Making plots in Matlab look nice

1 Introduction

By default, Matlab plots look rather boring. Go ahead, try this one.

```
x = linspace(-1,1,50);
y = x.^2;
plot(x,y);
xlabel('This is my x label');
ylabel('This is my y label');
title('Really boring plot :-\');
```

Yawn The biggest problem is...most everything is too small. And the default blue gets a bit old after a while.

2 Colors and Line Thickness

Sometimes you'll want a color that's not one of the defaults. It's very easy to use any RGB color code to create any color you want. you'll want to use the `’Color’` option in your plot statement. Right after color, include a vector with the RGB color values of whatever color you want. For example `plot(x,y,’Color’,1/255*[148 0 211])` will produce a very nice shade of purple. You can easily find RGB color codes for various colors online. One thing to remember, Matlab accepts RGB values between 0 and 1, the codes you'll find online go from 0 to 255, so you'll want to scale everything you find by 1/255.

While purple is very nice, the line is still a bit thin. the `’LineWidth’` option will allow you to change the thickness of the plot. 1 is default, in this case 2 looks pretty good. `plot(x,y,’Color’,1/255*[148 0 211],’LineWidth’,2)`

3 Axis Font

The defualt axis font size is too small. We're going to use `set` to do this. `set` is a built in function that sets properties for graphics objects. The code we want is `set(gca,’FontSize’,14)`. You can put this code immediately after the plot statement, or if you used `figure`, right after that. 14 looks good here, but you can play with it to get things looking just right for whatever plot you're making. `gca` means get current axis, which `set` then uses to increase the font size. Notice how all of the font on the plot changed to 14 pt, not just the axes. That's OK, we're going to change the rest next.

4 Axis Labels and Title

The axis labels should be a bit larger than the axis font, and the title should be a bit larger than the axis labels. For each of these, use the `’FontSize’` option for xlabel/ylabel and title. For example, replace your `xlabel` command with `xlabel(’This is my x label’,’FontSize’,16)`. Again, I thought 16 looked good here, but play around with it.

The title works the same way, only it should probably be a little larger. Something like: `title(’Much Better!!  :-D’,’FontSize’,18)`

After all that fiddling, our code isn't that much more complicated...

```
x = linspace(-1,1,50);
y = x.^2;
plot(x,y,'Color',1/255*[148 0 211],'LineWidth',2);
set(gca,'FontSize',14);
xlabel('This is my x label','FontSize',16);
ylabel('This is my y label','FontSize',16);
title('Much Better!! :-D','FontSize',18);
```

And the results are much more pleasing!

5 Using LaTeX symbols

Latex (pronounced ”lay tech,” not like the material) is a typesetting program commonly used in producing math and science documents. Latex symbols are very easy to use in matlab. Say you want to have something like $\sin(\theta)$ in your plot title. All you'd have to do is type in `title(’sin(\theta)’)` and you'd get a nice little θ symbol! This works in the `xlabel` and `ylabel` commands as well:

```
x = linspace(0,2*pi,100);
y = sin(x);
plot(x,y,'Color',1/255*[0 205 0],'LineWidth',2);
axis([0 2*pi -1 1]);
title('sin(\theta)','FontSize',18);
xlabel('\theta','FontSize',16);
```

Latex greek letters are all the same...\ then whatever it is. So, π is `\pi`, ξ is `\xi`. If you wanted $\cos(\omega t + \delta)$, just type `cos(\omega t + \delta)`. If you want capital letter, just capitalize it! So, Π is `\Pi`. Easy as π!

6 Legeneds

If you want to display multiple plots on the same set of axes, sometimes it is nice to use a legend to distinguish your plots. Including a legend is as simple as typing `legend(’plotName1’,’plotName2’,...)`.

7 Saving Figures

When saving figures, NEVER use jpg. They look terrible—pixelated with lots of compression artifacts. png files are better: they don't have any of the compression artifacts, but they can still be pixelated. Scale your figures to be large enough so that when you access the saved png, you don't see any pixelation.

You can tell matlab to save figures for you within a script. The command is `saveas(gcf, 'filename.ext', 'format')`. So to save the figure we generated above, you might type: `saveas(gcf, 'sinPlot.png', 'png')`. This can be much more convenient than going in and trying to save figures by hand (which requires a lot of clicks, and can be a pain if you are saving multiple figures).

8 Homework # 9

Create plots of the following functions, labeling axes and titling appropriately. Save each figure as a png and copy into a single document to submit to D2L along with a copy of your code.

- $f_n(\theta) = \sin(n\theta\pi)$ over the interval $\theta \in [0, 1]$ for $n = 1, 2, 4$. Plot all on the same graph with thick lines of different colors. Use a legend.
- $\Omega(x) = (x + 2)(x - 1)(x - 4)$ over the interval $x \in [-4, 5]$. Plot a red line along the x-axis and plot crosses at the roots of Ω.