Signal Processing in Astronomy
Project: Kinematics of Real Galaxies

Seyit Hocuk & Umut Yildiz

Abstract

The stellar kinematics of NGC 3379 is determined in this project. The kinematics are calculated using one of the four methods summarized by Jesus Falcon. Furthermore, other methods are discussed. Cross correlation method by Simkin (1973) and later by Tonry & Davis (1979) is tested and used to determined the kinematics of NGC 3379. The results are compared to later studies of velocity dispersions and the rotation curve of this galaxy by Hjorth & Tanvir (1997) and Halliday et al. (2001). We see that they are in good agreement.

1 Introduction

This project is part of a task for the course *Signal processing in astronomy* to learn how to process and analyse a raw signal. The course is supervised by Reynier Peletier.

The aim of this project is to determine the stellar kinematics of the galaxy NGC 3379. To achieve this goal, there is a given set of spectral data of a galaxy and four different stars. Given the approximation that the galaxy spectra is an average of these four star spectra. Our task is to determine the kinematics from the given data by using one of the four methods summarized by an article of Jesus Falcon. Furthermore we are going to discuss the differences of all the four methods and finally compare the kinematic results of the chosen method.

In the next sections, each question asked in the project description are answered accordingly.
For all the computing and plotting, the program Matlab is used.

2 Method Descriptions

The bulk of kinematic information available on elliptical galaxies is retrieved through observation and analysis of the line-of-sight velocity distributions (LOSVD's) of the stars in these galaxies.
The basic idea behind the study of LOSVD's is that a galaxy spectrum can be generated by the spectrum of a star of similar type that has been

red-shifted by the rotation of the galaxy and smeared out by the velocity dispersion of the stars (Minkowsky 1962). Galaxy spectra can be tought of the superposition of all the stellar spectra in that region. However, due to the different LOS of stars, the individual star spectra will be shifted relative to the other stars. This way, the lines in the galaxy spectrum gets broader. This is also the convolution of the spectrum of the stars with a velocity distribution along the LOS.
Assuming that all the stellar spectra are identical (S(u)), with u the stellar velocity, the galaxy spectrum can be given by the equation:

$$G(u) \approx int F(v_{los}) \cdot S(u - v_{los}) \cdot dv_{los}, \tag{1}$$

with F(v_{los}) as the fraction of stars contributing to the spectrum with a line of sight between V_{los} and $vlos$ + dv_{los}.

2.1 Fourier Quotient

Inherent in comparing galaxy spectra, a huge problem is the difficulty in distinguishing the effects of differences in line width from differences in line depth. Various systematic plans of actions for finding velocity dispersions have been developed to this day. These schemes allow for variation in both line width and depth (Simkin 1974). In Sargent's (Sargent et al. 1977) paper they estimated velocity dispersions in 13 galaxies using a Fourier technique closely related to Simkin's.

2.1.1 Description of the Fourier Quotient Method

In order to develop this method Sargent assumed that each galaxy spectrum is the convolution of an appropriate mean stellar spectrum with a Doppler broadening function. Then, the convolution theorem, the Fourier transform of the galaxy spectrum would be the product of the transform of the mean stellar spectrum with the transform of the broadening function (Simkin 1974; Illingworth and Freeman 1974).
The very important assumption that Sargent used that the broadening function is a Gaussian characterized by a dispersion σ and a redshift z. The theory behind it is supposing that we could compute the Fourier terms of a galaxy and mean stellar spectra. Then it is possible to fit the transform of our broadening function (again a Gaussian) to the ratio of the galaxy transform to the stellar transform, adopting the values of σ and z which yield the best fit.
We can estimate velocity dispersions by taking discrete Fourier transforms of our modified spectra $G(k)$ and $S(k)$ using an FFT algorithm and by then fitting the quotient to the Fourier Transform of our broadening function,

$$F(k) = \frac{G(k)}{S(k)} \approx \gamma \cdot exp[-\frac{1}{2}\left(\frac{2\pi k\sigma}{N}\right)^2 + \frac{2\pi\nu ki}{N}], \tag{2}$$

where σ and v are the velocity dispersion and mean radial velocity respectively. The stellar kinematics parameter γ is a normalisation factor which measures the strength of the galaxy lines with respect to stellar lines and N is the total number of pixels.

2.2 Cross-Correlation Method

Suppose $g(n)$ be the spectrum of a galaxy whose redshift and velocity dispersion are to be found and suppose $t(n)$ be a template spectrum of zero redshift and instrumentally broadened stellar line profiles. These spectra are discretely sampled into N bins, labelled by bin number n; the relationship between wavelength and bin number is,

$$n = A \cdot ln\lambda + B. \tag{3}$$

Because the spectra are binned linearly with $ln\lambda$, a velocity redshift is a uniform linear shift. The spectra are assumed periodic with period N for the purposes of discrete Fourier transforms and correlation functions derived hereform. Moreover, the spectra are continuum subtracted, endmasked, and filtered to remove low-frequency spectral variations which arise from continuum variations and non-uniform photo-cathode sensitivity, and high frequency noise components beyond the resolution. The endmasking assures that mismatch between the ends of the spectrum is removed.

2.2.1 Theory

Suppose $G(k)$ and $S(k)$ to be corresponding discrete Fourier transforms defined by,

$$G(k) = \sum_n g(n) e^{\frac{-2\pi ink}{N}} \tag{4}$$

and suppose Δ_g and Δ_k be the rms of the spectra;

$$\sigma_g^2 = \frac{1}{N}\sum_n g(n)^2. \tag{5}$$

At last, define the normalised cross-correlation function $c(n)$;

$$c(n) \equiv g \times s(n) = \frac{1}{N\Delta_g\Delta_k}\sum_m g(m)s(m-n). \tag{6}$$

Then Fourier Transforming the formula for $c(n)$ gives;

$$C(k) = \frac{1}{N\Delta_g\Delta_k} G(k)S^*(k), \tag{7}$$

where $*$ indicates complex conjugation.

2.3 Fourier Correlation Quotient

The procedures mentioned above which are used for the analysis of the kinematics of early type galaxies are used for determining the velocity distribution along the LOS. All these methods assume that the broadening of dispersion is to be of a Gaussian shape. This method is different in the way that it recovers the shape of the kinematical broadening function of early type galaxies. Fourier Correlation technique is a hybrid method which avoids the shortcomings of the deconvolution on the peak from the autocorrelation function of the template star. This procedure is shown to be less sensitive to template mismatching than the standard Fourier Quotient method. It allows not only a very reliable measurement of velocity dispersions but also a detailed analysis of the shape of the broadening function, such as an investigation of asymmetries caused by rotation, velocity dispersion anisotropy or multi-component structure.

$$F(k) = \frac{G(k) \cdot S^*(k)}{S(k) \cdot S^*(k)}. \tag{8}$$

As indicated eaerlier the important common property of both previous methods is that they assume intrinsically Gaussian velocity distributions. However, the results on the kinematics of the elliptical galaxies demonstrate that this simplifying assumption is not adequate (Bender 1990).

2.4 Unresolved Gaussian Decomposition

This algorithm models the LOSVD as the sum of a set of Gaussian distributions uniformly spaced in velocity. By choosing the dispersion, Δv, of these Gaussians so that the seperate components are resolved according to the Rayleigh criterion, such a sum can model any LOSVD that is smooth on scales smaller than Δv. The algorithm involves solving for the amplitudes of the individual components to produce the LOSVD that, when convolved with the stellar template, best reproduces the observed galaxy spectrum in a least-squares sense.

The optical spectrum at any point in a galaxy is made up from the sum of the spectra of its component stars which lie along that line of sight. These stars will be travelling at different line-of-sight velocities, and so the absorption lines in the observed galaxy spectrum will be broadened by the Doppler shifts in the individual stellar spectra. If we assume that the stars can all be characterized by a single 'template' spectrum $S(\lambda)$, then the galaxy spectrum $G(\lambda)$ is just the convolution of $S(\lambda)$ with the line-of-sight velocity distribution (LOSVD) of the stellar motions, $F(v)$:

$$g(ln\lambda) = \int F(v) S(ln\lambda - \frac{v}{c}) dv, \tag{9}$$

where c is the speed of light.
Unfortunately, the deconvolution of this equation to measure $F(v)$ is notoriously unstable in the presence of noise (Sergent et al. 1977, Tonry & Davis 1979). This instability is overcome by forcing $F(v)$ to conform to an easy model such as a Gaussian. But in this way, many kinematic features like the variations in mean and dispersion of the LOSVD at different positions in a galaxy could be estimated only if the assumed model for $F(v)$ is a good approximation to the LOSVD. On the other hand, there is no reason why the LOSVD should have a Gaussian shape. The motions of stars in the solar neighbourhood of the Milky Way show a long low-velocity tail in their azimuthal velocity distribution (Mihalas & Binney 1981), and the kinematics of external galaxies, both disc and elliptical, are also expected to show departures from a Gaussian distribution. The method is formulated by the following equation with the composition of several Gaussians uniformly distributed in mean velocity and velocity dispersion;

$$F(k) \approx \sum_{k=1}^{N} a_k \cdot e^{-\frac{(v_{los}-v_k)^2}{2\sigma_k}}. \tag{10}$$

In this equation a_k, v_k, σ_k are the amplitude, mean velocity and velocity dispersion of each Gaussian respectively (Kuijken & Merrifield 1993).

2.5 Gauss-Hermite Expansion

In this method, in order to parameterize the line profiles of elliptical galaxies, the observed line profile is expanded as a sum of orthogonal functions in a Gauss-Hermite series. This approach utilizes the fact that Gaussians are good low-order approximations to most realistic line profiles, and naturally leads to two parameters describing the deviations from a Gaussian.

$$F(v_{los}) \approx [\frac{\gamma\alpha(w)}{\sigma}] \cdot [1 + \sum_{j=3}^{n} h_j H_j(w)], \tag{11}$$

$$w = \frac{(v_{los} - v)}{\sigma}, \tag{12}$$

where γ, v, σ characterize the line strength, mean velocity and velocity dispersion of the Gaussian and the Hermite coefficients and (h_j) describes the deviations from the Gaussian shape of the LOSVD (Van der Marel & Franx 1993).

3 Results

For determining the parameters such as velocity dispersion, redshift and rotation, we started out using the Fourier quotient method and Fourier correlation quotient method which seemed an easy task. Far from the truth, we couldn't find a good Gaussian fit to determine the kinematics.
In the end, we decided to use the another method. Our approach was for the most part the same for the second method, the Cross-correlation method, which we present the outline in the subsection.
For all the measurements, we only used the logarithmic datasets.

3.1 Approach

For future calculations we first find how big each pixel is in km/s. In the fits header we find out that each pixel corresponds to an increment of $0.873\mathring{A}$. If we multiply this with the speed of light and then divide by the central wavelength we get each pixel corresponding to 60.73 km/s.

First we read all the fits files into arrays and also take only one dimensional slice of the 2D galaxy data, but we only use the logarithmic spectral data. We chose the slice at exactly the middle of the galaxy, with the help of the program DS9, at the highest intensity. This was at 660 pixels. We then take a nice part of the spectrum where the increase is linear and calculate the continuum with a first degree polynomial and then divide to this and subtract one. This way by subtracting the continuum, the spectrum is nicely workable.
From specific lines in the spectrum of the stars and galaxy we easily detect the shift, the red-shift. See figures (1) and (2) for the galaxy and first star spectra.
We found a red-shift of 14 ± 1 pixels. This corresponds to $\Delta\lambda$ of $12.24\mathring{A}$ ± $0.87\mathring{A}$. Together with subtraction of the continuum, we also correct the datasets for the redshift and normalize it.
After this we use edgetaper to make the edges nicer. Because at the extremes, the sensitivity is tought to be much lower.
Now we Fourier transform all data discretely using FFT algorithm. After transformation we now compute the cross- and autocorrelation in the same way as Simkin (1974) and Tonry & Davis (1979) did. Both cross-correlation between one of the stars and the galaxy and the auto-correlation are needed to calculate the velocity dispersion of the galaxy. To the computed correlation, a Gaussian is fitted as best as possible. To achieve this we zoomed in at the area of the correlation to fit the Gaussian. You can see our fits to correlation of star 1 to itself and to the centre of galaxy NGC 3379, see figures (3) and (4) respectively.
From these fits we get our dispersions (σ) from the widths of the Gaussian and the rotation speed from the amplitude. Our Gaussian is in the form of

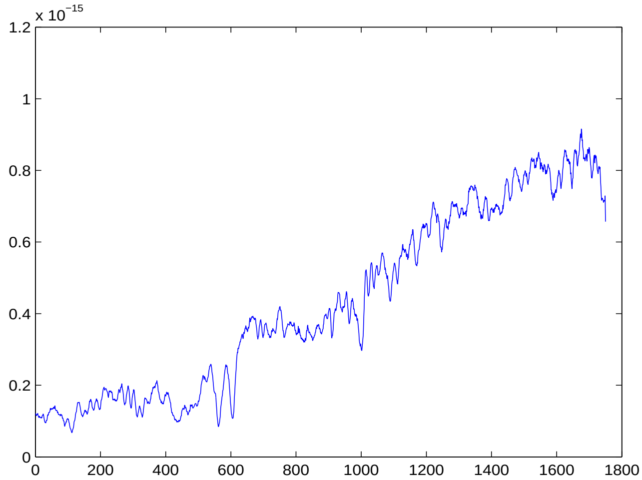

Figure 1: Logarithm of the galaxy spectrum. The x-axis is the spectrum range of 1750 pixels

$A \cdot e^{B \cdot (x-\mu)^2}$, where A is the amplitude of the Gaussian and B equals $\sqrt{\frac{1}{2\sigma}}$. To calculate the velocity dispersion of the galaxy at a certain radius we need both sigmas from the cross correlation and the star autocorrelation. The formula is as follows:

$$\sigma_{gal}^2 = \sigma_{crosscor}^2 - \frac{1}{2}\sigma_{autocor}^2. \tag{13}$$

The results for slice 660 of the galaxy spectrum, which is the galaxy centre, are: σ_{gal}= 4.3574 pixels = 264.61 km/s and v_{rot}= 0.7842 pixels = 47.62 km/s. We will see in the conclusion that these values are in good agreement with other estimates.

To get the velocity dispersion as a function of the radius of the galaxy, we let our code calculate the σ and v_{rot} for each slice of spectrum of the galaxy and plot these values against the radius. The reason we only chose radii between 600 and 720 (in pixel sizes) are because further away in radius the spectrum of the galaxy gets fainter in intensity and much more noise is introduced. Another point is that the velocity dispersion plot becomes flat after a radius anyway. You can see the velocity dispersion plotted against radius in the next plot, figure (5).

As for finding the rotation curve, we first tried to look at the shift of the lines at the different parts of the galaxy. Unfortunately we couldn't find a significant shift. The shift was at best 1 pixel, which is by the way also the scale of our measurement error. Thus, nothing valuable could be said about this.

Another approach we tried was to plot the amplitudes of the cross correlation

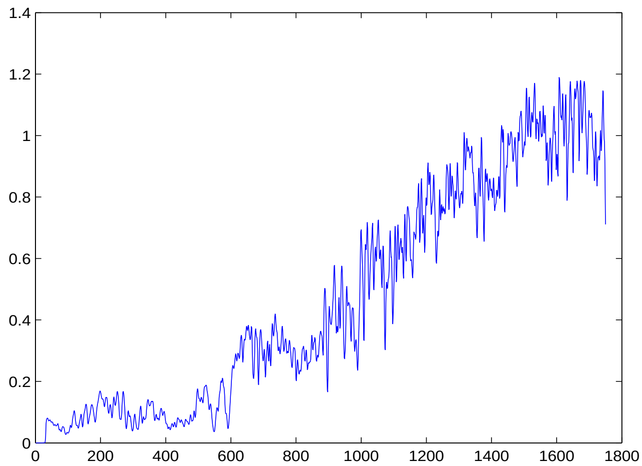

Figure 2: Logarithm of the star 1 spectrum. The x-axis is the spectrum range of 1750 pixels

of the galaxy centre with different parts of the galaxy. You can see this plot in figure (6).
Again we see from this plot nothing similar to the expected rotation curve and apart from a bump in the middle, the rotation velocity at every part of the system is the same around 54 km/s with a maximum of error range of ± 6 km/s.

3.2 Higher order Fourier parameters

As a final attempt to improve the fit, we tried to expand the into Gauss-Hermite series to get higher accuracy. You can see the formula for this on equations 11 and 12 taken from (Van der Marel & Franx 1993). This series leads to two more parameters h3 and h4. We incorporated these parameters in our Gaussian fitting function. Our problem was that Matlab couldn't handle so many parameters together with calculating the Hermite polynomials. The first few Hermite polynomials are given in equation 14:

$$\begin{aligned} H_0(w) &= 1 \\ H_1(w) &= 2w \\ H_2(w) &= 4w^2 - 2 \\ H_3(w) &= 8w^3 - 12w \\ H_4(w) &= 16w^4 - 48w^2 + 12. \end{aligned} \tag{14}$$

To adjust for the problem we calculated the new Gaussian fit with the two extra variables separately.

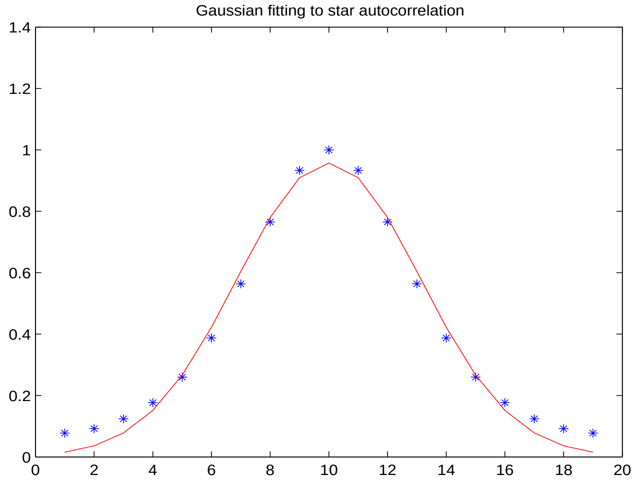

Figure 3: Autocorrelation of logarithm of star 1 with itself. Blue: autocorrelation data points, Red: best Gaussian fit. The axes are in pixel size.

For Gaussian with the h3 we noticed that the fitting was quite the same and the result in velocity dispersion, see figure (7), was almost exactly the same. The results were also the same for rotation giving in both cases same values for rotation and dispersion. This is understandable when looked at our h3 value. This is in the order of 6.0^{-5}. For fitting with the other parameter h4, Matlab still had many problems and very strangely the results were different every time. Sometimes it gave quite nice results other times it was completely off. The unreproducible results are therefore not mentioned.

4 Conclusion

There were four different methods explained in the article by Jesus Falcon and our first goal was to try and compare two of these methods. We started out by choosing the two Fourier Quotient methods, unfortunately we got stuck bad and we had spent too much time in this. So we decided to go further with one other method, the Cross correlation method, which was first initiated by Simkin (1974) and there after improved by Tonry & Davis (1979).

We find from our velocity dispersion plot shown in figure (5), a value of 272.4 km/s at the centre of the galaxy and a mean value of 217.2 km/s. We see that our central velocity distribution might be very high and we looked up another article where this is calculated. We see in the article by Halliday et al. (2001) a central dispersion value of σ_{max}=224.5 km/s. Our result is most likely too high. However when we compare our mean velocity dispersion value 217.2 km/s, it is in good agreement with 209.5 $\pm$ 1.3 km/s

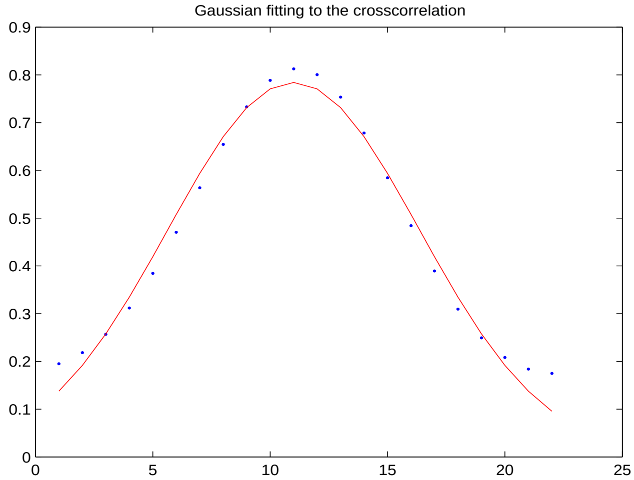

Figure 4: Cross correlation between the centre of the galaxy and star 1 in logarithmic scales. Blue: cross correlation data points, Red: best Gaussian fit. The axes are in pixel size.

(Halliday et al., 2001) and 197 $\pm$ 14 km/s (Hjorth and Tanvir, 1997).

For the rotation curve that we couldn't find, we predict that the given data had no information about the rotation or the data was already corrected for it. It should also be noted that perhaps it was too small for us to detect. But we do have found an average rotation velocity. This was 54 km/s with a large error bar of $\pm$ 6 km/s. Again if we compare our results with that of (Hjorth and Tanvir, 1997) and (Halliday et al., 2001), we see their values are not much different of 50.8 $\pm$ 4.9 km/s and 50.0 $\pm$ 5.7 km/s respectively.

We have achieved these results by correlating the galaxy with the first template star. Our job was also to carry the work out with the other templates and so we did, however these other templates gives quite different and bad results. We found out that all the template stars aren't within the galaxy itself. There are asymmetrical differences between galaxy and template spectrum. Of these four templates we have found that the best matching is the first star.

We also tried to improve on the result by using higher order parameters into our fitting function to change our Gaussian in shape just to get a more accurate fit. In theory this should word nicely. For our case it didn't matter though. We got the same results maybe slight better in the high decimals. This is understandable because our h3 value was in the order of 6.0^{-5}. When you have the new series multiplied by a factor 1 + h3*Hermite3 times the normal Gaussian, we think that the contribution of the higher order is far too low. Our fit is probably for the most cases already good enough.
Our final conclusion to the method used by us, is that we believe this cross

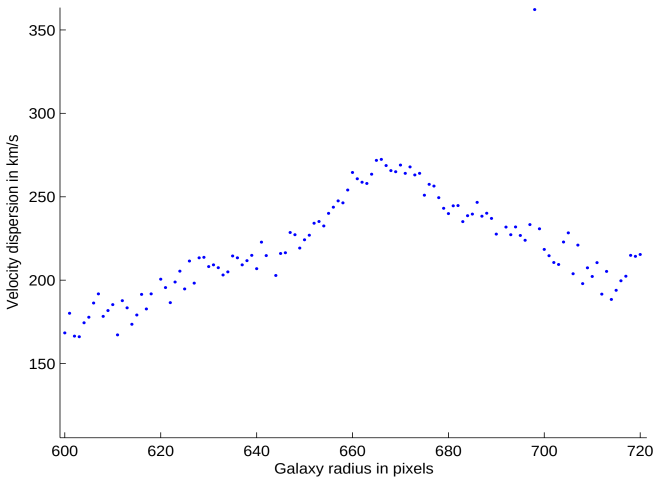

Figure 5: Velocity dispersion plot of the galaxy. Y-axis is the velocity dispersion in km/s. X-axis the radius in pixels with 660 the centre of the galaxy

correlation method is a good way for determining redshifts and velocity distributions. For finding rotation curves however, we are not sure if this is a good way to do it. Because we didn't find the expected curve.

5 The Master Code

```
clear all

basi   = 600;
sonu   = 720;
for nn = basi:sonu


%---------------Reading the files---------------
NGC3379_2d_log = fitsread('NGC3379_2d_log.fits');

%--------Name----------T_eff  log g [Fe/H]
%Ster 1: HD 130705     4336.  2.10  0.4
ster1l = fitsread('ster1l.fits');

%--------Name----------T_eff  log g [Fe/H]
%Ster 2: HD 127334     5545.  3.88  0.15
ster2l = fitsread('ster2l.fits');

%--------Name----------T_eff  log g [Fe/H]
%Ster 3: HD 131976     3506.  4.73   .00
ster3l = fitsread('ster3l.fits');

%--------Name----------T_eff  log g [Fe/H]
%Ster 4: HD 126141     6690.  4.39  0.02
ster4l = fitsread('ster4l.fits');


%---------------Taking a slice of the Galaxy data---------------
slice = nn

Gallog = NGC3379_2d_log(slice,:);
Gallcen = NGC3379_2d_log(660,:);

Gallog = Gallog(800:1600);
Gallcen= Gallcen(800:1600);
ster1l = ster1l(800:1600);
ster2l = ster2l(800:1600);
```

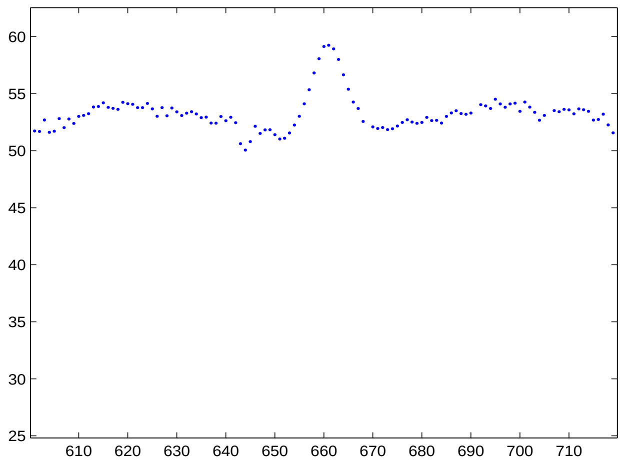

Figure 6: Rotation speed plotted against different radii of the galaxy. Y-axis is the rotation velocity in km/s. X-axis the radius in pixels with 660 the centre of the galaxy

```
ster3l = ster3l(800:1600);
ster4l = ster4l(800:1600);

%--------------Calculating the continiuum from the datasets with a polynomial---------------
xd = (1:length(ster1l));

%galaxy
[p4,s,mu] = polyfit(xd,Gallog,1);
polinomGallog = polyval(p4,xd,s,mu);
[p4,s,mu] = polyfit(xd,Gallcen,1);
polinomGallcen = polyval(p4,xd,s,mu);

%star 1
[p4,s,mu] = polyfit(xd,ster1l,1);
polinom1L = polyval(p4,xd,s,mu);

%star 2
[p4,s,mu] = polyfit(xd,ster2l,1);
polinom2L = polyval(p4,xd,s,mu);

%star 3
[p4,s,mu] = polyfit(xd,ster3l,1);
polinom3L = polyval(p4,xd,s,mu);

%star 4
[p4,s,mu] = polyfit(xd,ster4l,1);
polinom4L = polyval(p4,xd,s,mu);

%--------------Redshifting the star data and substracting the calculated continuum from data---------------
z1 = 14;
array = length(ster1l);
add = 0 + z1;    %additioneel nullen aan beide extremen van de spectrum

%Preallocate matrix
galaksiL= zeros(1,array);
galcenL = zeros(1,array);
ster1ls = zeros(1,array);
ster2ls = zeros(1,array);
ster3ls = zeros(1,array);
ster4ls = zeros(1,array);

for n = 1+add:(array-add)
      galaksiL(n) = Gallog(n)/polinomGallog(n) - 1;
      galcenL(n)  = Gallcen(n)/polinomGallcen(n) - 1;
      ster1ls(n) = ster1l(n + z1)/polinom1L(n + z1) - 1;
      ster2ls(n) = ster2l(n + z1)/polinom2L(n + z1) - 1;
      ster3ls(n) = ster3l(n + z1)/polinom3L(n + z1) - 1;
      ster4ls(n) = ster4l(n + z1)/polinom4L(n + z1) - 1;
```

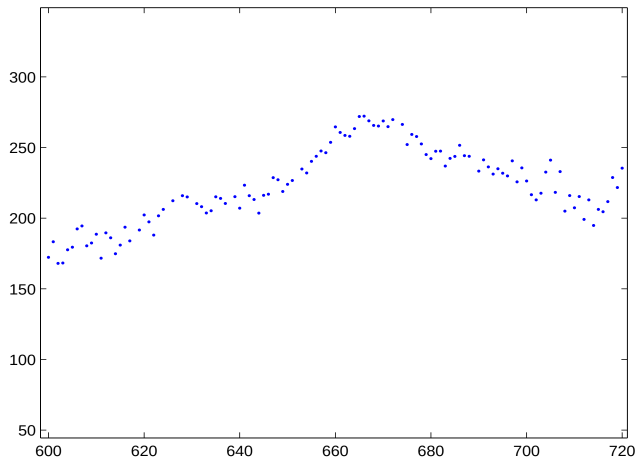

Figure 7: Velocity dispersion plot of the galaxy calculated with fitting Gaussian together with higher order (h3) Hermite term. Y-axis is the velocity dispersion in km/s. X-axis the radius in pixels with 660 the centre of the galaxy

```
end

%---------------Using Edgetaper---------------
PSF = fspecial('gaussian', [1 400], 0.5);
galaksiLE = edgetaper(galaksiL, PSF);
galcenLE  = edgetaper(galcenL, PSF);
ster1lsE = edgetaper(ster1ls, PSF);
ster2lsE = edgetaper(ster2ls, PSF);
ster3lsE = edgetaper(ster3ls, PSF);
ster4lsE = edgetaper(ster4ls, PSF);


%---------------Using FFT---------------
G1L = fft(galaksiLE);
GCL = fft(galcenLE);
S1L = fft(ster1lsE);
S2L = fft(ster2ls);
S3L = fft(ster3ls);
S4L = fft(ster4ls);

%--------------method 2: Cross correlation---------------


%standard deviations of the reduced data
stdG = std(galaksiLE,1);
stdC = std(galcenLE,1);
std1 = std(ster1lsE,1);
std2 = std(ster2lsE,1);
std3 = std(ster3lsE,1);
std4 = std(ster4lsE,1);
N = array;


%auto and cross correlation calculations
Cx   =  ifft( (1/(stdG*std1*N)) * G1L .* conj(S1L));
Cas  =  ifft( (1/(std1*std1*N)) * S1L .* conj(S1L));

Cx2  =  ifft( (1/(stdG*std2*N)) * G1L .* conj(S2L));
Cas2 =  ifft( (1/(std2*std2*N)) * S2L .* conj(S2L));

Cx3  =  ifft( (1/(stdG*std3*N)) * G1L .* conj(S3L));
Cas3 =  ifft( (1/(std3*std3*N)) * S3L .* conj(S3L));

Cx4  =  ifft( (1/(stdG*std4*N)) * G1L .* conj(S4L));
Cas4 =  ifft( (1/(std4*std4*N)) * S4L .* conj(S4L));
```

```
Crot =  ifft( (1/(stdC*stdG*N)) * GCL .* conj(G1L));


%cutting and pasting to get the edges in the middle
c = (array-1)/2;  %waar we het knippen

Ckx  = [Cx(c:array),Cx(1:c)];
Ckas = [Cas(c:array),Cas(1:c)];

Ckx2  = [Cx2(c:array),Cx2(1:c)];
Ckas2 = [Cas2(c:array),Cas2(1:c)];

Ckx3  = [Cx3(c:array),Cx3(1:c)];
Ckas3 = [Cas3(c:array),Cas3(1:c)];

Ckx4  = [Cx4(c:array),Cx4(1:c)];
Ckas4 = [Cas4(c:array),Cas4(1:c)];

Ckrot = [Crot(c:array),Crot(1:c)];


%----------------star 1

%fitting Gaussian to crosscor. of galaxy with star
start = 421;
eind = 442;
mu1 = 11;
ydata1 = (Ckx(start:eind));
xdata1 = (1:length(ydata1));
[estimates, model] = fitGaussian(xdata1,ydata1,mu1);
[sse, FittedCurve1] = model(estimates);
sigma_xcor = sqrt(1/ (2 * estimates(2)));

%fitting Gaussian to autocor. of star with itself
start = 394;
eind = 412;
mu2 = 10;
ydata2 = (Ckas(start:eind));
xdata2 = (1:length(ydata2));
[estimates, model] = fitGaussian(xdata2,ydata2,mu2);
[sse, FittedCurve2] = model(estimates);
sigma_acorStar = sqrt(1/ (2 * estimates(2)));

%calculating galaxy dispersion
SIGMA_gal1 = sqrt(sigma_xcor^2 - sigma_acorStar^2);
veldisp1(nn) = SIGMA_gal1;


% plot(xdata1, ydata1, '.', xdata1, FittedCurve1, 'r')
% title(['Gaussian fitting to the crosscorrelation']);
% figure
% plot(xdata2, ydata2, '*', xdata2, FittedCurve2, 'r')
% title(['Gaussian fitting to star autocorrelation']);
% figure


%----------------star 2

%fitting Gaussian to crosscor. of galaxy with star
start = 421;
eind = 440;
mu1 = 11;
ydata1 = (Ckx2(start:eind));
xdata1 = (1:length(ydata1));
[estimates, model] = fitGaussian(xdata1,ydata1,mu1);
[sse, FittedCurve1] = model(estimates);
sigma_xcor = sqrt(1/ (2 * estimates(2)));

%fitting Gaussian to autocor. of star with itself
start = 396;
eind = 412;
mu2 = 8;
ydata2 = (Ckas2(start:eind));
xdata2 = (1:length(ydata2));
[estimates, model] = fitGaussian(xdata2,ydata2,mu2);
[sse, FittedCurve2] = model(estimates);
sigma_acorStar = sqrt(1/ (2 * estimates(2)));

%calculating galaxy dispersion
SIGMA_gal2 = sqrt(sigma_xcor^2 - sigma_acorStar^2);
veldisp2(nn) = SIGMA_gal2;


% plot(xdata1, ydata1, '.', xdata1, FittedCurve1, 'r')
```

```
% title(['Gaussian fitting to the crosscorrelation']);
% figure
% plot(xdata2, ydata2, '*', xdata2, FittedCurve2, 'r')
% title(['Gaussian fitting to star autocorrelation']);
% figure


%----------------star 3

%fitting Gaussian to crosscor. of galaxy with star
start = 422;
eind = 442;
mu1 = 11;
ydata1 = (Ckx3(start:eind));
xdata1 = (1:length(ydata1));
[estimates, model] = fitGaussian(xdata1,ydata1,mu1);
[sse, FittedCurve1] = model(estimates);
sigma_xcor = sqrt(1/ (2 * estimates(2)));

%fitting Gaussian to autocor. of star with itself
start = 400;
eind = 408;
mu2 = 4;
ydata2 = (Ckas3(start:eind));
xdata2 = (1:length(ydata2));
[estimates, model] = fitGaussian(xdata2,ydata2,mu2);
[sse, FittedCurve2] = model(estimates);
sigma_acorStar = sqrt(1/ (2 * estimates(2)));

%calculating galaxy dispersion
SIGMA_gal3 = sqrt(sigma_xcor^2 - sigma_acorStar^2);
veldisp3(nn) = SIGMA_gal3;


% plot(xdata1, ydata1, '.', xdata1, FittedCurve1, 'r')
% title(['Gaussian fitting to the crosscorrelation']);
% figure
% plot(xdata2, ydata2, '*', xdata2, FittedCurve2, 'r')
% title(['Gaussian fitting to star autocorrelation']);
% figure


%----------------star 4

%fitting Gaussian to crosscor. of galaxy with star
start = 390;%422;
eind = 402;%442;
mu1 = 5;
ydata1 = (Ckx4(start:eind));
xdata1 = (1:length(ydata1));
[estimates, model] = fitGaussian(xdata1,ydata1,mu1);
[sse, FittedCurve1] = model(estimates);
sigma_xcor = sqrt(1/ (2 * estimates(2)));

%fitting Gaussian to autocor. of star with itself
start = 399;
eind = 410;
mu2 = 5;
ydata2 = (Ckas4(start:eind));
xdata2 = (1:length(ydata2));
[estimates, model] = fitGaussian(xdata2,ydata2,mu2);
[sse, FittedCurve2] = model(estimates);
sigma_acorStar = sqrt(1/ (2 * estimates(2)));

%calculating galaxy dispersion
SIGMA_gal4 = sqrt(sigma_xcor^2 - sigma_acorStar^2);
veldisp4(nn) = SIGMA_gal4;


% plot(xdata1, ydata1, '.', xdata1, FittedCurve1, 'r')
% title(['Gaussian fitting to the crosscorrelation']);
% figure
% plot(xdata2, ydata2, '*', xdata2, FittedCurve2, 'r')
% title(['Gaussian fitting to star autocorrelation']);
% figure


%----------------Galaxy

%fitting Gaussian to crosscor. of different radii to the center of galaxy
start = 394;
eind = 412;
mu4 = 10;
ydata4 = (Ckrot(start:eind));
```

```
xdata4 = (1:length(ydata4));
[estimates, model] = fitGaussian(xdata4,ydata4,mu4);
[sse, FittedCurve4] = model(estimates);
Rot = estimates(1);
rotation(nn) = Rot;

end

pixtr = 60.726;

plot(basi:sonu, pixtr * abs(veldisp1(basi:sonu)), '.');figure
plot(basi:sonu, pixtr * abs(veldisp2(basi:sonu)), '.');figure
plot(basi:sonu, pixtr * abs(veldisp3(basi:sonu)), '.');figure
plot(basi:sonu, pixtr * abs(veldisp4(basi:sonu)), '.');figure
plot(basi:sonu, pixtr * abs(rotation(basi:sonu)), '.')

mean_veldistr1 = mean(abs(veldisp1(basi:sonu))) * pixtr
mean_veldistr2 = mean(abs(veldisp2(basi:sonu))) * pixtr
mean_veldistr3 = mean(abs(veldisp3(basi:sonu))) * pixtr
mean_veldistr4 = mean(abs(veldisp4(basi:sonu))) * pixtr
```

5.1 The Gaussian fit function code

```
function [estimates, model] = fitGaussian(xdata, ydata, mu)
% Call fminsearch with a random starting point.
start_point = rand(1, 2, 3);
model = @expfun;
estimates = fminsearch(model, start_point);
% expfun accepts curve parameters as inputs, and outputs sse,
% the sum of squares error for A * exp(-lambda * xdata) - ydata,
% and the FittedCurve. FMINSEARCH only needs sse, but we want to
% plot the FittedCurve at the end.
    function [sse, FittedCurve] = expfun(params)
        A = params(1);
        lambda = params(2);
        h3 = params(3);
        %h4 = params(4);
        sigm = sqrt(1/(2*lambda));
        Hp3 = 8*((xdata-mu)/sigm).^3 - 12*((xdata-mu)/sigm);
        %Hp4 = 16*((xdata-mu)/sigm).^4 - 48*((xdata-mu)/sigm).^2 + 12;
        FittedCurve = A .* exp(-lambda * (xdata-mu).^2) .* (1 + h3*Hp3); % + h4*Hp4);
        ErrorVector = FittedCurve - ydata;
        sse = sum(ErrorVector .^ 2);
    end
end
```

References

[1974] Simkin, Measurements of Velocity Dispersions and Doppler Shifts from Digitized Optical Spectra, A&A, 31, 129

[1979] Tonry & Davis, A Survey of Galkaxy Redshifts, AJ 84.1511T

[1974] Illingworth & Freeman

[1981] Mihalas & Binney

[2001] Halliday et al. Astron Soc. 326 473-489

[1997] Hjorth & Tanvir, ArXiv:Astro-ph 9701025v1

[1993] Kuijken & Merrifield, A new method for obtaining stellar velocity distributions from absorbtion line spectra: Unresolved Gaussian Decomposition, MNRAS 264 712

[1989] Bender, Unrevealing the kinematics of early type galaxies, A&A 229 441

[1993] Marel, P. van der R, A new method for the identification of non-gaussian line properties in Elliptical Galaxies, ApJ 407 525-539

[1977] Sargent et al., Velocity Dispersions for 13 Galaxies, ApJ 212.326