mathematical symbols that can be used in an expression. You can compose your expressions by typing or by clicking the mouse on these options. If you happen to close this palette or need other symbols and standard expressions, you can open it from the **File** menu under **Palettes**.

Each line in your Mathematica script in an expression and can be evaluated, for example, begin your session with the following:

```
2+2 (then press shift-enter/return)

In[1]:= 2:2

Out[1]:= 4
```

If you do not want to see the output of an expression, you can simply put a semi-column at the end and this will suppress the output unless there are errors.

Entering Greek (and other) Characters

It is easy to input special characters from the keyboard. For example, ψ can be written by the typing "ESC-p-s-i-ESC" on the keyboard.

LOADING DATA INTO MATHEMATICA

Mathematica knows about several internally defined structures. You can import these structures into Mathematica. For example, the following can load a table of 3x1024 matrix of data into Mathematica into a variable called rawdata. The sampledata.dat has to be in the same directory as the file opened unless the full path of the file is given.

```
rawdata = Import["sampledata.dat", "table"];
```

The most common structure for data input is the table structure. But it also knows about lines, list or words.

First thing you want to do is to extract from your data, rawdata, the columns of data to which you want to fit a function. This requires some more complication than in Matlab since some function names mean different things in different libraries, you need to add a particular library for Mathmetica to understand which command you want to use. For matrix manipulation, this should be added as a line at the beginning of the file and evaluated.

```
<< LinearAlgebra`MatrixManipulation`
```

The help browser will always say under which library the function is included in. Copying this line is good enough.

In this case, we want to take the second and third columns of data to be our two vectors to be fitted:

```
data = TakeColumns[rawdata, 2, 3];
```

USING THE ROUTINES

Mathematica has several fitting routines available for different applications. For example, the standard Fit, PolynomialFit, Regress and more sophisticated NonlinearRegress.

The fitting routines are contained in different libraries.

PolynomialFit is in <<NumericalMath'PolynomialFit'
Regress is in <<Statistics'LinearRegression'
NonlinearRegress is in <<Statistics'NonlinearFit'

What is common to all these fitting algorithms is that they all take a model which has to be defined a function of variables. The variables while defining a function need to have an **underscore** that comes after them to indicate that they are not constants. Otherwise, pre-defined constants can also be used in defining a function. The function assignment is indicated by the **:=**. Here, we define a functions as the sum of two Gaussians plus a background as such:

FitFunc[x_ , a1_ , a2_ , a3_ , b1_ , b2_ , b3_ , back_] := a1 * Exp[-((x- a2)/a3)^2/2]+ b1 * Exp[-((x - b2)/b3)^2/2] + back

Everytime you define a function or use a predefined function in Mathematica, it is important to use square brackets and a capital letter as the first character. Here, the function Exp is the predefined **exponentiate** function but could have been interpreted as a variable if it was not capitalized or had parenthesis instead of square brackets.

These fitting algorithms also ask for the data in a format where they are ordered pairs. The **data** variable we loaded earlier from a file is in this format. If you want to define it inside Mathematica, the general syntax is :

points = {{0, -1}, {1, 2}, {3, 7}, {4, 6}, {7, 9}, {11, 14}};

Other parameters can be unique to the fitting algorithm and the application of it. Here, we discuss the most useful NonlinearRegress or NonlinearFit here in detail. The general syntax is :

NonlinearRegress[**data**, **model**, **variables**, **parameters**]

It is customary that the variables be associated with some initial guess values. Although this may not be necessary for simple functions, it is a good idea to start off the regression.

z = NonlinearRegress[data, FitFunc[x, a1, a2, a3, b1, b2, b3, back], {x}, {{a1, 38}, {a2, 624}, {a3, 6.7}, {b1, 82}, {b2, 640}, {b3, 5}, {back, 86}}, RegressionReport $->$ {BestFitParameters, ParameterCITable}, MaxIterations $->$ 50]

There are several options under RegressionReport which will suit your needs. You can look this up in the Mathematica user's guide.

{A1, A2, A3, B1, B2, B3, BACK} = {a1, a2, a3, b1, b2, b3, back} /. BestFitParameters /. z;

The confidence intervals are calculated separately for a lower and an upper bound which means that you can write your final answers like $10.2^{+.4}_{-.6}$

PLOTTING QUICKSTART

Each command in Mathematica comes with a bunch of options. You can at any time see these by Options. For example,

Options[Plot]

In general, Plot takes to arguments, the function and the an ordered set of the variable, the minimum and the maximum value of it. For example,

Plot[FitFunc[x, A1, A2, A3, B1, B2, B3, BACK], {x, 600, 700}]

However, usually this is not enough. We want to see the data and the fit on top of each other.

p1 = ListPlot[data, DisplayFunction -¿ Identity, PlotRange -¿ All, PlotStyle -¿ {RGBColor[0, 0, 1]}]; p2 = Plot[FitFunc[x, A1, A2, A3, B1, B2, B3, BACK], {x, 600, 700}, DisplayFunction -¿ Identity, PlotRange -¿ All, PlotStyle -¿ {RGBColor[1, 0, 0]}]

Here, the two commends do not plot anything since DisplayFunction is assigned to Identity which means it will not really be plotted. They assign the graphs to some variables which we will make visible or 'show' by the following command:

Show[p2, p1, Axes -¿ None, Frame -¿ True, FrameLabel -¿ {Channels, Counts}, RotateLabel -¿ False, TextStyle -¿ {FontFamily -¿ "Times", FontSize -¿ 12}, DisplayFunction -¿ $DisplayFunction]

Now, this generated a nice graph which can be incorporated into a paper or a presentation by exporting it to postscript from the Menu.

Getting Help in Mathematica

To learn more about any Mathematica commands, you can go to the **Help** menu and bring up the **Help Browser**. Typing a simple English word describing what you want to do, in the box Go to: brings up a choice of related items usually. Try this for example with Import.

Sample Notebooks available in /8.13/mathematica

1. gauss3.nb demonstrates non-linear curve fitting and plotting
2. mean-path-calc.nb demonstrates numerical integration

This document was originally written Bilge Demirköz. The Junior Lab Technical staff greatly appreciate her help in bringing Mathematical capabilities to Junior Lab.