

Notes on plotting data and theory in Mathematica

Introductory comments

Always try to compare your data to a possible theory. Making such comparisons is often nicely done via plotting of the data and the theory.

It's often the case that you take some data, say on amplitude and phase of some signal, and that you might need to plot the data. *Mathematica* has lots of tools to help. Here are some examples of plotting in Mathematica, to help you generate easier to understand plots of your data and theoretical expectations.

You can get additional inspiration from the *Mathematica* Activity 1 and Activity 2.

General important comments on using Mathematica

A few very general rules :

1) NEVER name your functions, data lists, or any other thing that you define in Mathematica, EVER EVER, with names that start with Capital Letters. Ever. All built-in functions start with capital letters. If you use them, then some day you will end up redefining some internal function, and you will be very confused by the behavior of *Mathematica* thereafter. It's surprisingly easy to say something like, "Sin = 0.5" and thereby redefine the Sin function. Don't start names with capital letters.

2) *Mathematica* tries to help you by using colors to tell you about things. For example, when something is presently not defined it appears in blue font. Use that fact, and you'll start to understand why some of your efforts are not working.

3) Often you will define some function, and try to plot it, but all you get is an empty graph. What's wrong?? Many times, you can see what's wrong if you numerically evaluate the function at a single point. If all the necessary items have been defined, you are going to get a number. If not, *Mathematica* prints out the best it can do, and you can see what items are yet to be defined.

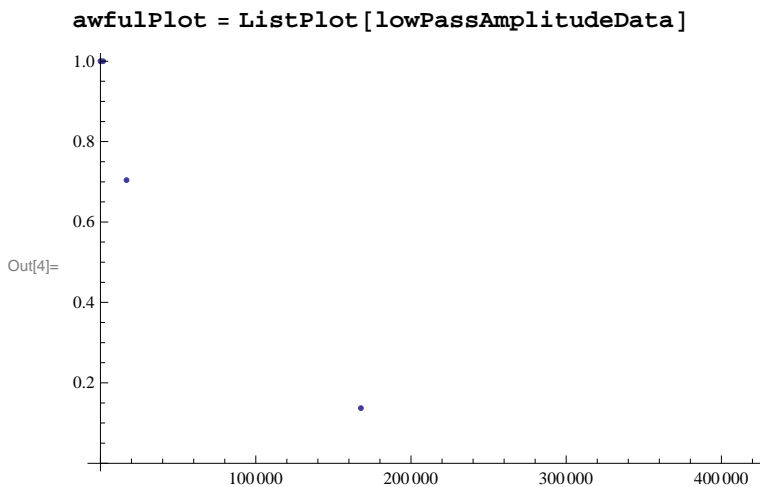
Some low - pass data to plot

Let's start with some data on the Vout/Vin for a low - pass filter. Here is a sample set of data from one of your colleagues, from their Lab 3 report:

```
In[3]:= lowPassAmplitudeData = {{16.770, 1.00}, {167.7, 1.00},
    {1677, 1.00}, {16770, 0.704}, {167770, 0.137}, {1677000, 0.059}}
Out[3]:= {{16.77, 1.}, {167.7, 1.}, {1677, 1.},
    {16770, 0.704}, {167770, 0.137}, {1677000, 0.059}}
```

Notice a couple of things about this data set that are excellent: First, notice that the data set was taken at a variety of frequencies and that the frequencies are separated by factors of ten. That fact indicates that the folks taking the data were thinking about logarithmic scales. Second, notice that the data includes points that are well outside the 'interesting' region. In this case, 'interesting' means the region where the low pass filter begins to attenuate the input signal. This data set has points that are well inside and well outside the region of the knee frequency, so you have a chance to see whether the overall behavior follows expectations.

OK, so here is an example of plotting the data :



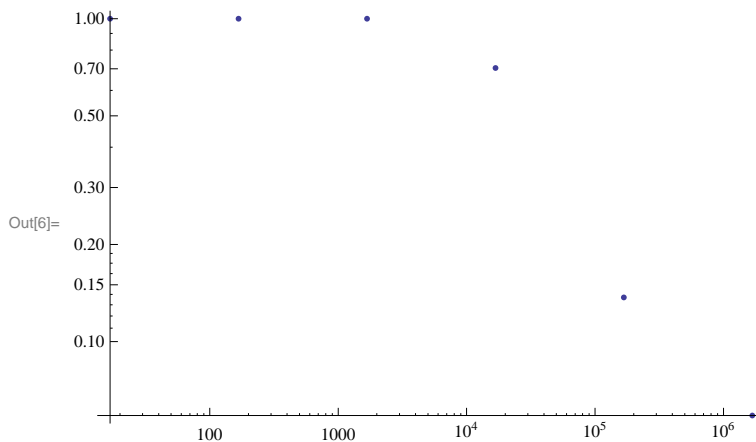
The plot of data points shown above is named 'awfulPlot' for a good reason. It's awful. So awful, that it's hard to even start..., but here are a few problems:

- 1) No axis labels (what is being plotted here??)
- 2) Fonts are too small for a human to read without technology.
- 3) Where are all the data points? *Mathematica* has cut off the point at 1.677 MHz in an effort to help you see the data points at much lower frequency. Generally, *Mathematica* makes choices about auto-scaling that are not always helpful.
- 4) The data points themselves are too small to be easily visible.
- 5) With this choice of linear axes, many of the data points are crammed into the left end of the plot, so you can't see them.

Spread the data out using LogLog axes

Here's an improvement that lets you see all the data more clearly by using a set of logarithmic axes:

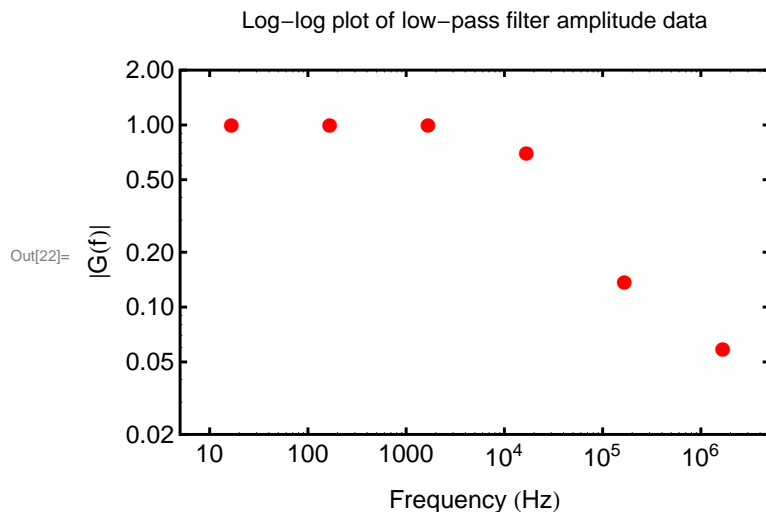
```
In[6]:= awfulListLogLogPlot = ListLogLogPlot[lowPassAmplitudeData]
```



All the other complaints still hold. There are no labels and everything is too small.

Here is a version that solves many of these problems:

```
In[22]:= lessAwfulListLogLogPlot = ListLogLogPlot[lowPassAmplitudeData,
  Frame → True, PlotLabel → "Log-log plot of low-pass filter amplitude data",
  PlotRange → {{5, 5 000 000}, {0.02, 2}},
  FrameLabel → {"Frequency (Hz)", "|G(f)|"},
  LabelStyle → {FontFamily → "Arial", FontSize → 13},
  FrameStyle → Thickness[0.005], PlotStyle → {Red, Thickness[0.01]},
  PlotMarkers → {Automatic, Medium}]
```



Now you can see the data points, the plot is labeled so you know immediately what is being plotted, and everything is more readable. There are many esthetic choices you can make beyond these, but at least the plot is more easily understood.

Plotting theory

OK, so we can apply the same approach to our plots of theory. For example, the low - pass amplitude is predicted by theory to be given by :

$$\text{In[16]:= lowPassAmplitudeTheory}[f_, \text{res}_, \text{cap}_] := \frac{1}{\sqrt{1 + (2 \pi f \text{res cap})^2}}$$

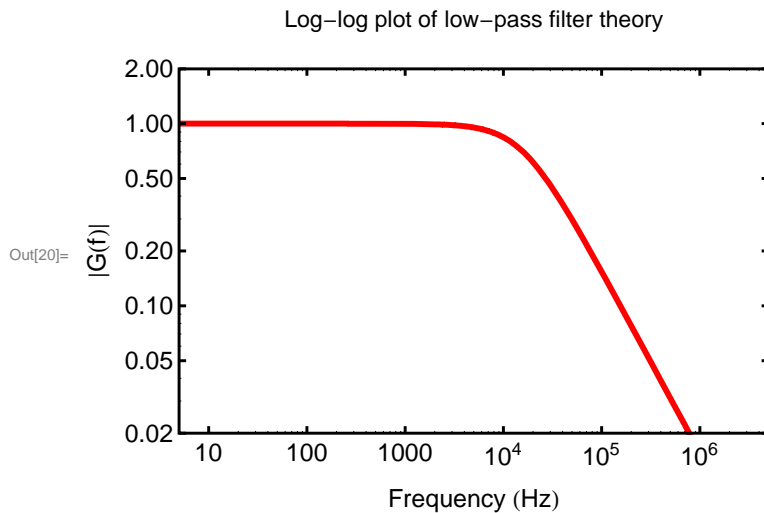
Let's test the function to be sure that it evaluates. The data was taken on a low-pass filter constructed from a 4.67kOhm resistor and a 2.19 nF capacitor. That pair of components should give us a 3dB point around 16kHz, so let's see if we get something reasonable from the function :

In[17]:= `lowPassAmplitudeTheory[16 000, 4670, 2.19 * 10^-9]`

Out[17]= 0.697222

The function appears to work, so let's try plotting it in a less awful way :

In[20]:= `lessAwfulTheoryLogLogPlot =
LogLogPlot[lowPassAmplitudeTheory[f, 4670, 2.19 * 10^-9], {f, 5, 2 000 000},
Frame -> True, PlotLabel -> "Log-log plot of low-pass filter theory",
PlotRange -> {{5, 5 000 000}, {0.02, 2}},
FrameLabel -> {"Frequency (Hz)", "|G(f)|"},
LabelStyle -> {FontFamily -> "Arial", FontSize -> 13},
FrameStyle -> Thickness[0.005], PlotStyle -> {Red, Thickness[0.01]},
PlotStyle -> {Red, Thickness[0.01]}`



Looks pretty good.

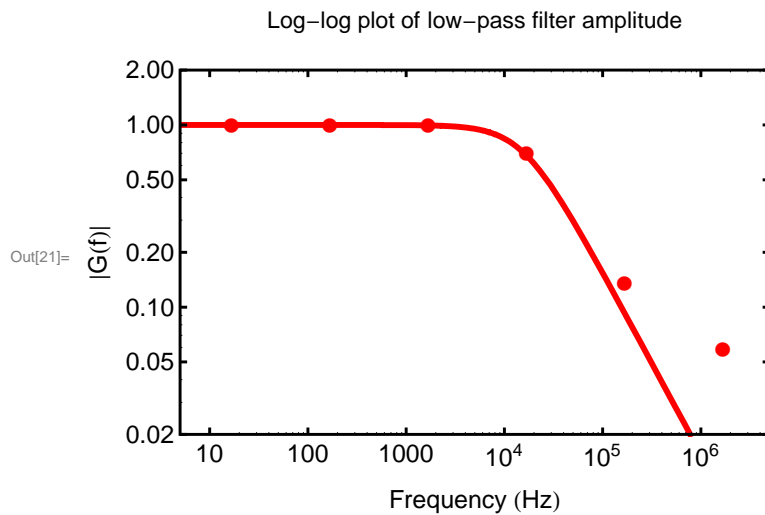
Compare your data to the theory

It's always useful to compare the data that you took to the theory (if any!) that you might have in mind for explaining it. Doing the comparison in real - time, while you are taking the data, is often an excellent idea : If you compare the data to the theory as you go, it makes it much more interesting (kinda like

betting on a horse race!) and much more likely that you will notice when your measurements and theoretical expectations are beginning to deviate.

For example, here's a comparison of the nice data above to the nice theory :

```
In[21]:= Show[lessAwfulListLogLogPlot, lessAwfulTheoryLogLogPlot]
```



See how nicely things were going, but then ... something interesting is happening at the higher frequencies. Wonder why the signal seems to be less attenuated than we expected. Could it be a non-ideality in the filter? Could it be that we were reading some 'amplitude' from the scope screen, that was larger than the actual sine wave, say due to noise on the sine wave? Could it be ... ?? It's in this type of comparison that you begin to confront in a serious way, whether you do or do not actually understand what's going on!

Notice also that a simple table or list of the data fails pretty completely to tell us whether something interesting is happening. Plots are great for turning loose our built-in image processing systems.