

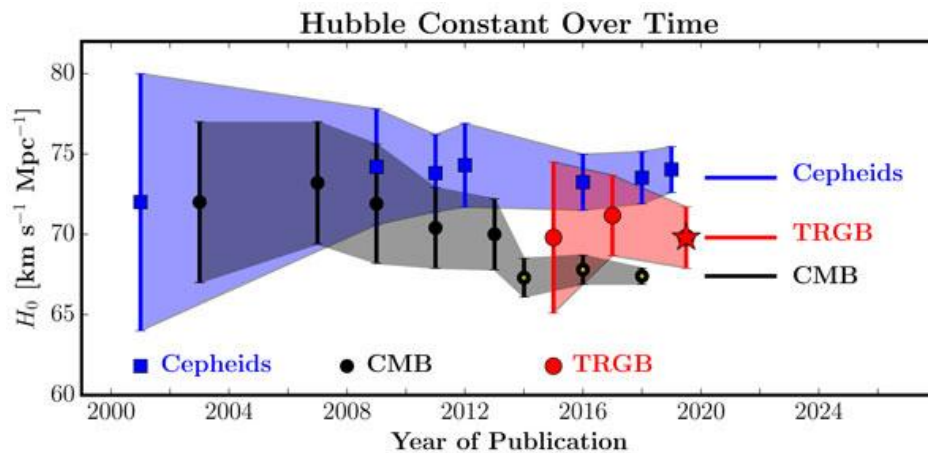
# Hubble Law Part 1: Extragalactic Redshift

## Objective:

In the 1920's, there was a debate between astronomers on the existence of galaxies, or what they referred to as 'Island universes'. Known as the Great Debate, Harlow Shapley and Heber Curtis were at odds on origins of spiral galaxies. One thinking that these nebulae like objects are held within our galaxy, which was believed to be the entirety of our universe at the time, whereas the other postulated that their distances spanned much further than our milky way. It wasn't until Edwin P. Hubble discovered that distant galaxies were all moving away from the Milky Way (and the Local Group) that astronomers started to believe that these extragalactic objects existed. Not only that, the farther away he observed, the faster the objects were receding. He found the relationship that is now known as Hubble's Law: the recessional velocity of a galaxy is proportional to its distance from us,

$$v = H_0 \cdot d$$

Where  $v$  is the velocity along the observers line of sight in km/s,  $d$  is the distance to the object in parsecs, and  $H_0$  is the notorious Hubble constant. In our more modern context of astronomy, the Hubble constant is a fundamental constant indicating the expansion rate of spacetime and the age of our universe as a whole, and is studied extensively in the field of Cosmology. The value of the Hubble constant has seen drastic revisions throughout the last century. Even today there isn't a consensus. This is largely due to how involved the Hubble constant is within various fields of astronomy & cosmology. It shows up in the Cosmic Microwave Background (CMB), Structure Formation of the universe, as well as our standard candles (Cepheids, SNIa, etc). This is highlighted in Fig. (1).



Today's measurements of the Hubble constant are far more sophisticated than that of Hubble's original calculation. His original calculations put the Hubble constant to around 500 km/s/Mpc, more than five times current values. The reason for this discrepancy is our poor understanding of galactic distances in the early 20th century. At the time, galaxies were assumed to be much closer to us than where we know them to be today. This lab will focus on using similar methods to determining both the velocity and distances of galaxies in an attempt to calculate the Hubble constant. Let's see if we can do better than Hubble.

## Finding Radial Velocities of Galaxies:

The velocity is relatively easy for us to measure using the Doppler effect. An object in motion (in this case, being carried along by the expansion of space itself) will have its radiation (light) shifted in wavelength. For velocities much smaller than the speed of light, we can use the regular Doppler formula:

$$\frac{\lambda - \lambda_0}{\lambda_0} = \frac{v}{c}$$

Where  $\lambda$  is the measured wavelength,  $\lambda_0$  is the wavelength measured in the rest frame,  $v$  is the velocity along that line of sight (radial velocity), and  $c$  is the speed of light. The quantity on the left side of this equation is usually called the redshift, and is denoted by the letter  $z$ . The velocity of the galaxy is determined by measuring the redshift of spectral lines in the spectrum of the galaxy. Most galaxies are rich in spectral features: the "K and H" lines of ionized calcium and the H-alpha line of hydrogen.

Let's start by reading in the spectral data of a single galaxy and try to automate the process once we have a grasp on the process. Load a single raw data file of Normalized flux (column 1) and wavelength in Angstroms (column 2).

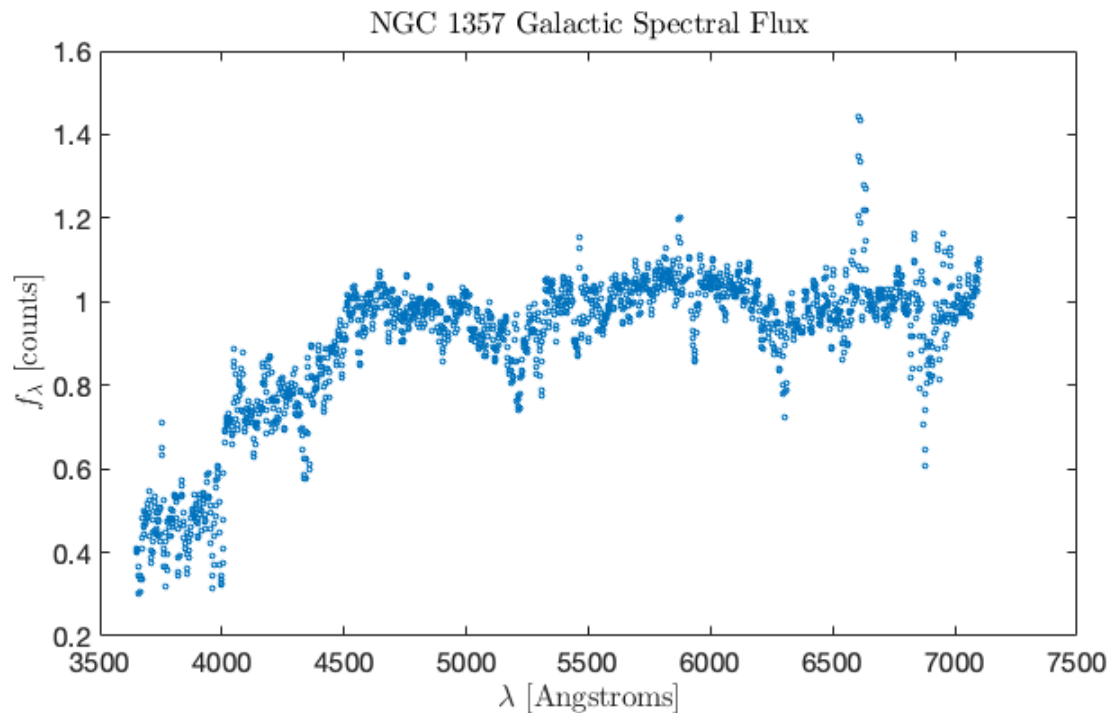
```
directory = '~/tutorialSims/HubbleLaw/NGCData/Spectra/';
files = dir(fullfile(directory, '*.txt'));
rawData = dlmread(strcat(directory,files(1).name));

wavelength = rawData(:,1);
normal_flux = rawData(:,2);
```

The data should now be visible in the **workspace** environment as 1-dimensional data arrays. Next would be to inspect the spectral data by plotting each point. Making sure not to plot it with a line, as this can cause issues in interpreting noisy spectral data.

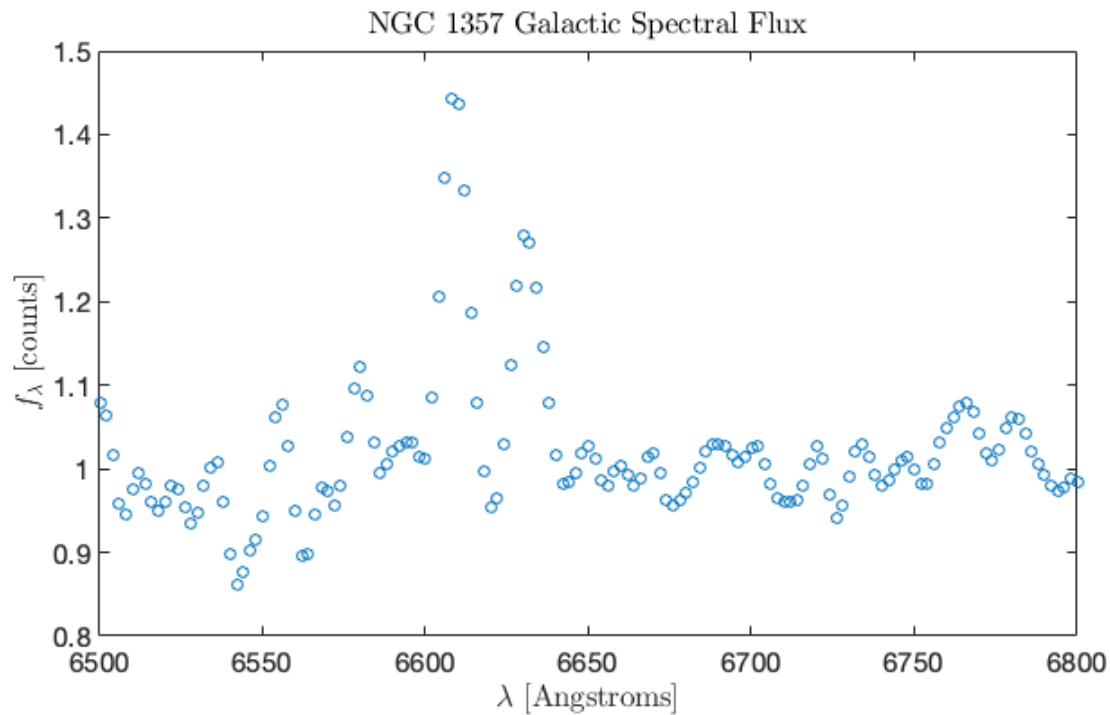
```
fig1 = figure('Renderer', 'painters', 'Position', [500 500 700 400]);
ax1 = axes(fig1);
grid on

p = plot(wavelength, normal_flux, 'o','MarkerSize',3);
set(gca,'fontsize',15)
title('NGC 1357 Galactic Spectral Flux','Interpreter','latex')
xlabel('$\lambda$ [Angstroms]','Interpreter','latex')
ylabel('$f_{\lambda}$ [counts]','Interpreter','latex')
```



These plots of "jiggly lines" are telling us all about these galaxies, just as stellar spectra tell us all about stars. Remember the primary objects found in spiral galaxies: stars of all ages, masses, and composition; dust; and HII regions. We expect, because the bright HII regions and massive OB stars will dominate the light of a spiral galaxy, to see strong emission lines of hydrogen. We can even see that the hydrogen alpha line is the strongest peak in the spectrum at 6562.8 Angstroms. To make it things visually easier to work with, let's focus the plot around some range around the suspected hydrogen alpha peak.

```
p0 = plot(wavelength, normal_flux, 'o');
xlim([wavelength(1) wavelength(end)])
xlim([6500 6800])
set(gca,'fontsize',15)
title('NGC 1357 Galactic Spectral Flux','Interpreter','latex')
xlabel('$\lambda$ [Angstroms]','Interpreter','latex')
ylabel('$f_{\lambda}$ [counts]','Interpreter','latex')
hold on %include the hold on statement so the figures carry through
```



## Smoothing the Spectra:

Not all of the lines come from the light of the galaxy. Each spectrum contains noise; we just cannot get away from it. You should notice that some of the spectra are much "noisier" than other spectra. This noise tends to hamper accurate identification of some of the lines. We can smooth out the function using a smoothing function; in this case, a n-point moving average smoother is build into MATLAB *smooth()*,

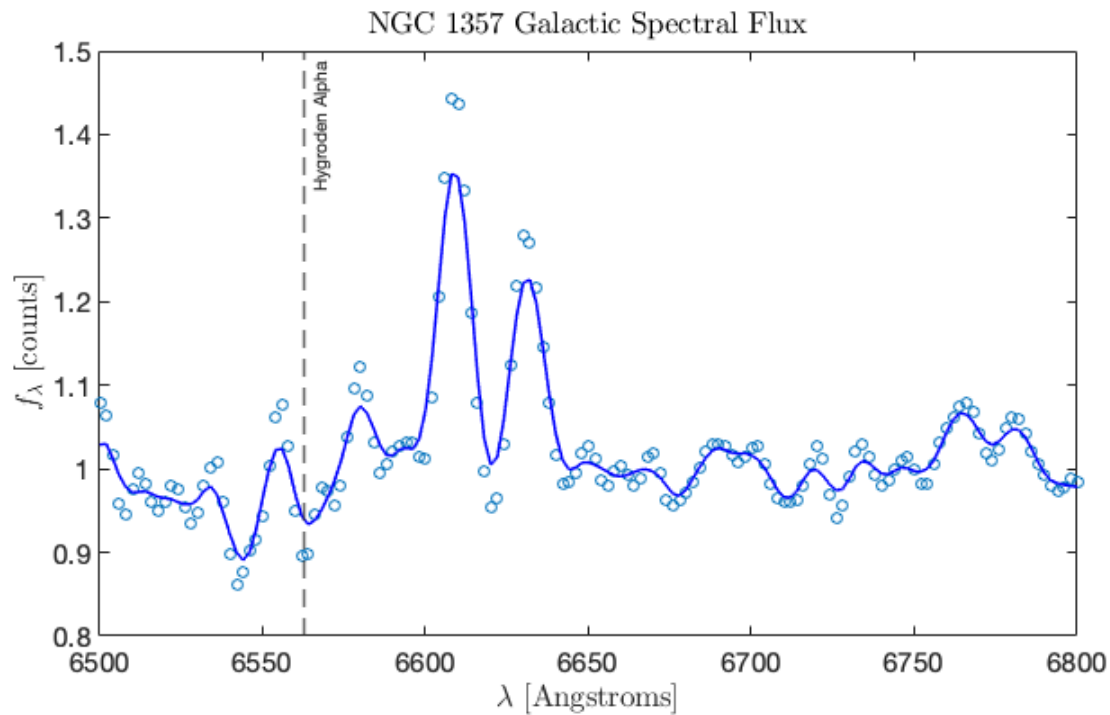
$$y'(n) = (y(n-2) + y(n-1) + y(n) + y(n+1) + y(n+2))/5$$

```
smooth_flux = smooth(normal_flux);
```

Now plot the smoothed data on top of the original data to see if we can identify the lines better. Knowing visually where the unshifted hydrogen alpha line will make the inspection process easier for us. To do this, we'll add an annotated vertical line to overlay on our plot using *xline(value)*.

```
p1 = plot(wavelength, smooth_flux, '-b','LineWidth',1.5);
p1.Color(4) = .5;

x1 = xline(6562.8,'--','Hygroden Alpha');
x1.LineWidth = 1.5;
x1.FontSize = 25;
x1.LabelVerticalAlignment = 'Top';
```



## Finding the Peaks:

Looking at the smoothed data we can easily point out the various emission and absorption lines of the galaxy. In our case, we're concerned with only the hydrogen alpha line which is an emission line. We can focus our attention on trying to find all of the maximums of the data instead of the minimum. It may look like a daunting task, luckily MATLAB provides us with in house peak finding algorithms, in the form of `findpeaks(data)`. It takes in our (smoothed) data and gives back multiple arguments, in order:

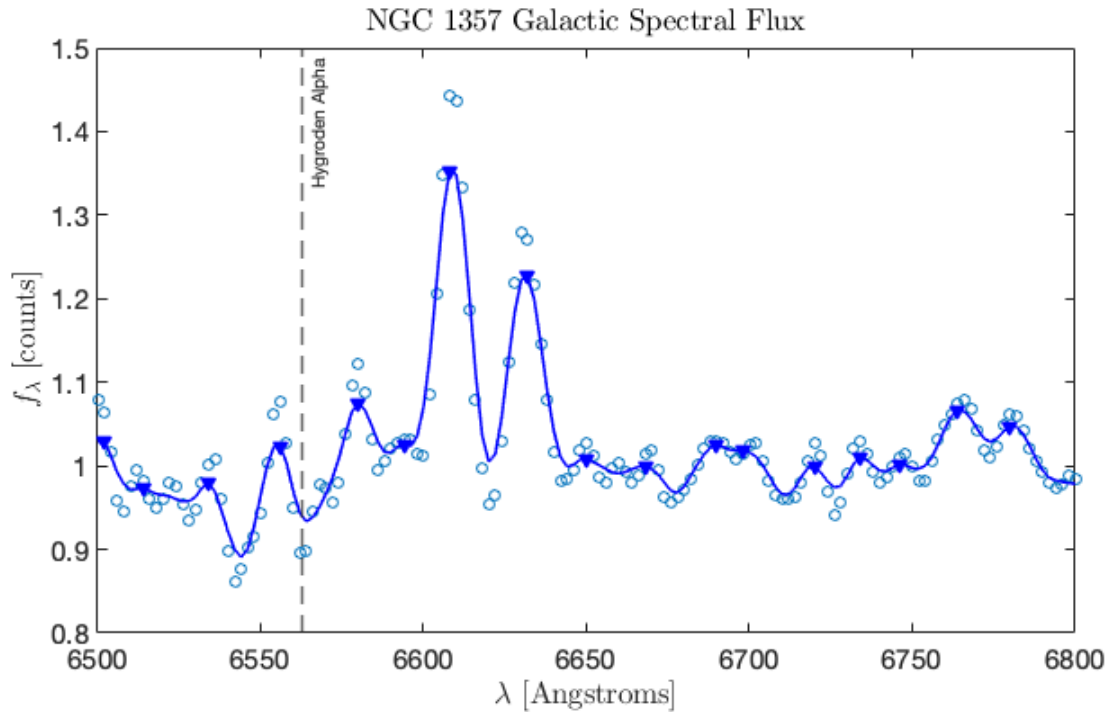
- `pks` → a 1D array of the peak values (maximums)
- `locs` → the locations of all of the peaks, in order of `pks`
- `w` → a form of the width of the peak similar, but not exactly, to the FWHM
- `p` → the height of the peak relative to its baseline (where it reaches its first minimum)

```
[pks,locs,w,p] = findpeaks(smooth_flux);
```

```
pk_wavelength= wavelength(locs);  
pk_flux = smooth_flux(locs);
```

Now plot the peak points overlaying both our original data and the smoothed data.

```
p2 = plot(pk_wavelength, pk_flux, 'vb','MarkerFaceColor','b','MarkerSize',7);
```



From inspection we can notice that the data's hydrogen alpha peak  $\lambda$  (the largest emission line peak) is to the right of the at rest hydrogen alpha line  $\lambda_o$ . Meaning, that our spectrum is redshifted and therefore moving away from us, in our line of sight. The next step would be to create a method of accessing the position of the max peak, and generalize it for all our galactic spectra. (Hint: Look up the *max(data)* and the *find(condition)* functions). at the end you should have, the wavelength of the max peak  $\lambda_{max}$ , the value of its prominence  $f(\lambda_{max})$ , and the corresponding width  $w(\lambda_{max})$ .

```
maxpk = max(pk_flux);
maxindex = find(pk_flux == maxpk);
maxwavelength = pk_wavelength(maxindex);
maxwidth = w(maxindex);
```

## Gaussian Peak Fitting Consistency Check:

The target emission peak has been located and information has been extracted  $\lambda_{max}$ ,  $f(\lambda_{max})$ ,  $w(\lambda_{max})$ . We'll use this information to check the consistency of the emission peak profile. Ideally, these emission and absorption lines are modelled by a Gaussian profile. Although this is not exactly correct. Spectral lines will be broadened and skewed by various astrophysical processes, such as rotation and pressure effects. Since we're dealing with galactic spectra, these effects are minimal compared to dense (and sometimes fastly rotating) stars. With this assumption, we'll fit a Gaussian profile over our shifted hydrogen alpha peak using the model,

$$P(x|a_1, a_2, a_3, a_4) = a_1 \cdot \exp(-((x - a_2)/a_3)^2) + a_4$$

Here  $a_1$  is the peak height,  $a_2$  the corresponding central wavelength,  $a_3$  encodes the width, and  $a_4$  is a background shift. If you need a reminder of how to implement a model to data using the Least Squares/ $\chi^2$

fitting function, go back to *example1.mlx* of the tutorial. First thing, create the *fittype* using our above model (or any model of your choice <https://www.mathworks.com/help/curvefit/list-of-library-models-for-curve-and-surface-fitting.html#btbcvnl>).

```
ft = fittype('a1*exp(-(x - a2)/a3).^2) + a4');
```

Unlike in *example1.mlx*, where there was clearly only one peak, our data has hundreds of potential peaks that the fitting algorithm might converge to. To ensure that the fitter found our hydrogen alpha line we'll help it by setting "starting points" (or "most likely around here" points) and defining upper/lower bounds on each parameter. The starting points used will be equal to the values our *max* peak function found  $(a_1, a_2, a_3) = (f(\lambda_{max}), \lambda_{max}, w(\lambda_{max}))$ . We're missing the last parameter, the background  $a_4$ .

You'll notice that the emission lines lay on top a background function, the full dataset slowly increasing and plateauing after  $\lambda = 4500$  Angstroms. If we had no emission or absorption lines in our spectra, the only thing left would be this background usually referred to as the continuum of the spectra.  $a_4$  is only a constant, and therefore can't encapsulate all of the continuum background. Find a way to estimate, reducing the data to a single value. This, of course, could be the average value of our data. For better accuracy we'll probably want to focus our averaging for only the flat part of the continuum, near our emission peak in question. We'll call this variable *bkg*.

```
bkg_first_index = find(wavelength == 6500);
bkg_second_index = find(wavelength == 6800);
bkg = mean(smooth_flux(bkg_first_index:bkg_second_index));
```

Now that we have our starting points, input them into the *fit()* function using the fit type already created. As mentioned above, a starting point and both, an upper and lower bound will be given to the fitter so to converge onto the proper emission line:

- Starting Point:  $[f(\lambda_{max}), \lambda_{max}, w(\lambda_{max}), bkg]$
- Upper Bound:  $[a_1^{up}, a_2^{up}, a_3^{up}, a_4^{up}]$
- Lower Bound:  $[a_1^{low}, a_2^{low}, a_3^{low}, a_4^{low}]$

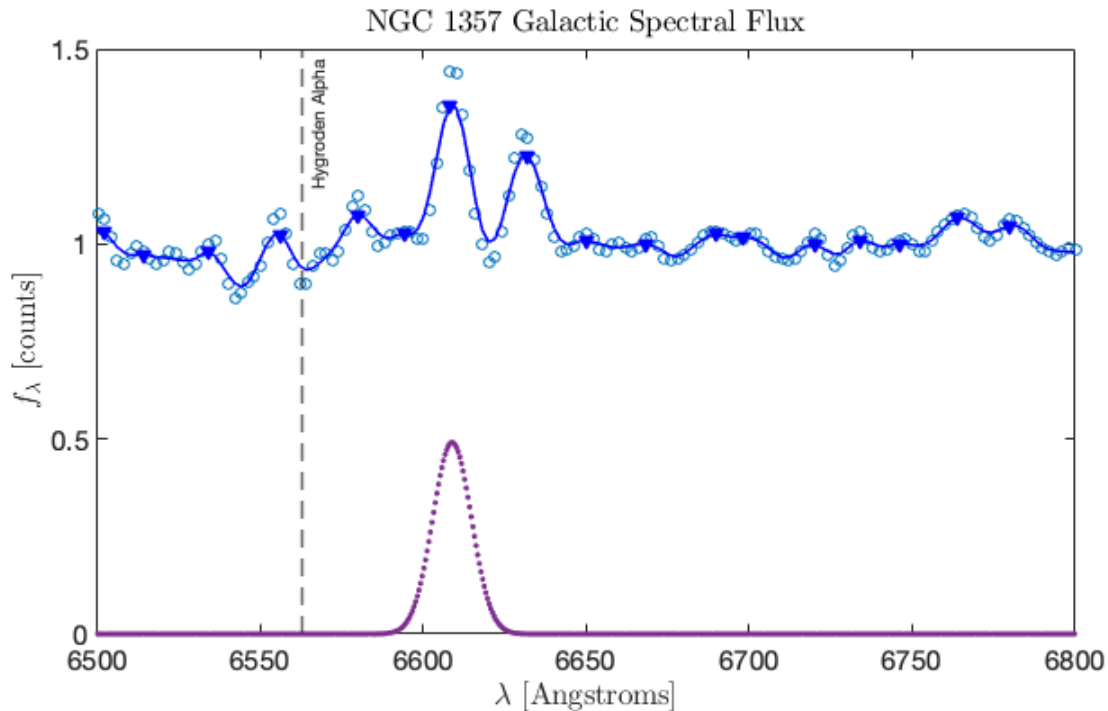
Set your upper and lower bound values to what makes physically sense. Some tuning might be necessary to get the convergence right.

```
f = fit(wavelength, normal_flux, ft, 'StartPoint', ...
        [maxpk, maxwavelength, maxwidth, bkg], Lower = [0, 6500, 0, 0], ...
        Upper = [3, 6700, 15, Inf])
```

```
f =
General model:
f(x) = a1*exp(-(x - a2)/a3).^2 + a4
Coefficients (with 95% confidence bounds):
a1 =      0.4912   (0.3031, 0.6794)
a2 =      6609   (6606, 6611)
a3 =       8.366   (4.662, 12.07)
a4 =      0.8989   (0.8904, 0.9074)
```

Inspecting the fitter output is one way of checking for consistency of our values. For full visual inspection we'll plot a Gaussian function with the parameters estimated with the fitter. To do so, create a vector of x-values (using `linspace()`) and create a new Gaussian function with the x-vector as input. Finally, plot it over our figure.

```
x = linspace(6500,7000,1000);
Gaussfit = f.a1*exp(-((x - f.a2)/f.a3).^2);
p3 = plot(x,Gaussfit,'.', 'LineWidth',1.5);
hold off % stop carrying over the figure
```



## Putting it all Together to Calculate $\lambda_{H\alpha}$ :

The goal of this lab is to find the redshifts, and correspondingly the radial velocities, of the galaxies surveyed in:(put database here). So far, we've spoken about a single galaxies spectrum. It's now time to put it all together by looping over all datasets in folder (folder name). Creating an array of all of the final radial velocities in order of the filenames. You're for loop should contain the following steps in order

- Read in the  $i^{th}$  file in the directory.
- Smooth the flux data.
- Find  $(f(\lambda_{max}), \lambda_{max}, w(\lambda_{max}))$  for each file.
- Calculate the redshift and save it in an array.
- Save the widths into an array.

```
i = 0;
redshift = zeros(length(files),1);
linewidth = zeros(length(files),1);

for file = files'
```



```

i = i + 1;
file.name;
Data = dlmread(strcat(directory,file.name));

wave = Data(1000:end,1);
flux = smooth(Data(1000:end,2));

[pksl,locsl,wl,pl] = findpeaks(flux);
pkwavel = wave(locsl);
pkfluxl = flux(locsl);

maxpkl = max(pkfluxl);
maxindexl = find(pkfluxl == maxpkl);
maxwavel = pkwavel(maxindexl);
maxwidthl = wl(maxindexl);

redshift(i) = (maxwavel - 6563.0)/6563.0;
linewidth(i) = maxwidthl/1.5;

```

end

The last step being to use your calculated difference  $\lambda - \lambda_o$  in the Doppler shift relationship. Since we're taking a ratio of wavelengths, it doesn't matter what units we calculate our difference in. Don't forget to calculate the uncertainty for each data point. Since the width of our emission lines are larger than the resolution of the data  $\Delta\lambda = 2$  angstroms, we'll use the width instead to estimate the error in our wavelengths. Propagating this to the redshift using,

$$\sigma_v = \frac{\partial v(\lambda)}{\partial \lambda} \sigma_\lambda.$$

```

c = 299792.458; % speed of light in km/s
radial_vel = (redshift)*c;
radial_vel_std = (linewidth/6563.0)*c;

```

## Write to file:

Write the output into a file for use later.

```

A = [radial_vel radial_vel_std]';
fileID = fopen('~\tutorialSims\HubbleLaw\NGCData\radial_vel.txt','w');
fprintf(fileID,'%5d %5d\n',A);
fclose(fileID);

```

## Questions: