Tutorial 7.5: Extra Credit

Imad Pasha
Chris Agostino

March 21, 2015

1 Introduction

For the sake of all of our spring breaks, I'm going to try to make this as short and straightforward as possible. My main goal is to simply expose you, at least conceptually, to one of the main things you will be seeing in A-120. (As is much of this class). Today, we are going to get back to working with spectra.

2 2D Spectra

In tutorial 3, we worked with a spectrum taken from a 1-D spectrometer (with a single row of ccd pixels picking up the spread light). In a more realistic case, we work with spectra that is taken using a complicated setup that diffracts lights in two directions at once (up down and side side) by angles defined by the grating equation,

$$m\lambda = d(sin\theta_i + sin\theta_m), \tag{1}$$

where m refers to the order, θ_i the angle of incidence, and θ_m the angle each order is refracted at, respectively. This system allows us to look at spectral features in much smaller ranges. If you load up one of these 2d spectra (which are fits files) and plot them using imhshow, you will see something like figure 1. This doesn't look like much, but we will soon be extracting the spectral information from this image.

Go ahead and load the fits image (or txt backup) into python like we did in tutorial 5, and use imshow to recreate figure 1. Remember to vertically reverse your image first, before using origin='lower' in your imshow command. The setting I used for vmin was np.median(image), and I didn't specify a vmax.

3 Extracting the Orders

At the end of the day, you can think of the 2d image as a convenient way to store a really long spectrum, by chopping it into pieces and stacking them one on top of the other. That's not quite right, since there is overlap from one row to the next, but its close enough.

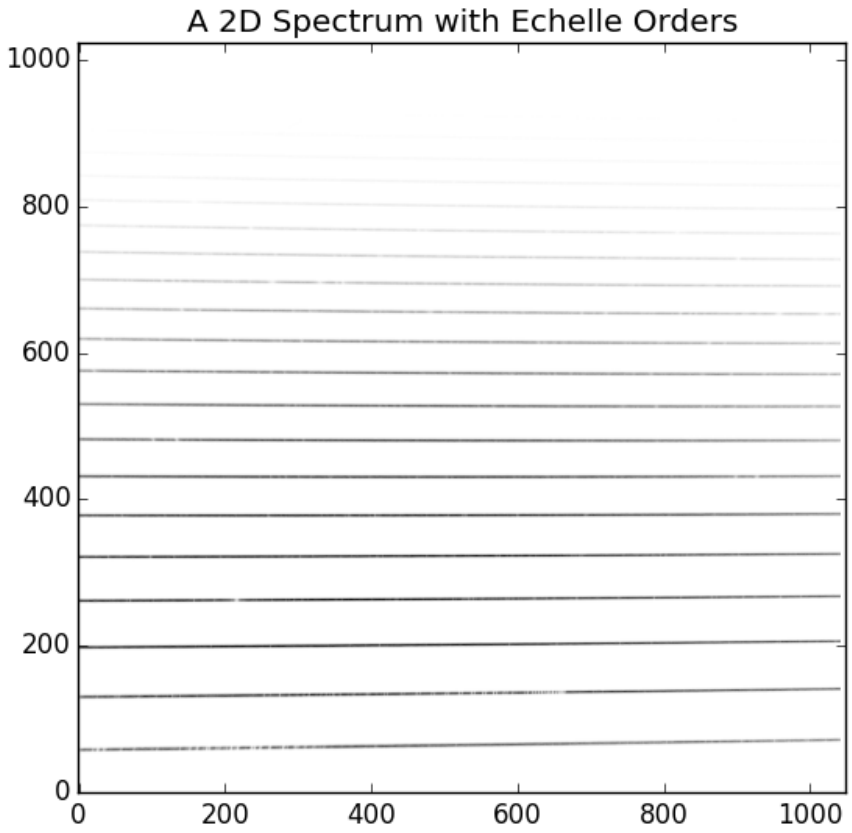

Figure 1: A Typical Spectrum taken using a CCD pixel array. Each of the black "bars" contains spectral information in what are known as "echelle orders". Notice how they get dimmer as you move to higher orders. There's a lot going on here, with the orders actually shearing and tilting, which is all stuff you'll have to deal with in lab, but not here. Each order represents about 20 nm of range in wavelength, and they (sort of) stack end to end.

For now, lets pick a single order to extract and look at. In fig. 1 there is an echelle order that seems to line up almost perfectly with the 200th y pixel. we want to pull it out. If you zoom in on the area around 200, you'll see that the dark bar of the order itself is contained between 193 and 203 (fig. 2).

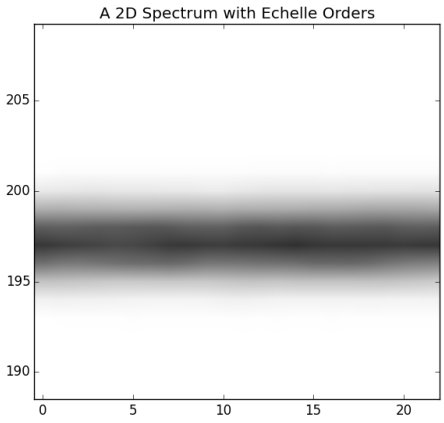

Figure 2: Zoomed Image showing vertical extent of echelle order

Extract the relevant rows of the fits image between 193 and 203 and store them in a variable called order1. You will want to remember your 2D slicing technique and how to pull rows vs columns.

Notice, we still have a 2D image, not a 1D spectra. We just pulled the part of the 2D image containing one order. We will now perform a step known as flattening. Basically, there are a few ways to do this. If you wanted to pull a 1D spectrum out of this long skinny 2d image, you could just pick one row within it and read off the values from it. A slightly more statistically relevant way would be to take the vertical mean or median of every column within your long horizontal bar. This can be accomplished by setting

spectrum = order.mean(axis=0)

By choosing axis=0, you are telling it to make a 1d array with the means of every column in your bar (flattening it). Try just calling plt.plot(spectrum) and see what it looks like. You should get something like fig. 4.

Lets be real, this is pretty cool. All of the information about these solar absorption features is contained within how bright/dim that black bar from before was (though we couldn't detect any difference with our eye. Though if you look back at fig. 1, you'll see it gets lighter towards the right, which is mirrored in how this spectrum tails off towards the right end of the plot (this is primarily a spectrometer effect, and the next order up picks up where this one starts dying down)).

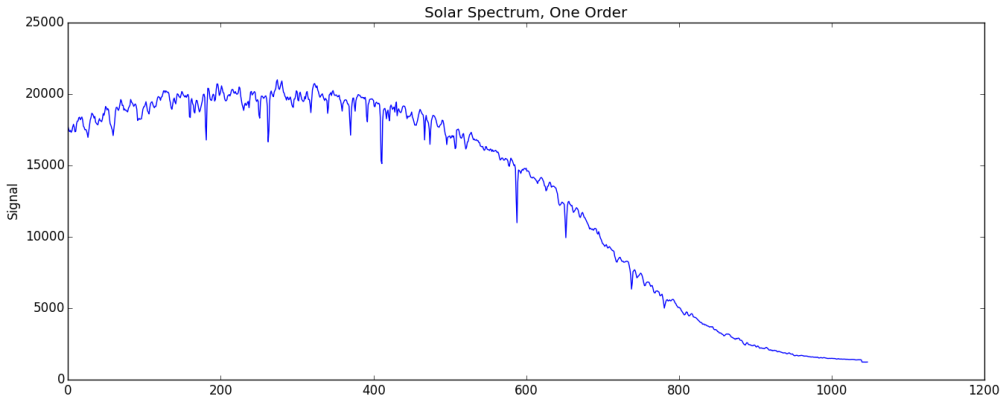

Figure 3: The 1D solar spectrum extracted from the 2D image.

4 Fitting

Just for fun, use what you know about np.polyfit to overplot a 4th degree fit to this die off and see if it matches up pretty well. It should look something like this:

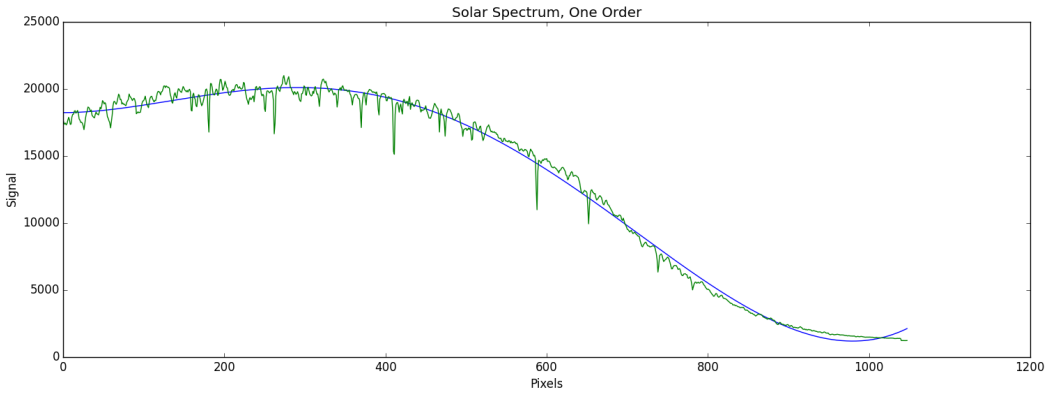

Figure 4: Fitting a 4th degree polynomial to the solar spectrum

Doing such a fit can be useful for generating an idea of the continuum spectrum- the overall shape minus the little absorption dips.

Hopefully that was short and sweet. Not much in the way of analysis, but you now know how to get 1D spectra out of 2d fits images from spectrometers. Enjoy the rest of your spring break!