ASTR4004/ASTR8004
Astronomical Computing
Lecture 06

Christoph Federrath

22 August 2017

Plotting, IDL

1 Extracting data from existing plots/graphs

1. Last time we learned about plotting data from text files and/or functions in gnuplot. Here we now learn how to grab data from existing plots to digitize them in order to make your own plot of some published data from a graph. This can be very useful, because you might be working on a research project where you produce data that you want to directly compare with other existing data from the literature in the same plot frame. Instead of having to read the data from the plot by hand/eye and making a table by hand, there are nice tools that can make this process of extracting data from an existing plot much easier for you. Here we will focus on WebPlotDigitizer, which is such a tool.

2. First go to `http://arohatgi.info/WebPlotDigitizer/` and launch the App.

3. Go through the tutorial, which already has an example image of a plot loaded by default. You can extract data points manually or automatically (selected by colour and masks).

4. The first step is to calibrate the axes of the plot and then you are ready to extract data points.

5. Finally, the extracted data can be formatted as you wish and the extracted data pairs can be copied into a text file, which you can then use for further processing, e.g., for plotting with your own style and together with other data (e.g., from your own work), e.g., in gnuplot or IDL or python, etc.

6. Try it by uploading an image of a different plot/graph, calibrate the axes and see how you can extract data in manual and automatic mode.

2 The Interactive Data Language (IDL)

2.1 Getting started

1. Login to `misfit` and make a new directory `IDL/` in your home dir:
   `> mkdir IDL`

2. Download the IDL startup package prepared for you: http://www.mso.anu.edu.au/~chfeder/teaching/astr_4004_8004/material/IDL_startup_package.tar.gz and copy it to misfit into the new IDL/ directory.

3. Unpack the tarball. This will create subdirectories ASTROLIB/, MPFIT/, and textoidl/, as well as three files: idlstartup, setcolors.pro, constants.pro.

4. ASTROLIB is a useful astronomy IDL library, MPFIT is an IDL non-linear fitting package, and textoidl is a Latex-to-IDL string conversion library that lets you use Latex syntax in IDL to make Greek letters, sub- and super-scripts and special symbols like you are used to in Latex.

5. The idlstartup file is important to control the way IDL starts up. In our case, it defines paths and automatically runs the script constants.pro, which defines useful constants.

6. Add this line to your .bashrc:
   `export IDL_STARTUP=${HOME}/IDL/idlstartup`
   This will make sure that idlstartup is executed everytime you start IDL.

2.2 Simple IDL tasks

1. Now that we have set up the IDL environment, we can start IDL:
   `> idl`
   As for gnuplot, this will lead you to the IDL command line from which IDL is controlled.

2. First, lets make a simple calculation and print the result to the screen:
   `idl> print, 1+1`

3. We can also directly define variables, modify them and print their content:
   ```
   idl> a = 1+1
   idl> a = a*3
   idl> print, a
   ```

4. Finally, lets do some astro calculation involving units. Now that the constants.pro script is already loaded every time you start up IDL, we can use the constants defined in there. For example, if we wanted to print the mass of the sun in CGS units or Newton's gravitational constant, we'd simply type:
   ```
   idl> print, m_sol
   idl> print, g_n
   ```

5. Those can then be combined to say calculate the freefall time ($t_{\rm ff}$) of a typical star-forming core with a density of $\rho = 4 \times 10^{-19}\,{\rm g\,cm^{-3}}$ (which corresponds to a gas number density of about 10^5 particles per cubic centimetre):
   ```
   idl> rho = 4d-19
   idl> t_ff = sqrt ( 3.0*!pi/(32.*g_n*rho) )
   idl> print, t_ff / (1d5*year)
   ```