FreeMat: Why are we doing this and how do I get started?

Bruce Emerson*
COCC Science Department
(Dated: September 16, 2013)

This short (hopefully) document will attempt to provide a rationale for compelling students in my physics classes to learn how to use FreeMat (a MATLAB clone) to present and analyze data and generate appropriate graphics for use in their lab reports and other documents. Like learning to use LaTeX this is another skill set that is widely, though not universally, used across the scientific and engineering disciplines. Additionally this document will provide basic guidance for installing the software on your personal computer and generating some basic outputs relevant to labs.

1. WHY?

This is, of course, the fundamental question. Why might you see it as valid to spend the effort, and it will take effort, to learn how to use a math calculating engine like FreeMat? On a philosophical level I would suggest that your are in a physics class to engage with the process of figuring out how the universe works. Physics, and all the sciences, are engaged in the effort to 'look under the hood' of Mother Nature and try understand how it works. Even though the results of this effort are cool to know about, and important to understand, it is the relationship between the smaller 'parts' and how their interactions lead to the world we observe around us that is the essence of our study.

In the same sense it can become important to understand how to generate and present a plot of your data without the intervention of some other programmer's perceptions of data presentation. This is my core frustration with Excel and other similar programs. Such software is very convenient but in my professional life and work I have very often felt the need to generate a different form of output. This has been very difficult to implement in the standard software. To have better control of the analysis and presentation of data we need to understand the nuts and bolts of the process better and FreeMat (and all it's cousins - MATLAB, MAPLE, Mathematica, and the rest) allows, and requires us, to do so.

On a more pragmatic level what I observe is that a large fraction of my students end up crossing paths with something like FreeMat later on in their school careers. At OSU MATLAB is a standard tool in a number of departments. It used to be MAPLE and in other schools it is Mathematica or another tool. All of these math engines share common features so knowing how to use one helps with using others. At their core all of the math engines use command structures which are closely related to programming languages which are common across the disciplines. Knowing a programming language helps with FreeMat coding and FreeMat coding helps with programming. In Electrical Fundamentals there is a simulation program called SPICE which shares many coding features with all the math engines.

Think of it this way: If you open up the hood of your car and learn how the various systems look and interact you then understand better how a car functions. While it is true that different manufacturers use different specific parts providers you still understand a great deal about how a Ford truck works if you know how your Toyota Camry works. The similarities are greater than the differences and your skills in maintaining and troubleshooting a Toyota are valuable and relevant to any car you may own.

2. GETTING FREEMAT INSTALLED

One of the primary concerns for me is that software I ask you to use be easily available (cheap is good) and that it operates effectively the same across platforms. FreeMat is available at `http://freemat.sourceforge.net/` [?] and there should be a version that works on your system. I personally end up working on both Macs and Windows machines (one at home and one in the office) and I find very few differences. Admittedly, when they do crop up, they are aggravating.

For ease of use I would encourage you to install FreeMat on your personal computer if you can and carry your files around on a flash drive so you can share them or modify them as necessary.

Of course there is still the challenge of learning how to use FreeMat. I will try to help jump start you up the learning curve by providing sample files that perform various useful tasks. Not surprisingly there is also a very nice FreeMat user manual [1] which I would recommend keeping handy. If you find other resources I should link here please share them with me.

FreeMat is installed on all the lab machines in the science department (supposedly) so even if you don't have access to a machine at home you can work at school. At this time FreeMat is not installed on machines in the computer labs - sorry.

*Electronic address: `bemerson@cocc.edu`; URL: `http://coccweb.cocc.edu/bemerson/public_html/physics/ZPhysicsGrp/physics.html`

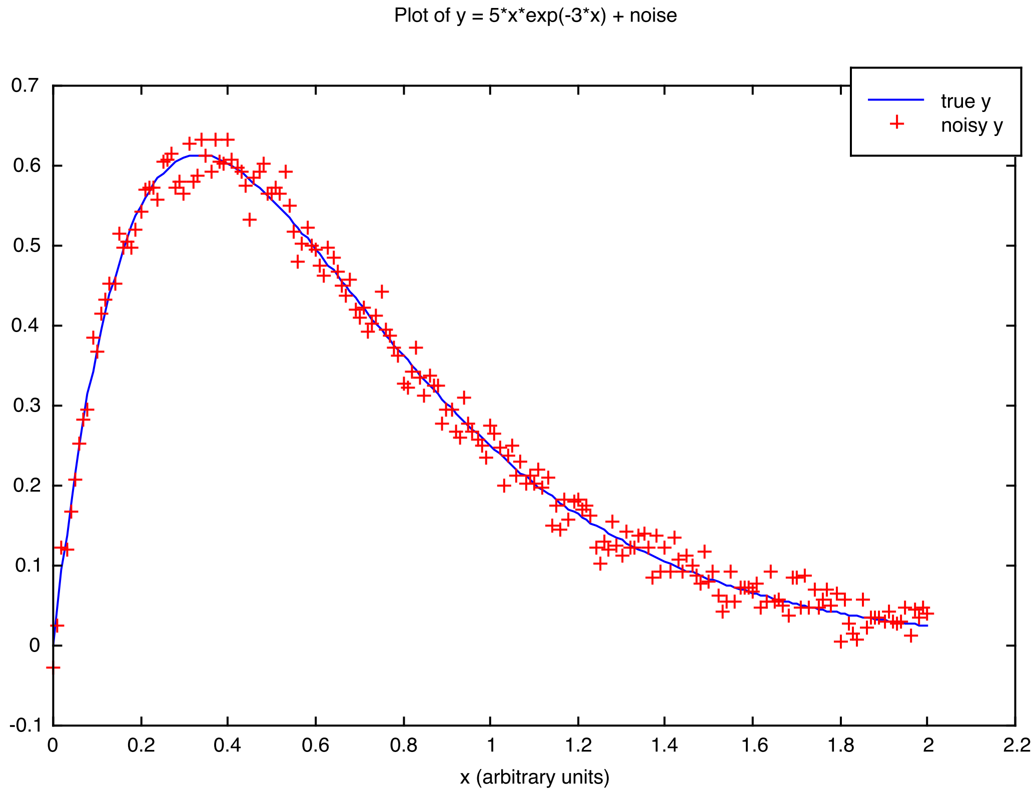

FIG. 1: This is what your sample output might look like. There are two functions plotted here. One is an exponential as a line. The other is the same exponential with random noise and plotted as a set of data points

3. BASIC TASKS

For now there are two basic tasks I'd like to be sure that you know how to execute in FreeMat. The first of these is how to create a plot of a particular function and then export it as a pdf file that you can include in your LaTeXdocuments. The second is a straightforward way to present a plot of your data along with a function that represents your physics model of the relationship between the variables.

Initially when you open FreeMat you will get a command window. You can type various mathematical operations into the command window and FreeMat will execute them and tell you the answer. To create files that execute more complex tasks you will want to open an Edit window (third icon from the left at the top on Macs) in which you will create your various applications.

3.1. Creating a Plot

You can open this FreeMat file *FunctionPlot.m* from the file I might provide you or you can enter the following commands into the FreeMat editor yourself.

```
%  File:  FunctionPlot.m
%
%  Generate a function and add noise to it.
% Plot the function as a solid line and the
% noisy data as open circles

x = 0:0.01:2;
          %  generate the x-vector
y = 5*x.*exp(-3*x);
          %  and the "true" function, y
yn = y + 0.02*randn(size(x));
          %  Create a noisy version of y
plot(x,y,'-',x,yn,'ro');
          %  Plot the true and the noisy
xlabel('x (arbitrary units)');
          %  add axis labels and plot title
ylabel('y (arbitrary units)');
title('Plot of y = 5*x*exp(-3*x) + noise');
legend('true y','noisy y');
```

When you execute this file (assuming all goes well and there are no tiny syntax errors) you will see the output shown in Figure 1.

Once you have generated the FreeMat output you are still in FreeMat which is fun but not very useful for lab reports or other documents. If you save the plot the dialog box will give you the option of saving the plot

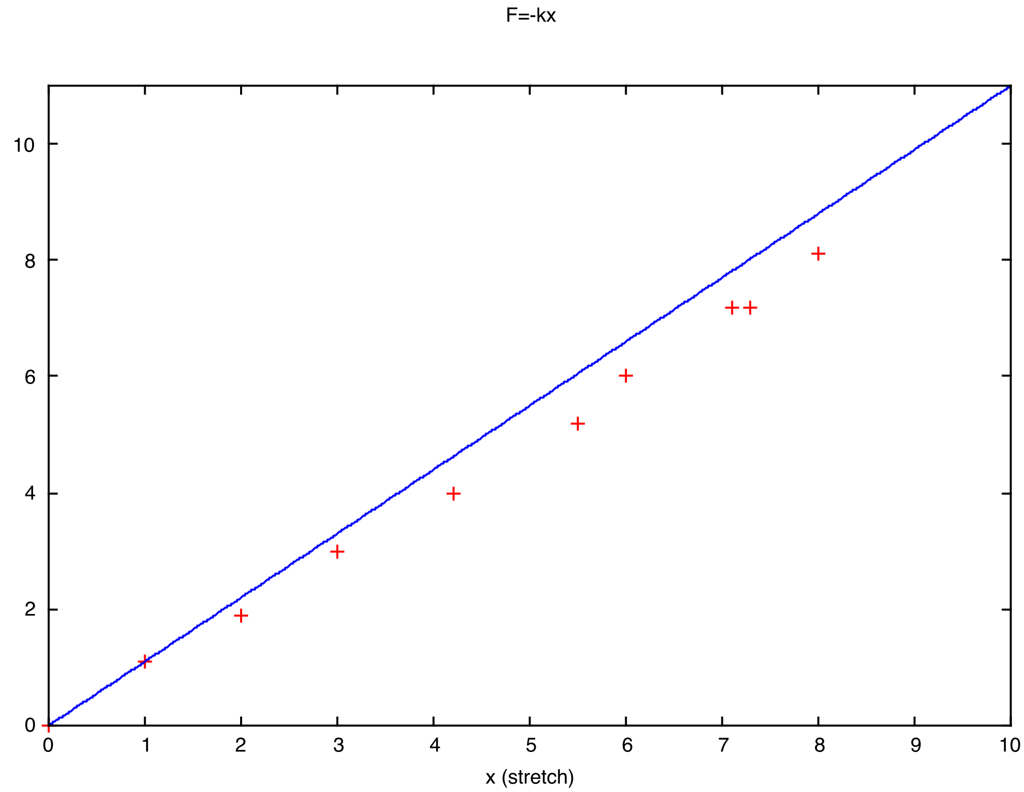

FIG. 2: This is what your sample output might look like. This time there are data points plotted as + signs and a linear physics model on top of the data.

as a pdf file. This is strongly recommended due to the scalability of pdf files.

If you wish to include your actual FreeMat code in your lab report you will need to use the `\begin{verbatim}` command which tells LaTeX to not interpret what you write as potential commands. Like most LaTeX environments you close with `\end{verbatim}` which is how I included the FreeMat code previously. There is lots of potential for confusion and frustration (I've been trying to make this work the way I want for a couple of hours now so you get the picture).

3.2. Plotting your Data

You can open this FreeMat file *ScatterPlot.m* from the file I might provide you or you can enter the following commands into the FreeMat editor yourself.

```
%  File:  ScatterPlot.m
%
% A sample FreeMat script to plot
% some data and label the plot

x = [0,1,2,3,4.2,5.5,6,7.1,7.3,8] ;
                 %  x vector filled with x data
y = [0,1.1,1.9,3,4,5.2,6,7.2,7.2,8.1];
                 %  vector of y values
 % disp(x,y);
                 %  this would allow me to
                  % display my data values
                  % to check them
xmodel=0:0.01:10;
                  %  this creates a range of
                  % x values for the model
ymodel=1.1*xmodel;
                  %  this creates y values
                  % for the model
plot(x,y,'r+',xmodel,ymodel,'b-');
                   %  create the plot with both
axis([0,10,0,11]);
                   %  set limits on axes
xlabel('x (stretch)');
                   %  label the x-axis
ylabel('Spring Force (N)');
                   %  label the y-axis
title('F=-kx');
                   %  label plot
```

When you execute this file (assuming all goes well and there are no tiny syntax errors) you will see the output shown in Figure 2.

In the process of doing this I learned that in LaTeX I need to define labels for figures after the `\caption{...}` statement or LaTeX can get confused with it's numbers. Useful discovery.

[1] FreeMat 4.0 User Guide, Gary Schafer and Timothy Cyders `http://www.floss4science.com/new-freemat-4-user-guide/`.

[2] FreeMat source code for all forms and operating systems `http://freemat.sourceforge.net/`.

[3] Webpage for Timothy Cyders who co-wrote the manual and uses FreeMat for all sorts of things `http://www.ohio.edu/people/tc285202/freemat.html`.

[4] Site2241 is apparently the blog home of the original author, Gary Schafer, of the FreeMat manual. Makes for some interesting reading.... `http://www.site2241.net/index.htm`.

Acknowledgments

As usual I need to thank all those in the open source/Creative Commons universe who are willing to share their knowledge and expertise with everyone. Thanks also to my students who make it worth trying to get better as a teacher/learning facilitator:) [1]