ASTR4004/ASTR8004
Astronomical Computing
Lecture 09

Brent Groves
21 September 2017

Using IDL for Astronomy Analysis

1) Intro to IDL v2.0

You've already learnt about the basics of IDL, but there are some powerful tools in the IDL toolbox. IDL is amazingly powerful when it comes to vector & matrix manipulation (as it is specifically made as a "Interactive Data Language". It has a fortran & C- heritage (so if you program in these its familiar). As its so powerful for data analysis, it has a strong history with astronomy (which makes it further useful).

You've already obtained the astrolib that we'll be using. This is one of the reasons IDL is so useful for astronomy. However there are several tools out there that make IDL so useful for us.

You should definitely look at "Coyote's guide to IDL", and also download the associated "Coyote Library" in same IDL directory you have the astrolib:

http://www.idlcoyote.com/documents/programs.php

This is an immensely useful library for analysis and displaying the best pictures, and the guide is very useful for getting the best out of IDL. In particular you should look at JD Smith's (an Astronomer at Toledo) guide to HISTOGRAM:

http://www.idlcoyote.com/tips/histogram_tutorial.html

This guide is both helpful and displays why IDL is so powerful.

Another place to be looking is Craig Markwadts curve fitting guide: https://www.physics.wisc.edu/~craigm/idl/fitting.html

This library is also commonly used in astronomy and will be explored in today's lecture.

2) Loading Astronomical images and data cubes (fits files)

First, download some fits data, in particular a cube from the SAMI galaxy survey:

https://datacentral.aao.gov.au/asvo/surveys/sami/

in particular lets go for the red cube of a star forming galaxy (227970):
https://datacentral.aao.gov.au/asvo/sov/sami/227970/spectral_cube-red?format=fits

and place this in our working directory.

Look at the fits file first:
FITS_INFO,'227970_red.fits'

Notice the 5 extensions (images/data) in the fits file

Lets load the primary (extension=0) including the header

data=mrdfits('227970_red.fits', 0,datahd)

the data will be a 50 x 50 image with each spatial pixel ('spaxel') having a spectrum with 2048 wavelength steps.

Look at the header: print, datahd

Based on this you can create a wavelength vector using the IDL routine FINDGEN

Then lets plot the central spectrum: PLOT, wavelength, data[24,24,*]

That big feature in the middle is Hα emission line!

3) Creating fits and images

Lets look at a Halpha image – zoom in on Halpha: PLOT, wavelength, data[24,24,*], xrange=[6850,6950]

Lets find the array indices that correspond to the Halpha range: indices=WHERE(wavelength GE 6910 AND wavelength LE 6930)

Now make an Halpha image: Halpha=TOTAL(data[*,*,indices],3)/n_elements(indices) (The latter is to maintain units).

To look at this we can use the program ATV,Halpha

Now save this. We can even add a header to make this a proper image.

Newheader=dataheader

SXADDPAR,newheader,"NAXIS",2

SXDELPAR,newheader,"NAXIS3" (and delete other 3rd axis parameters

And now save the file: writefits,"227970_Halpha.fits",newheader

4) Fitting Gaussians to data using MPFIT

Now to fit a Gaussian to Halpha. For this we use MPFIT (in particular MPFITFUN) and the astrolib function GAUSSIAN

First we need the errors on the data (in an extension of the fits data):

variance= mrdfits('227970_red.fits',1,varhd)

Error=SQRT(variance)

Then we need to subtract off the continuum. Lets estimate it as a blueward part:

Continuum=MEAN(data[24,24,ind-30])

Results=MPFITFUN("GAUSSIAN",wavelength, data-continuum,error,[0.4,6917.0,2.0])

Fit_parameters=MPFITFUN("FUNCTION_NAME", xvalues, yvalues, error, initialguess_param)

And overplot the result:

PLOT,wavelength, data[24,24,*], xrange=[6850,6950]

OPLOT, wavelength, GAUSSIAN(wavelength,results)+continuum,color=cgcolor('green')

As a final part – lets create a program to fit GAUSSIANS across the SAMI maps!