Using IDL to Generate Publication-Quality Plots
(27 September)

Purpose: To tie in what we learned about IDL with what you've been learning in the first project, and to give you something practical to do with IDL. Learning IDL is best done informally as the need arises....

Assignment to be completed tonight in lab:

- Write an IDL procedure to generate a publication-quality, postscript plot of your color-magnitude diagram with a zero-age main sequence overlay (and possibly theoretical evolution curves) from the first lab. Design this as a main level procedure, and make sure all the necessary variables are saved in an "xdr" file (save,file='mystuff.xdr',/var/xdr) after you've loaded up all the values.
- In this procedure, first plot out the ZAMS as a line and your color-magnitude data as individual points. Have the program prompt you for a distance, apply the distance modulus and replot the data to do "spectroscopic parallax". You'll probably want to do this interactively in a loop until you get the "best fit" absolute magnitudes for your cluster.
- Write and call an IDL subroutine from your main-level procedure to calculate age given the cutoff (B-V), and print this value on your plot.
- Turn in hardcopies of your programs and your plots.

Some things you'll need to know about IDL plotting:

- The best way to get data from excel to IDL is to first write it into a simple ascii file with just the columns of #'s you want to read into IDL vectors. In idl, there's a cute library program called "io". For example, if you have an ascii file with 2 columns of #'s, which you want to put into idl vectors x and y, you just type io,'filename.txt',0,x,y The 0 tells the program to figure out how big to make the vectors and how long the file is. If you already know this (e.g. by typeing the unix wordcount –wc- command, then you can enter it directly.
- The basic command is plot,x,y
- There are a plethora of optional keywords that can be specified to change the plot appearance. All of these keywords have a corresponding system variable that can do the same thing. For example, to plot each data point as an asterisk instead of connecting the data with a line, you could either set !psym=2 and then make your plot, or you could enter plot,x,y,psym=2
- To add to a plot, use the oplot,x,y command (without any keywords)

- Compare the different possibilities for psym, linetype, thickness
- !fancy=5 or 6 is good for publication quality; don't use less than !fancy=4
- By default, IDL rounds the axis values. If you want it to use exactly the max and min specified, use !type=12. Look at the other things you can do by setting !type to different values.
- Some other useful keywords/system variables: xtitle, ytitle, xrange, yrange, xticks, yticks. All of these can also be changed in the !X and !Y system variables (which are "structures"; e.g. !X.range=[0,100]).
- You might have to look into "vector drawn fonts" to make your axis labels look right. If you want to annotate a plot (keep this to a minimum), you can use xyouts.
- Let's stick with black & white plots for tonight. But it is possible to change the color table with loadct and then add a color keyword to the plot (e.g, plot,x,y,color=100). Color ranges from 0 to 255, but what comes out depends on which color table is loaded.
- Once you have the plot looking right on your x-windows display, you need to replot everything to the postscript device. The procdure for this is...
    - set_plot,'ps'
    - device,file='outputname.ps',/portrait
    - plot,x,y (or .run your plot program)
    - device,/close
    - set_plot,'x'
- If you write a clever program, you can include all of this inside the program (e.g. check device and plot accordingly or prompt for where the plot should be sent)