XUVI

Summer Project, Astronomy Club, IITK

Acknowledgments

We are indebted to the mentors for their guidance and support throughout the project. We would like to thank them for giving us a great learning experience. We are also extremely grateful to the Astronomy Club, IITK and The Science and Technology Council, IITK for giving us this opportunity to work on this project.

Mentors

Adit Jain | aditj20@iitk.ac.in

Nikita Singh | nikitas20@iitk.ac.in

Rajarshi Dutta | rajarshi20@iitk.ac.in

Mentees

K Arnav

Jaya Santhi

Nandan Madhuj

Priyanshu Bhatia

Ritik B Kumar

Shreya Rajak

Contents

1 Introduction

Ever since the dawn of civilisation, we have wondered about our place in the Universe.

Who are we?
What is our place in the Universe?
Are we alone?

These simple questions have the most profound answers. And the hunt for these answers is what drives modern science.The excitement to better understand the world around us, which appears to be governed by few simple laws but which manifests these laws in an infinitude of varieties, keeps the soul of science, ever ignited with enthusiasm. With modern techniques of computation, and advancements in mathematics, new tools have made the quest ever exciting.

About 100 years ago, the American astronomer Edwin Powell Hubble started observing the spectra from distant galaxies. He surprisingly found that most galaxies emitted light as if they were moving away from us - An expanding Universe. Furthermore, this expansion must've been going on for a finite time, or else, we would see no galaxies as they'd have moved away from us long ago. Hence, the expansion must have begun at a finite time in past. So, The Universe must have begun at a finite time in the past.

This revolutionary discovery began simply by observing the light that some objects were emitting. And so, the importance of light can not be overemphasized.

In the present project, we look into the basics of spectroscopy, and computational methods for astronomical data analysis. We understand the principles behind a spectroscope and implement those on a small model and use Python and its libraries for light curve analysis. We implement algorithms for detecting exoplanets and analyse solar activity (i.e detect solar flares) from the XSM data of the Chandrayaan-2 orbiter.

All the codes used and discussed can be found at github.com/astroclubiitk/XUVI

2 Python and it's Libraries

2.1 Introduction

Python is an easy to use, general purpose programming language. It's syntax is similar to English - which makes it easy to read and write. It has several libraries - developed for particular tasks. For example, NumPy is a Python library known for its high-level mathematical functions and speed.

For handling astronomical data in this project, we used NumPy, Pandas, Matplotlib and SciPy. We also learned about AstroPy which is extensively used for larger projects, but for our purpose, AstroPy is slow and unnecessary.

2.2 NumPy

NumPy is a Python library which facilitates working with large arrays and multi-dimensional matrices. Other than several functions for matrices, NumPy offers several other mathematical functions, random number generators, linear algebra tools, Fourier Transform and much more.

NumPy has special functions to create diagonal, identity or zero matrices. It can also generate a matrix with random entries.

Note that in our code snippets, numpy is written as np.

- NumPy arrays follow zero based indexing. This means that the indexing starts at 0 and ends at $n - 1$ for nth element of an array.
- Addition(+) and Subtraction(-) arithmetic operations are defined over arrays. These follow element wise addition/subtraction.
- Surprisingly, Multiplication(*) and Division(/) are also are defined element wise.
- For matrix multiplication, we use the dot operator: $np.dot(x, y)$.
- NumPy has several useful mathematical functions like Trigonometric, Hyperbolic, Logarithmic etc. It also has useful tools like Convolutions of 1D sequences and the Fast Fourier transform.

2.2.1 Fourier Transform

The Fourier Transform is a method for expressing any function as a **sum of periodic components** (sines and cosines of different frequencies). Theoretically, any (smooth) periodic function can be expressed as a linear combination of finite number of sinusoidal functions. A non-periodic function(or a continuous but non-differentiable) too can be expressed as a linear combination of periodic function. However, in this case we have to add an infinite number of sinusoidal functions. Though we can always approximate by adding a large variety, there's always a lack of accuracy.

A technique called Reverse Fourier Transform can be used to return to the original function from its Fourier Transform. This is especially useful for **noise reduction** in a signal.

When both function and its Fourier Transform, are replaced by their discrete counterparts, we have what is called a Discrete Fourier Transform (DFT). DFT has become a mainstay for numerical computing nowadays partly because of a very fast algorithm for computing it, called the Fast Fourier Transform (**FFT**).

NumPy has functions related to FFT under the np.fft module. An example for using the FFT is as follows:

```
t = np.arange(256)
y = np.sin(t)
sp = np.fft.fft(y)

freq = np.fft. fftfreq(t.shape[0])
```

The above code performs Fourier Transform on $y = sin(t)$. Plotting a graph of $freq$ v/s $sp.real$, we get:

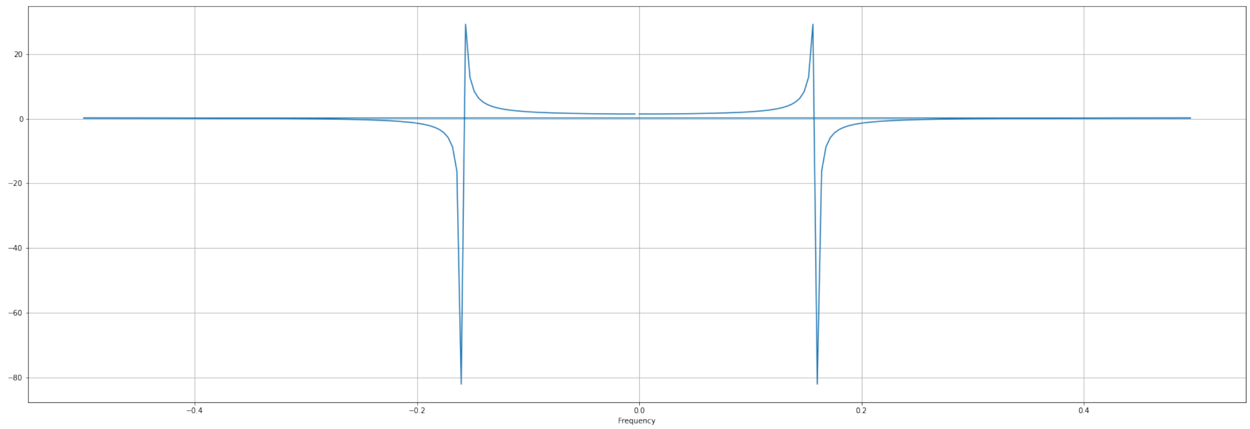


Figure 1: Fourier transform of sin(t) gives peak(dips) near $\frac{1}{2\pi}$.

The above plot shows that most of the frequencies are absent in our function. However, we see sudden change in behaviour at around frequency $= \pm 0.159$

This corresponds to the frequency of our function sin(t).

$$T = 2\pi \Rightarrow f = \frac{1}{T} = \frac{1}{2\pi} \approx 0.159 \tag{1}$$

Note that the frequency is both positive and negative in the plot because We may as well take $T = -2\pi$

Hence, the Fourier Transform of a function from time domain to frequency domain gives us the different frequencies present in that function. In the present case, only 1 frequency ($\frac{1}{2\pi}$) was present.

2.2.2 Convolution

The convolution of two functions $f(t)$ and $g(t)$ is denoted by $(f * g)(t)$

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau)\, d\tau = \int_{-\infty}^{\infty} g(\tau) \cdot f(t - \tau)\, d\tau = (g * f)(t) \tag{2}$$

NumPy offers convolve function to get the convolution of 2 1D arrays. Convolution is much similar to weighted mean of one function w.r.t the other. This method may also be used for noise reduction. For example, 5 points may be replaced by 1 point by taking some appropriately weighted mean. This reduces the number of points and hence the noise.

Some Complex functions cannot be expressed except as a convolution of two functions. For example, an Elementary Flare Profile, is a function obtained by convolution of two function. We discuss these aspects of convolution during data processing and analysis in subsequent sections.

2.3 Pandas

Pandas stands for "Python Data Analysis Library". This library simplifies the data pre-processing steps and facilitates working with data. A key feture of Pandas is that it converts data in files such as csv, excel sheets etc into a tabular data frame. This makes our life much much simpler. From here onwards, pandas will be referred to as pd. The following code reads a csv file and stores it into a data frame named data.

```
data  = pd.read_csv("data.csv")
```

The data gets converted into a tabular form which can be accessed as columns of data. The below code extracts useful information about the DataFrame.

```
print(data.columns)
```

The output:

```
Index(['Index', ' Height(Inches)"', ' "Weight(Pounds)"'], dtype='object')
```

The DataFrame stores 3 columns corresponding to 'Index',' Height(Inches)' and "Weight(Pounds)." The following code gives a scatter plot between column[2] and column[0] of the data-frame (Weight vs Index) and column[1] and column[0] (Height v/s Index)

```
data.plot.scatter(x = data.columns[0], y = data.columns[2]);
data[data.columns[1]].plot();
```

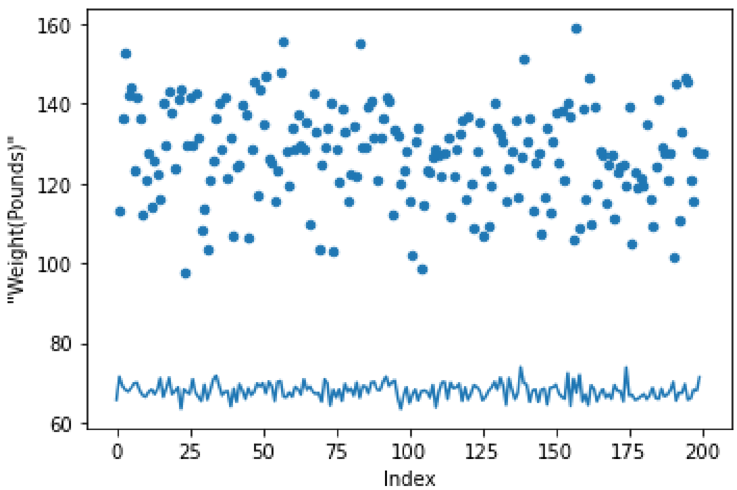


Figure 2: Plot obtained from the data

2.4 SciPy

SciPy is one of the most popular libraries available in Python. It is especially useful for Scientific Computing. It has tools for working with arrays, inbuilt interpolation implementing algorithms, algorithms for solving differential equation and eigenvalue problems. It also provides several complex data-structures.

2.4.1 Curve Fitting

SciPy has a function $curve\_fit$ available under $scipy.optimize$. It uses least squares error to fit a curve to a data. $curve\_fit$ finds all important coefficients if we know the general form of the function.

Consider the function $y = a \cdot e^{-x} + b \cdot sin(x)$

```
def func1(x):
    return 3*np.exp(-x) + 2*np.sin(x)

x = np.linspace(0, 100, 1000)
y = func1(x)
```

Suppose we are given x and y as above. But we do not know the relation between them.

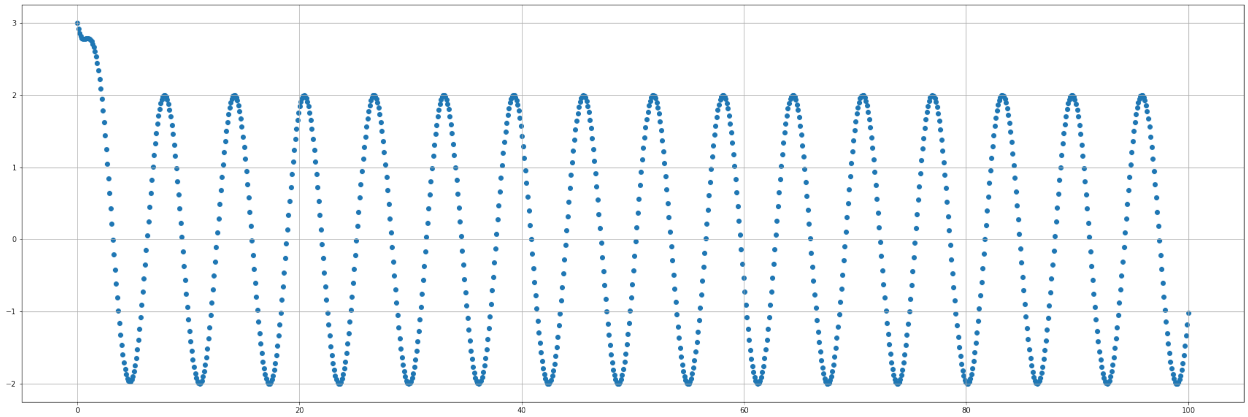


Figure 3: Initial Data with unknown relation between x and y.

We make a calculated guess that y is a linear combination of e^{-x} and $sin(x)$
We can follow the following scheme to find an approximate relation between x and y.

```
from scipy.optimize import curve_fit

def func2(x, a, b):
    return a*np.exp(-x) + b*np.sin(x)

popt, pcov = curve_fit(func2, x, y, p0=(1,1))
a, b = popt
y_scipy = func2(x, a, b)

print("The parameters of the function are {} and {}". format(a, b))
```

The above code works as follows: $curve\_fit$ function returns to items $popt$ and $pcov$. $popt$ contains the optimum choice of variables for fitting the curve over the data. $pcov$ the covariance matrix, whose determinant is the measure of accuracy of the fitting.

After obtaining the optimum value of a and b, we define a new variable, y_scipy and plot it to compare with the original data. For our case, the curve fits exactly, but this is generally not true. The output of above code is:

```
The parameters of the function are 3.0 and 2.0
```

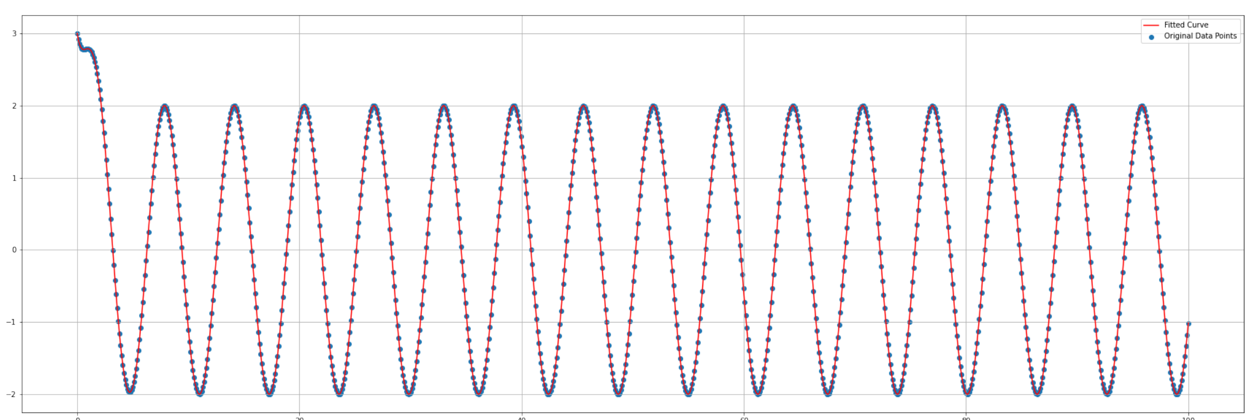


Figure 4: Raw Data and The curve obtained by curve_fit.

2.4.2 Interpolation

Interpolation is much used for astronomical data analysis. It is a technique used for constructing new data in place of missing data.Interpolation helps to estimate functional value at intermediate points by creating an appropriate function passing through the known data points. Hence it helps in joining discontinuous functions.

Scipy offers various interpolation techniques like linear, cubic, spline, bi-quadratic etc. We take an example of interpolation using the function: $y = x^3cos(x) + sin(x^2) + e^{-x^3}$.

```
x = np.linspace(0, 50, 10)
y = x**3*np.cos(x) + np.sin(x**2) + np.exp(-x**3)
```

Suppose, the above data is real and we obtain it from some reliable source. We do not know the exact relation between the two variables x and y.

Clearly our data is scarce, and we are unable to guess which type of function may be used for curve fitting. So we go for interpolation, to get a function that joins these points in the best possible way and hence gives us the expected value at missing points.

```
f = interp1d(x, y, kind='cubic')          #Using cubic interpolation
x_dense = np.linspace(0, 50, 100)         #increase the number of data points
y_interpolated = f(x_dense)               #get the value of y at new data points
```

Plotting the original and interpolated data,

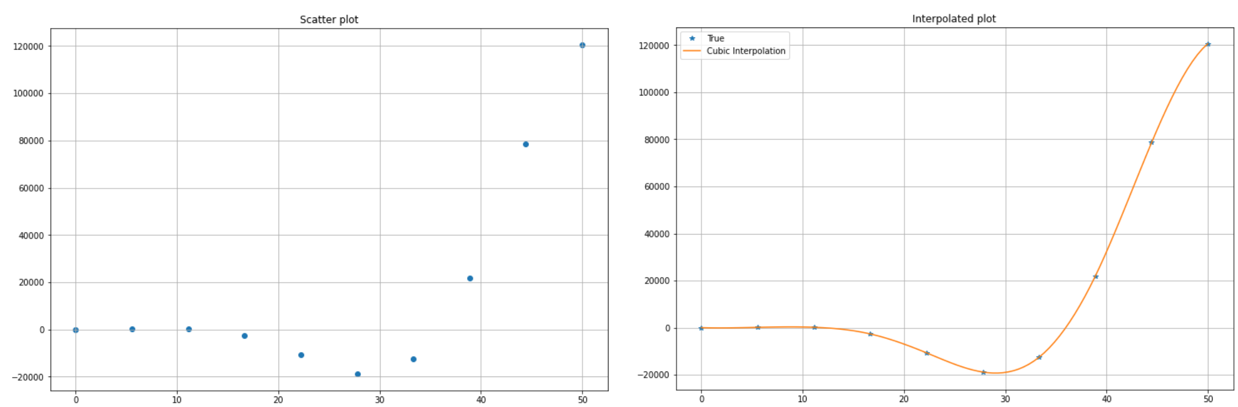


Figure 5: Initial scattered data and Final interpolated data.

2.4.3 Minimisation

Scipy’s minimize method helps minimizing in both constrained and non-constrained problems. The function may involve one or more independent variable. An example for minimisation is as follows:

```
def fun(x):
    return -8 + x**2*np.sin(x) + x*np.cos(x)

x = np.linspace(-6, 6, 100)
```

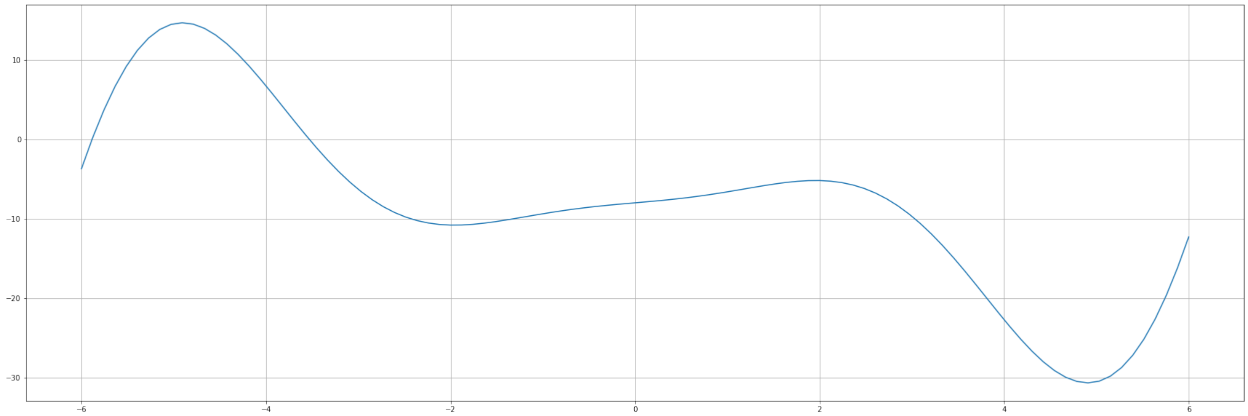


Figure 6: Plot of function fun(x)

Now, starting at any given x value, we can find the nearest point of local minima as follows:

```
from scipy.optimize import minimize
res = minimize(fun, 4)
print(res)
```

Starting at x = 4, the above code finds the nearest local minima. The output is as follows:

```
   fun  : -30.675199539 40562
hess_inv: array([[0.03785082]])
     jac: array([-1.66893005e-06])
 message: 'Optimization terminated successfully.'
    nfev: 12
     nit: 5
    njev: 6
  status: 0
 success: True
       x: array([4.90565959])
```

The nearest minima occurs at around x = 4.905. The corresponding minima is -30.6752

3 Spectroscopy

3.1 What is Spectroscopy?

The technique of examining the electromagnetic radiation that comes from objects (like stars, nebulae etc) and use it to predict the object's composition, temperature, density, motion etc is called Spectroscopy.

Spectroscopy is used extensively in all Fields of science. Be it Biology, Chemistry or Physics. For example, Radio-frequency spectroscopy of nuclei in magnetic field has been employed in the medical technique known as magnetic resonance imaging (MRI) to visualise the internal composition of the body with unprecedented resolution.

With continuous spectrum, we identify elements by the presence of dark bands, and it can also tell scientists how hot an object is: As the temperature goes up, the spectrum has increasing amounts of green, blue and violet colors. (This is why a piece of iron on warming looks Red as Blue frequencies are scarce, but on increasing temperature, it glows white due to presence of all frequency of light.)

In addition, different cool gases will absorb different wavelengths of light and generate a signature spectrum with dark lines at characteristic places. Because of this, we can determine the composition of gases by observing light that has passed through them.

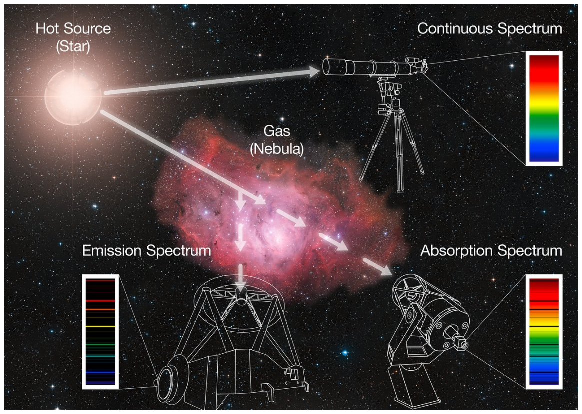


Figure 7: Predicting the composition of Nebulae.

3.2 Astronomical Spectroscopy

Astronomical spectroscopy is used to measure five major bands of radiation in the electromagnetic spectrum:

- Radio
- X-rays
- Ultraviolet
- Visible
- Infrared

It is used to derive many properties of stars and galaxies, such as their composition and movement.

Astronomical Spectroscopy began with Issac Newtons initial observations of the light of the sun, dispersed by a prism. However, when the spectrum was closely examined, the rainbow was found to be interrupted by hundreds of tiny dark lines (called Fraunhofer lines). These lines showed that some wavelengths are being absorbed by gases in the outer atmosphere of the Sun, and from this, we can determine which elements are in the Suns atmosphere.

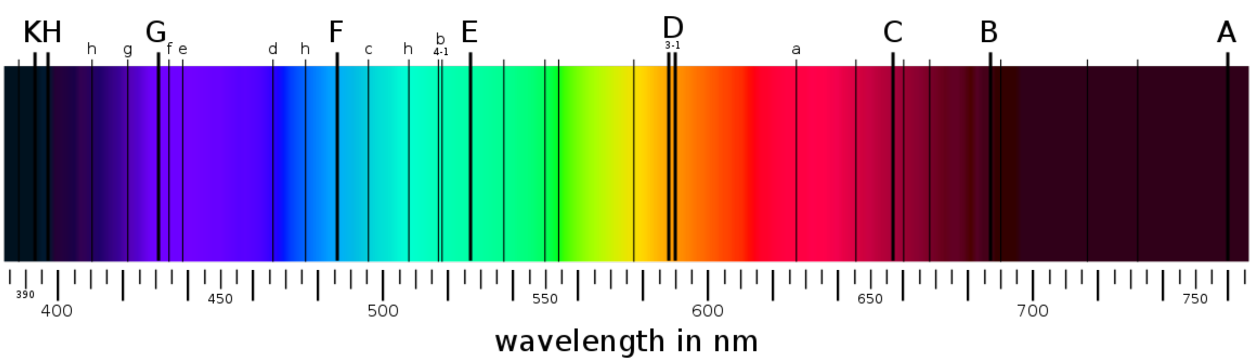


Figure 8: The Fraunhofer lines

3.3 Working of a spectroscope

Astronomical spectroscopes contains these essential elements:

- Slit on to which the light from the telescope would be focused
- Collimator, which would take the diverging light beam and turn it into parallel light
- Disperser (usually a reflection grating)
- Camera that would then focus the spectrum onto the detector

In the case of an objective-prism spectroscopy, the star itself acts as a slit and the Universe for a collimator

The slit sits in the focal plane, and usually has an adjustable width w. The image of the star(or galaxy or other object of interest) is focused onto the slit. The diverging beam continues to the collimator, which has focal length Lcoll. The f-ratio of the collimator (its focal length divided by its diameter) must match that of the telescope beam, and hence its diameter has to be larger the further away it is from the slit, as the light from a point source should just fill the collimator. The collimator is usually an o-axis paraboloid, so that it both turns the light parallel and redirects the light toward the disperser. In most astronomical spectrographs the disperser is a grating, and is ruled with a certain number of grooves per mm,usually of order 1001000. If one were to place ones eye near where the camera is shown in the image present in two slides before, the wavelength of light seen would depend upon exactly what angle i the grating was set at relative to the incoming beam (the angle of incidence), and the angle the eye made with the normal to the grating (the angle of diffraction). How much one has to move ones head by in order to change wavelengths by a certain amount is called the dispersion,and generally speaking the greater the projected number of grooves/mm (i.e., as seen along the light path), the higher the dispersion, all other things being equal. The relationship governing all of this is called the grating equation and is given as:

$$m\lambda = \sigma(\sin i + \sin\theta) \tag{3}$$

Modern spectroscopes often replace the prism with narrow slits called diffraction grating. The slits spread the light into different wavelengths by different amounts, which makes it possible to measure the wavelengths.

3.4 DIY Spectroscope

Materials Required:

- Cardboard Box and Tube
- CD
- Stationery (Pencils, Scissors, Rulers)
- Adhesives

We started off by making a small slit in front of the box from where the light would go inside the box. A CD is placed using adhesives such that the light coming through the slit falls on it at a particular incident angle and not normally. An elliptical hole is made at the adjacent face of the slit so that the tube can be inserted there to view the spectrum.The light entering through the slit gets reflected from the CD to the other adjacent wall which can be viewed from the cardboard tube. The CD is used to split the light entering into its components. Hence, the spectroscope is ready to use.

Figure 7: A small model Spectroscope

Figure 8: Spectrum as seen through the tube.

Figure 9: Materials used for the model

4 Light Curves

Light Curves are graphs that show the brightness of an object over a period of time. For studying objects which periodically emit Electromagnetic Pulses or which periodically become bright and dim, light curves are simple yet powerful tool.

An astronomical event can be classified by observing its light curve and comparing it with standard known light curves.

For example, the figure given below is a light curve of an eclipsing binary star system.

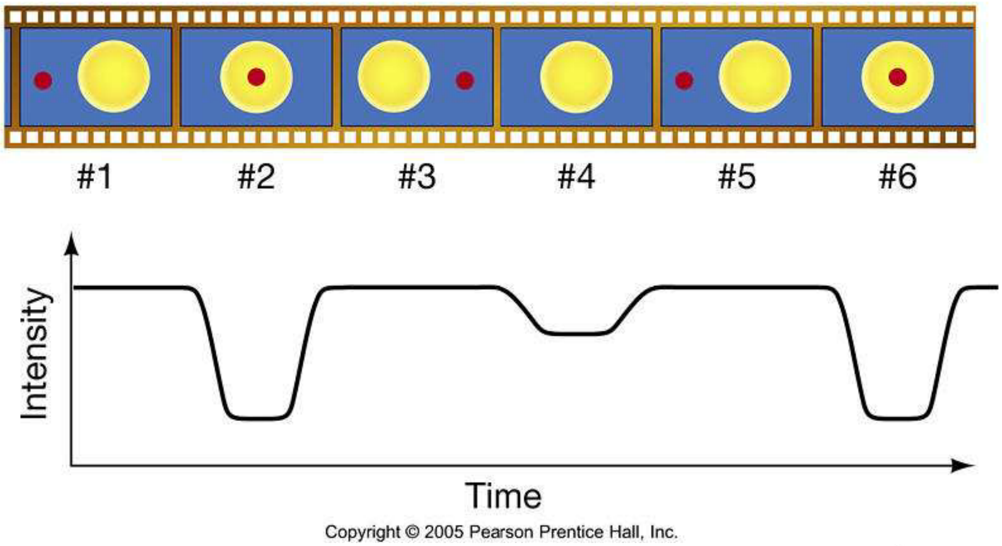


Figure 10: Eclipsing Binary Star Light Curve

The figure on right shows a Light Curve for stellar Death.(A supernova explosion.) Several standard light curves are used for classifying astronomical events.

In subsequent sections, we use light curves for exoplanet detection (transit method) and for solar flare analysis.

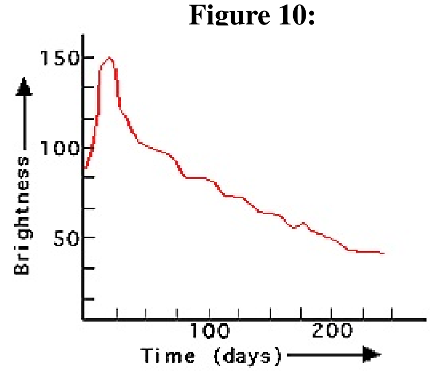


Figure 11: Light Curve for a Supernova explosion

5 Exoplanets: Worlds Beyond Our Solar System

Exoplanets are planets that orbit a star other than our sun. The prefix exo comes from the Greek and means outside; these worlds are far, far outside our own solar system. Astronomers have confirmed more than 4,000 exoplanets orbiting distant stars, with at least 1,000 more more awaiting confirmation.

5.1 Why didn't we see them before?

It's because exoplanets are so far away, several light-years away at their closest. And unlike stars Exoplanets do not shine with their own light. Like our own Earth, they shine only with light reflected from their local stars. In contrast to their stars, exoplanets are exceedingly dim; even the largest are drowned in the light of their vastly brighter stars.

5.2 Detection Techniques

5.2.1 *Direct Imaging*

This method involves directly capturing the exoplanets by removing the overwhelming glare of the stars they orbit.

On a bright day, you might use a pair of sunglasses, or a car's sun visor, or maybe just your hand to block the glare of the sun so that you can see other things. This is the same principle behind the instruments designed to directly image exoplanets. They use various techniques to block out the light of stars that might have planets orbiting them. Once the glare of the star is reduced, they can get a better look at objects around the star that might be exoplanets

The major problem astronomers face in trying to directly image exoplanets is that the stars they orbit are millions of times brighter than their planets. Any light reflected off of the planet or heat radiation from the planet itself is drowned out by the massive amounts of radiation coming from its host star. So we move on to the next method.

5.2.2 *Radial Velocity*

Orbiting planets cause star to wobble in space, changing the colour of the light astronomers observe.

Even though the planet is small, it still has some gravitational force. It still has an effect on its host star, even if that effect is much less pronounced than the one the star has on the planet. As you might imagine, the bigger the planet, the bigger the effect it has on its star. Small planets, like Earth, make their stars only wobble a tiny bit. Bigger planets, like Jupiter, have a much stronger effect.

A star's 'wobble' can tell us if a star has planets, how many there are, and how big they are.

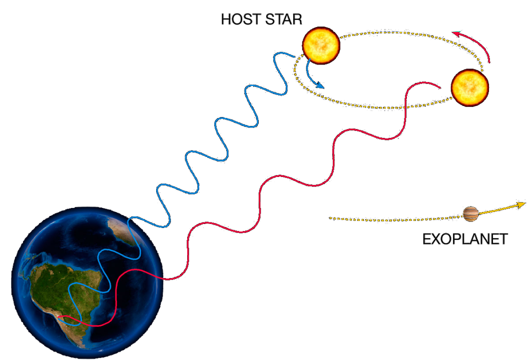


Figure 12: Doppler shift

Wobbling stars are great for finding exoplanets, but how do we see the wobbling stars?

The method used is one called 'Doppler shift' where astronomers experience change in colours while observing the star .The reason is because when an object that emits energy (like an ambulance speaker or a massive, burning star) moves closer to you, the waves bunch up and squish together. And when the object is moving away, the waves stretch out.

Those changes in the wavelength change how we perceive the energy that we're seeing or listening to. As sound waves scrunch together, they sound higher in pitch. And when visible light waves scrunch together, they look more blue in color.

When sound waves stretch out, they sound lower in pitch. And when visible light waves stretch out, they make an object look more reddish.

The radial velocity method was one of the first successful ways to find exoplanets, and continues to be one of the most productive methods. Often, this method will be used to confirm planets found with other methods - an extra step that can prove a planet exists.

5.2.3 Astrometry

The orbit of a planet can cause a star to wobble around in space in relation to nearby stars in the sky.

Doppler shifts aren't the only way astronomers can find stars that are wobbling due to the gravity of their planets. The wobble can also be visible as changes in the star's apparent position in the sky.

If the target star has moved in relation to the other stars, astronomers can analyze that movement for signs of exoplanets.

Given below is the Radial Velocity of star(m/s) vs time(s) obtained from observation:

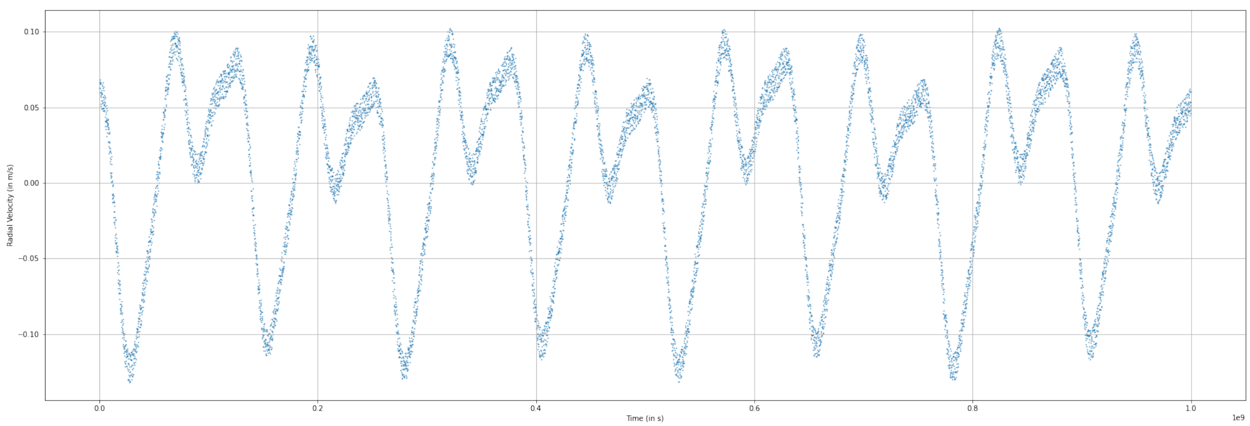


Figure 13: Raw Observational Data

The Fourier Transform of the data looks as follows:

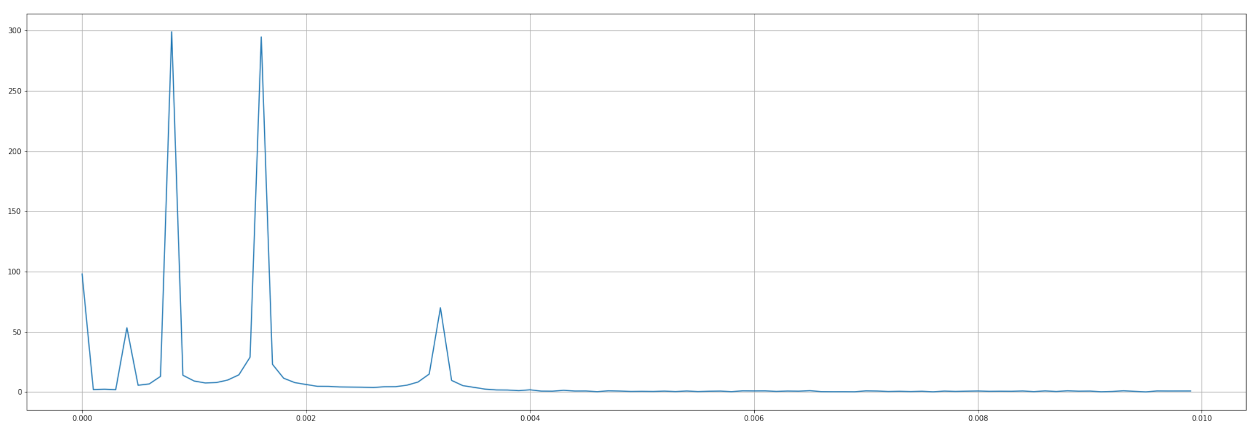


Figure 14: Fourier Transform of Observed Data

The FT reveals presence of 4 different frequencies in the motion of the star. This strongly suggests the presence of 4 different massive objects orbiting the star.

We correct the Fourier Transform, by removing the peak at frequency = 0, and also remove frequencies other than the 4 major ones. This is because these smaller frequencies are irrelevant, and most likely caused by noise in the data.

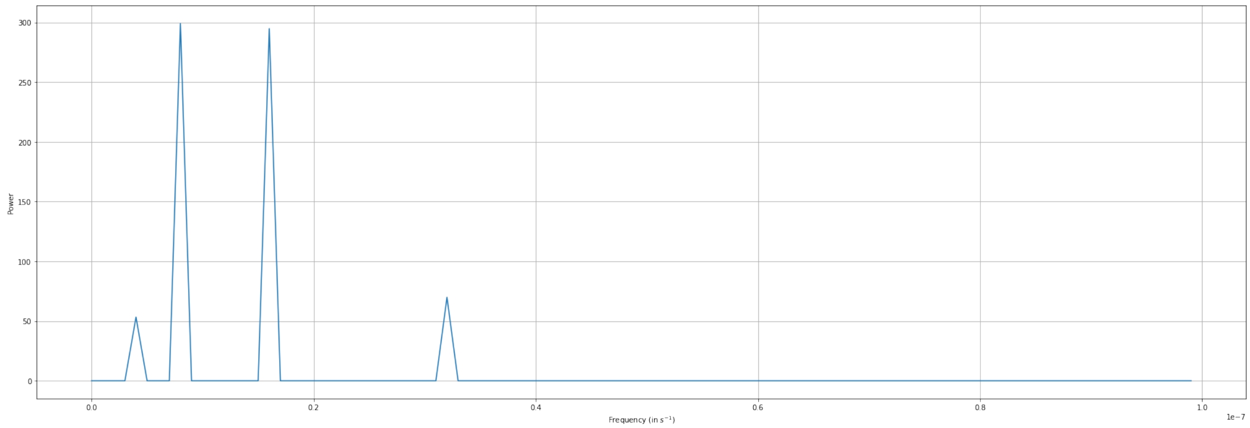


Figure 15: Corrected Fourier transform

Taking the Inverse Fourier Transform of the corrected data on frequency domain, we obtain the below plot in which the noise reduction is clearly visible.

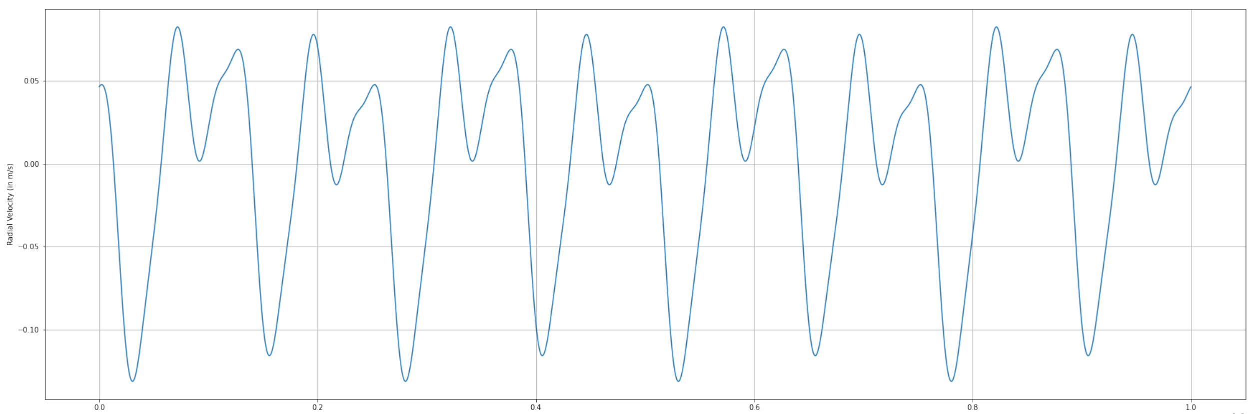


Figure 16: Smoothed velocity graph

To obtain information about individual planets, we separate out the different frequencies from the FT.

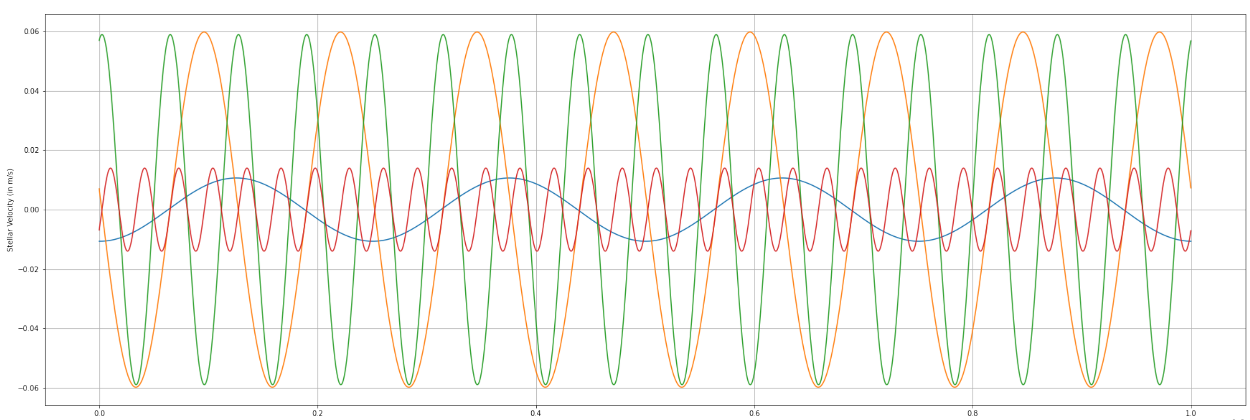


Figure 17: Motion due to individual planets separated into different frequencies.

We obtain the time-period of revolution for each planet from the above data. If, by other means, we know the Mass and Radius of the Star (using other techniques like spectroscopy etc), we can find important information about the planets.

For the given problem, we have taken mass of the star as 10^{30} Kg (Half of Sun). We directly get the **Time_period** and **angular frequency** from the above data. For simplicity, we assume circular orbits of star(radius = r) and planet(radius = R). We expect $R >> r$. The centrifugal force on the planet is given by the mutual gravitational pull:

$$m \cdot \omega^2 \cdot R = \frac{G \cdot M \cdot m}{(R+r)^2} \tag{4}$$

Under the approximation that r<<R,

$$\frac{G \cdot M}{\omega^2} \approx R^3 \tag{5}$$

Hence, we have the Radius of Revolution of the planet (R). We now use equation of centre of mass:

$$m \cdot R = M \cdot r \tag{6}$$

We know M and R from our initial problem and calculations respectively. Using the equation of SHM, we obtain the value of r:

$$R = \frac{V_{max}}{\omega} \tag{7}$$

Both V_{max} and ω can be obtained data of Figure 12. So, we can obtain the mass of the planet. Hence, we have all the relevant data about the system containing 4 exoplanets.

The final output is shown below:

```
exoplanet number: #1
mass of exoplanet:      0.15 earth masses
radius of revolution:            3.172 AU
revolution period:      7.927 earth years

exoplanet number: #2
mass of exoplanet:      0.66 earth masses
radius of revolution:            1.998 AU
revolution period:      3.964 earth years

exoplanet number: #3
mass of exoplanet:      0.52 earth masses
radius of revolution:            1.259 AU
revolution period:      1.982 earth years

exoplanet number: #4
mass of exoplanet:      0.09 earth masses
radius of revolution:            0.793 AU
revolution period:      0.991 earth years
```

5.2.4 Transit Photometry

When a planet passes directly between its star and an observer, it dims the star's light by a measurable amount

When a planet passes directly between an observer and the star it orbits, it blocks some of that star's light. For a brief period of time, that star actually gets dimmer. It's a tiny change, but it's enough to clue astronomers into the presence of an exoplanet around a distant star.

Consider the following data for the variations in relative brightness of a distant star with time.

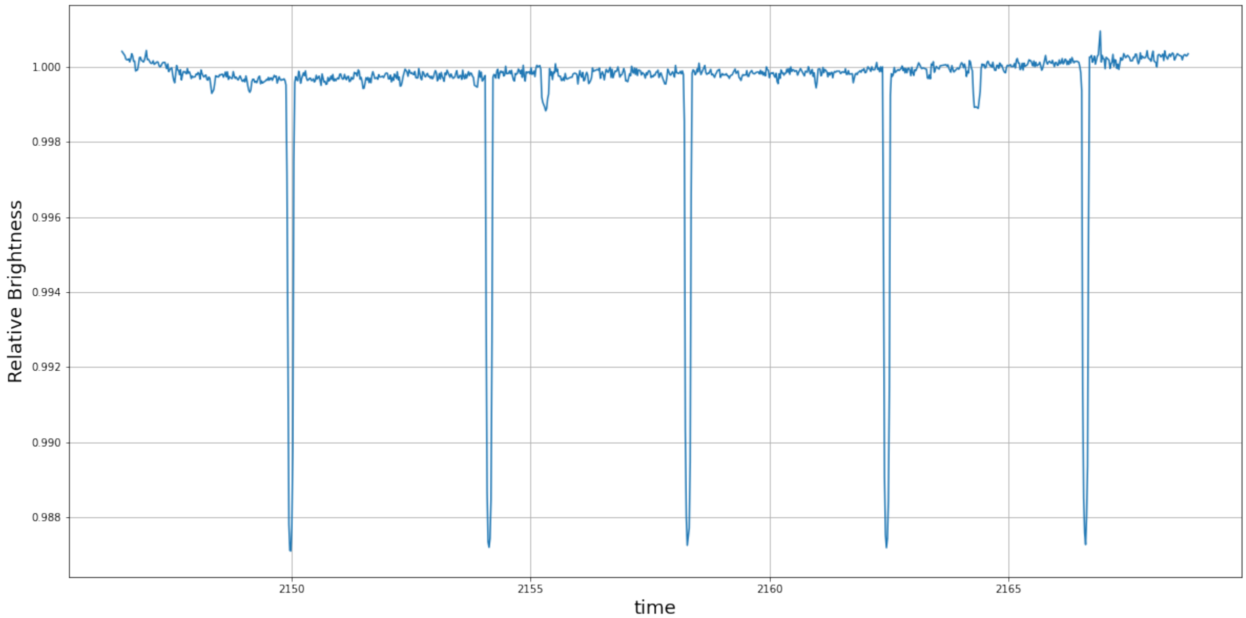


Figure 18: Periodic dips in the brightness of a star

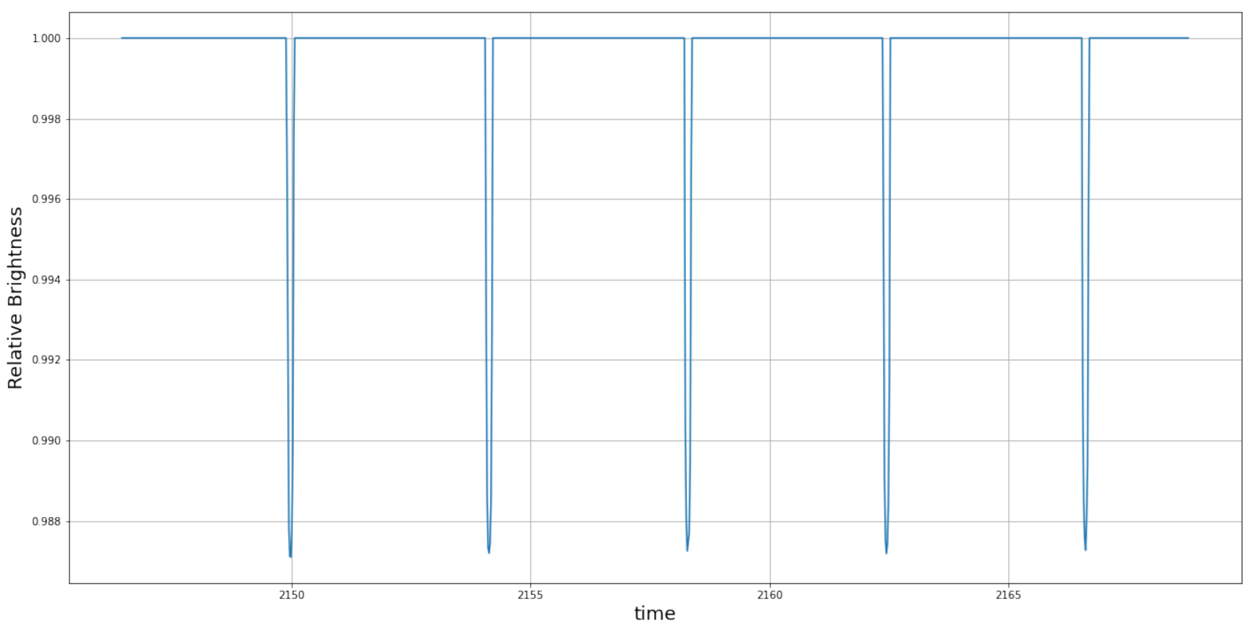


Figure 19: Relative Brightness data after reducing noise

The periodic change in the brightness of this star suggests presence of some planet orbiting it such that it periodically comes between us (the observers) and the the star.

The data directly gives us the time period of revolution of the exoplanet. Since, there's a dip whenever the planet passes in front of the star, this can happen only after one full orbit. Hence, the time period is given by the time difference between consecutive dips.

```
The average time period was found to be : 4.16 days.
```

The star and the planet are point sized sources of light. The dips signify, that the planet is smaller than the star(as it should be). But by how much?

Assuming the full disc of star has radius R and the full disc of the planet has radius r, the brightness is directly related to the area of the source.

Let the Relative Brightness during dips be given by b, then,

$$b = \frac{(\pi \cdot R^2 - \pi \cdot r^2)}{\pi \cdot R^2} \tag{8}$$

Hence, the dip in relative brightness d is given by

$$d = 1 - R' = \frac{r^2}{R^2} \tag{9}$$

Now, if we know the radius of star, we can find the radius of the planet. For the present problem, we used the Radius of Star as $7 \cdot 10^8$

```
The average radius was found to be      : 79190 Kms
```

Lastly, we can directly find the duration of transit from the data as the time difference between start and end of a dip.

```
The average duration of the transit is  : 4.02 Hours
```

6 The Sun

The Sun is the heart of our solar system. It is nearly perfect round and is made up of hot plasma. With continuous nuclear reactions ,energy emissions and magnetic activity, Sun is the most active site in the solar system.

6.1 Sunspots

The Solar surface is a site of immense activity. Sunspots are dark patches present on the photosphere. Their temperature is about 4200K much lower than the 5800K of the surrounding region in the photosphere. Due to this, the photon emission is significantly lower from these regions. Hence the name Sunspots. It is important to note that these spots are only temporary. They are caused due to magnetic activity in the Sun.

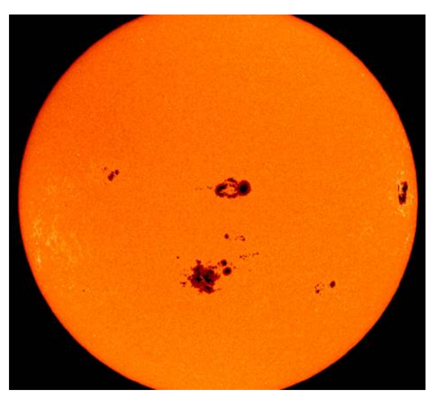


Figure 20: Sunspots

The magnetic field of Sun is not uniform over the surface. If in a region, the magnetic field becomes really high (approx 2500times), than the surrounding regions, then this area becomes a region of high magnetic pressure. This inhibits the flow of matter from inside the surface in these regions. Hence, due to lack of gases, photon emission drops.

It is interesting to note that the Sun undergoes an approx 12 year cycle with respect to its magnetic activity.

6.2 Solar Flares

A solar flare is a sudden outburst of energy during which magnetic energy is converted into kinetic energy of fast moving particles, mass motions and radiation across the entire electromagnetic spectrum.

The energy released may vary from 10^{17} J for micro-flares to 10^{25} J for largest flares.

6.3 Data Capturing

Solar activity can be measured as the number of photons striking the observational area per unit time. The data we're using is obtained from Solar X-Ray Monitor (XSM) aboard the Chandrayaan-2 orbiter and is made available by ISRO via Pradan Website.

6.4 Preprocessing

Now we start working on actual data obtained from XSM aboard the Chandrayaan-2 Orbiter.

6.4.1 Reading the data

```
from astropy.table import Table

table = Table.read("ch2_xsm_20200928_v1_level2.lc")
print(table)
```

We use Table function under *astropy.table* to read the lightcurve (LC) file available to us. We then stored this data into a variable named *table*. Necessary information about the data can then be obtained (by printing).

```
       TIME             RATE      ERROR    FRACEXP
         s             ct / s    counts/s
------------------ --------- --------- -------
118022400.44376001 56.586166 7.5223775     1.0
118022401.44376001  75.02865  8.661908     1.0
118022402.44376001  67.52199  8.217176     1.0
118022403.44376001  81.25248  9.014016     1.0
118022404.44376001 79.532425  8.918096     1.0
118022405.44376001  77.37406  8.796252     1.0
118022406.44376001  77.34502  8.794601     1.0
118022407.44376001  70.27271  8.382882     1.0
118022408.44376001  81.94259  9.052215     1.0
118022409.44376001 75.475586  8.687669     1.0
               ...       ...       ...     ...
118108789.44376001  86.36383  9.293214     1.0
118108790.44376001  78.35937  8.852082     1.0
118108791.44376001  82.79164  9.098991     1.0
118108792.44376001  87.59899  9.359433     1.0
118108793.44376001 86.327354  9.291251     1.0
118108794.44376001  85.11869  9.225979     1.0
118108795.44376001  79.41163  8.911321     1.0
118108796.44376001  81.39757   9.02206     1.0
118108797.44376001 87.616745  9.360382     1.0
118108798.44376001  82.65829 9.0916605     1.0
Length = 84181 rows
```

Since now we know the different columns present in the data, we can obtain relevant information.

```
t1 = table["TIME"]
r1 = table["RATE"]
```

We now plot rate vs time to judge the data qualitatively.

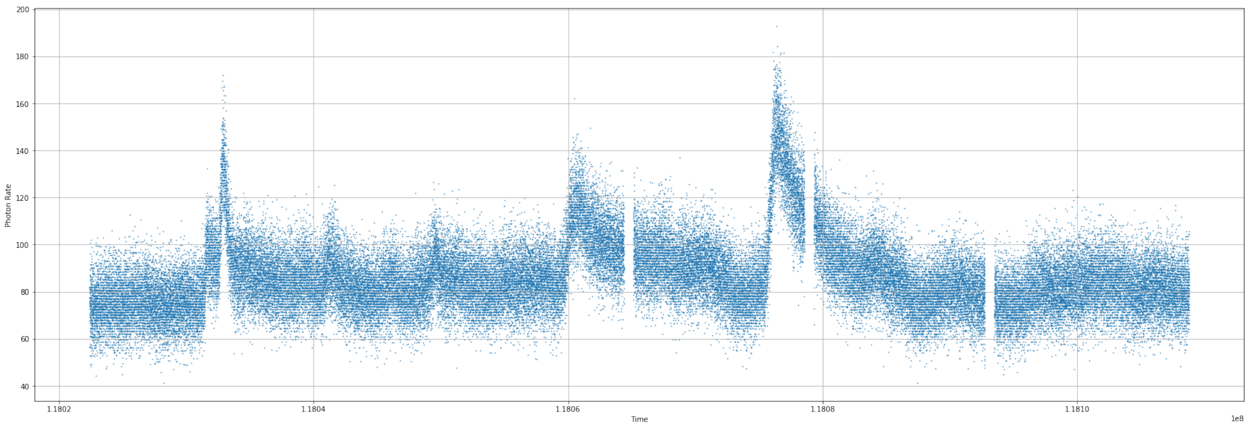


Figure 21: Raw data

Two problems need to be addressed:

- Noise
- Discontinuity

6.4.2 Separating continuous segments

We can reduce noise on continuous data. So, first we try to separate out different continuous segments. From our data, we note that most observations separated by 1 second. We use this fact to obtain the discontinuities as the points where consecutive observations have a lag of more than 1 second.

The data from this file has 4 continuous segments. This can be verified from the plot obtained from Figure 10 also which has 3 discontinuities.

Hence, we separate out all continuous segments by forming 4 arrays, each storing all points between the start and end of each individual segment. Hence, the continuous segments are separated.

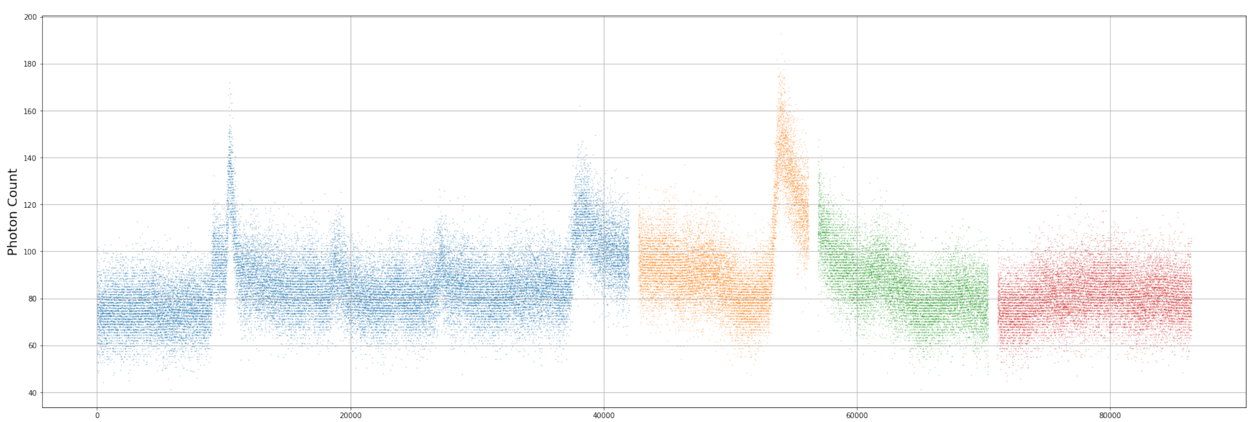


Figure 22: Data after separating out continuous segments

6.4.3 Noise Reduction

The general scheme was to replace n continuous data points with a single point which was their mean. And then, moving some distance ahead (say $\frac{n}{k}$) and repeating.

Hence, we are decreasing the number of data points to reduce noise and shifting by small amounts ensures smoothness.

We plot the function to see the results.

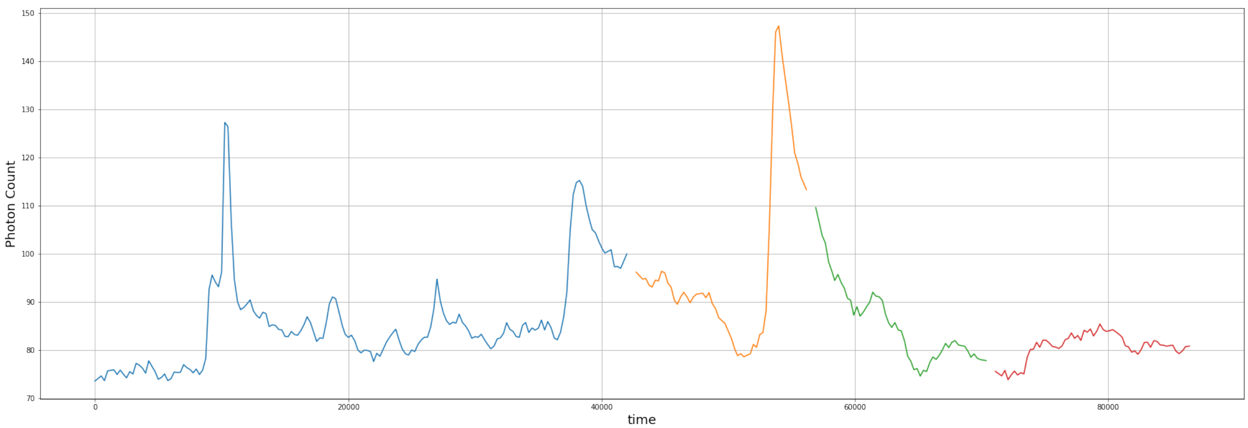


Figure 23: Smoothened Data

We then re-combined the data. After noise reduction, and recollecting the continuous segments, the state of the data is shown below:

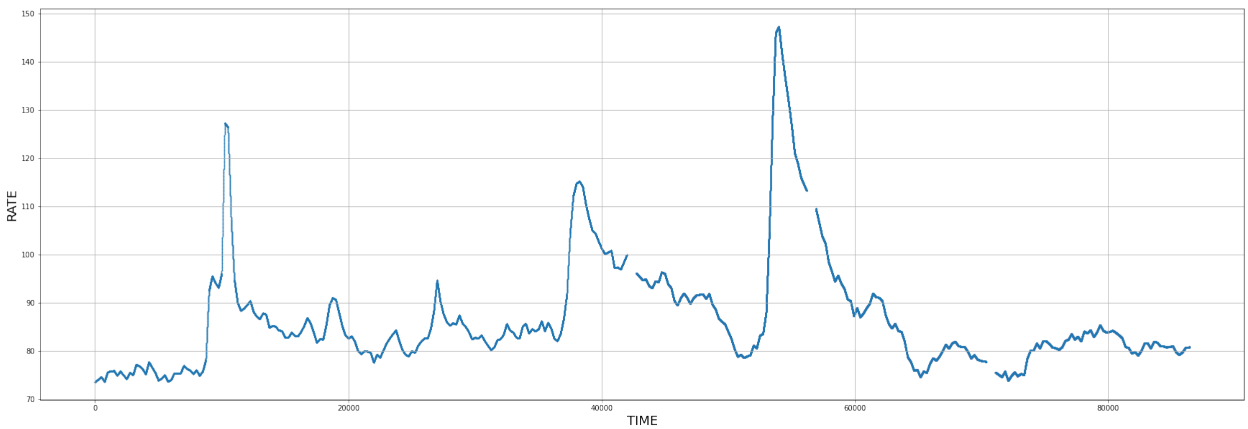


Figure 24: Data after noise reduction

6.4.4 Interpolation

We then removed the discontinuities using linear interpolation. A plot between final rate and final time is shown below:

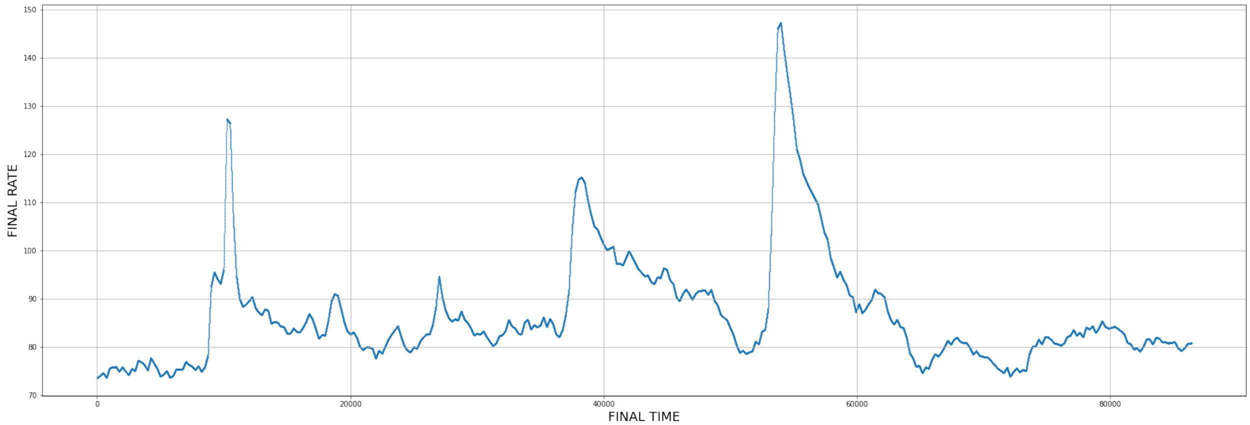


Figure 25: Final Pre-Processed data

6.5 Detection

Step 1: We start by getting all the maxima. These can be the peak of a possible flare. The corresponding start of the flare is chosen to be a minimum just before a given maximum.

Step 2: The first level filtering is done based on the slope of ascent of the Solar Flare. A solar flare has sudden rise and slow decay, we expect the slope of ascent to be bigger certain threshold to claim that the photon count did increase suddenly.

Step 3: Suppose and due to noise in the data, there's a minima during the ascent phase, in that case, our algorithm detects two flares instead of one. Hence, Another filtering is required such that, if the time of start of two flares is too close, and the minima is small, we count the flare as one.

Step 4: Finally, we expect a solar flare to be powerful, only the shape doesn't matter, the energy does too. Hence, we take the mean of peak count rate of all remaining Flares, and filter off those which are lower in energy than this mean. These are simply too weak to cause any effect and are insignificant for use in scientific studies.

Step 5: For completion, we also choose an appropriate minima, just after the peak. This is where the solar flare ends and possible another begins.

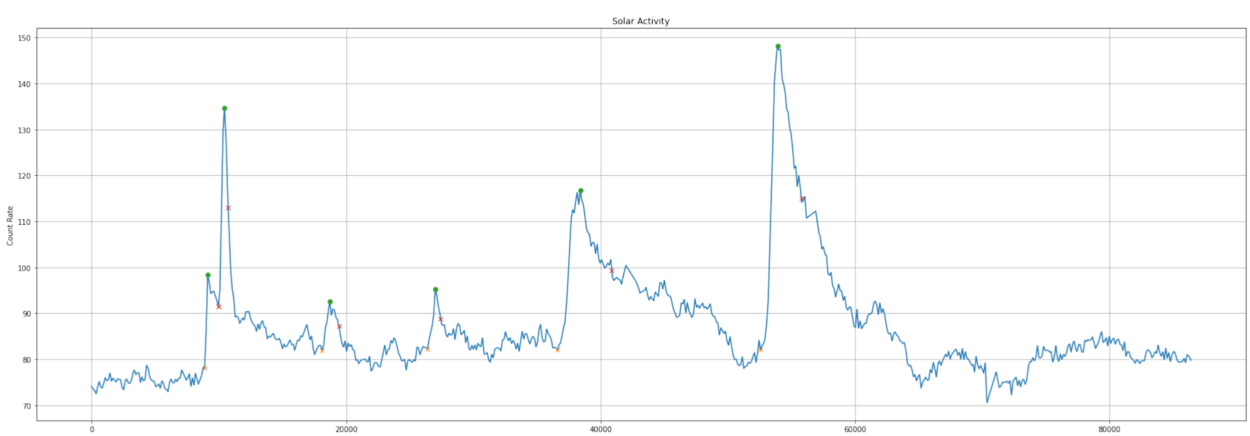


Figure 26: Final Data after detecting peaks, starts and ends of each Solar Flare

6.6 Modelling

Consider the following two functions:

1. Gaussian Function

$$f_1(t) = A \cdot e^{\frac{-(t-B)^2}{C^2}} \tag{10}$$

2. Exponential Function

$$f_2(t) = e^{-Dt} \tag{11}$$

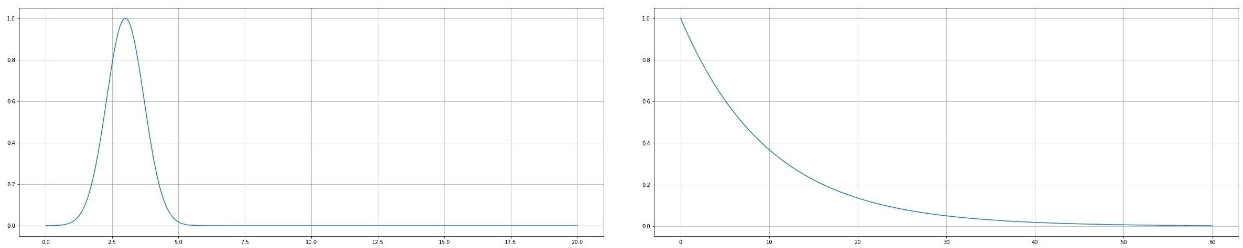


Figure 27: Functions $f_1(t)$ and $f_2(t)$

6.6.1 Convolution of f_1 and f_2

A solar flare can be modelled as the convolution of the gaussian and the exponential function given above. Following method may be used to calculate the convolution of the two functions:

```
def f1(t,a,b,c):
    tmp = - ((t-b)/c)**2
    return a*np.exp(tmp)

def f2(t,d):
    x = np.exp((-1)*d*t)
    return x

t1 = np.linspace(0,20,2100)
x1 = f1(t1,1,0.1,2)
t2 = np.linspace(0,60,1600)
x2 = f2 (t2,0.1)
x = np.convolve(x1,x2)
```

The output for which is as follows:

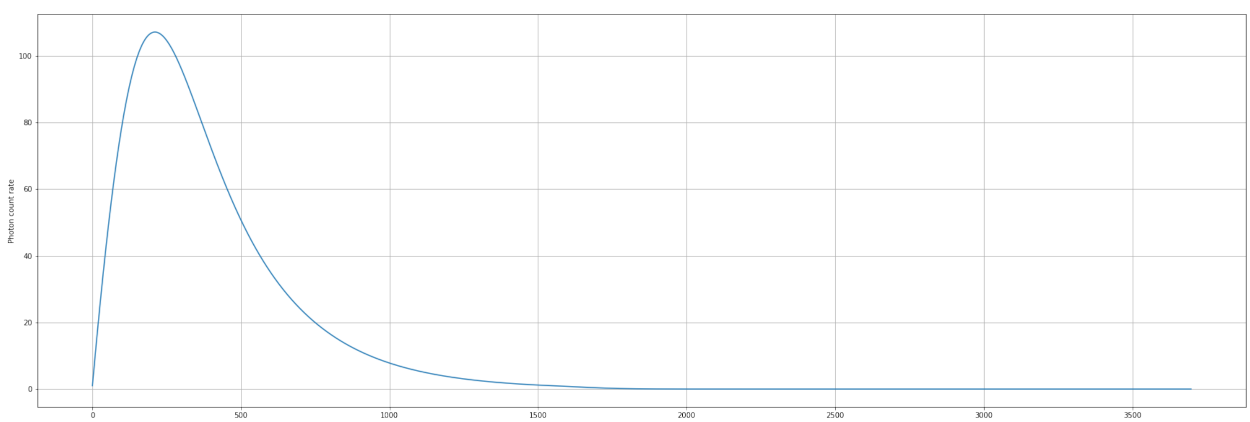


Figure 28: Elementary Flare Profile (EFP)

Mathematically,

$$EFP(t) = \frac{1}{2}\sqrt{\pi}ACe^{D(B-t)+\frac{(CD)^2}{4}} \cdot [erf(\frac{2B + C^2D^2}{2C}) - erf(\frac{2(B-t) + C^2D}{2C})] + Et + F \qquad (12)$$

where the linear background radiation, $f_{bg}(t)$ is represented as:

$$f_{bg}(t) = Et + F \qquad (13)$$

6.6.2 Duration of the Flare

During detection, we marked the apparent start and end points of a flare, these are the points between which, we are sure that a single flare is observed. However, to get the actual starting, ending point of flare, we will have to extend the solar-flare rise and decay phases, until they reach the actual background photon count of that day (minimum photon count). The Elementary Flare profile is a very complicated function. For actually modelling the flare, we model the rise and decay phase separately. We modelled the rise phase, as a line. This is justified from the EFP because, the rise is sudden, there's a significant curvature only near the peak. Hence, to a good approximation, rise phase is linear.

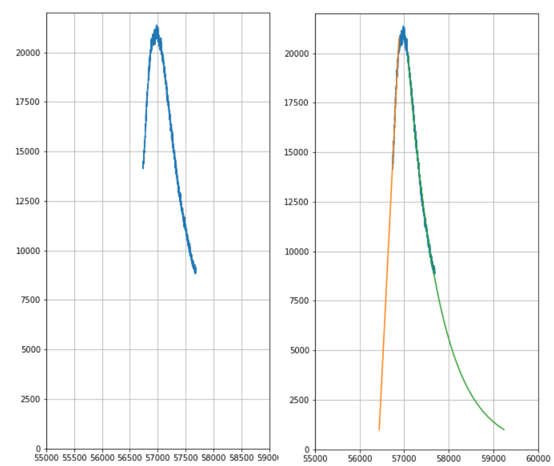


Figure 29: Modelling the rise and decay phases

The decay phase was modelled using $A \cdot e^{-b \cdot \sqrt{t}}$. Actually we wanted to use $A \cdot e^{-b \cdot x^c}$ but curve_fit works best for linear fitting, hence we tested all values of c, the least covariance determinant came for c = 0.5.

As is evident from the figure, the fitting is a very good approximation. Using the curves obtained from curve fitting, we can get the time corresponding to the background photon count. This gives us the actual start and end of the Flare.

Why are starting and end points of a flare event important?

It is important to understand why knowing the starting and ending time is important. The XSM (X-Ray Solar Monitor) and CLASS(Chandrayaan 2 Large Area Soft X-ray Spectrometer) instruments aboard the Chandrayaan-2 Orbiter work together, the XSM points towards the Sun to detects flares, and CLASS points toward the Lunar Surface. During a flare event, the elements present on the lunar surface become active, their abundance can be known through observing the spectra from the surface.

Hence, knowing exact start and end of a flare event from XSM data is important as it may then be used for analysing relevant data from CLASS also.

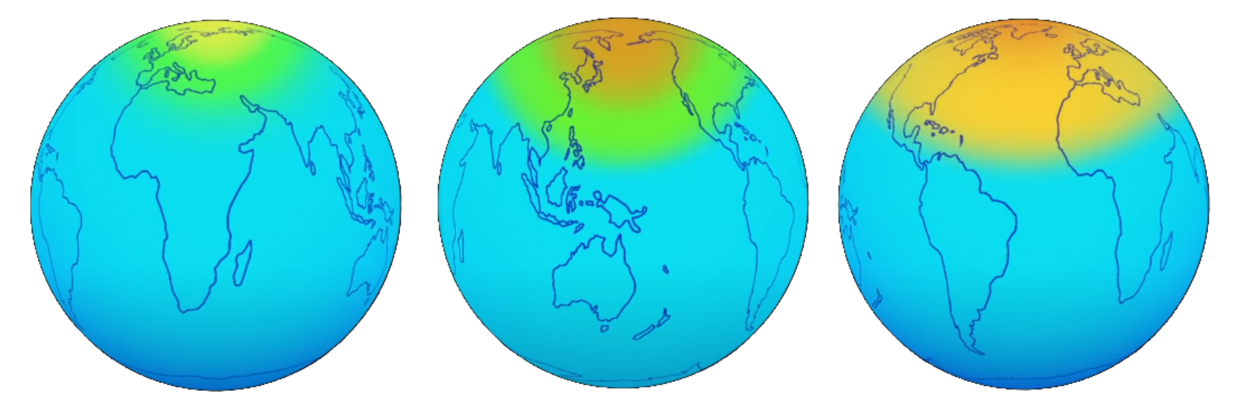
6.7 Classification

We can classify Solar Flares on the basis of it's peak photon count rate (and hence peak flux) using the following table.

CLASS	Count Rate (s^{-1})	Flux (Wm^{-2})
A	N < 250	$X < 10^{-7}$
B	250 < N < 2500	$10^{-7} < X < 10^{-6}$
C	2500 < N < 25,000	$10^{-6} < X < 10^{-5}$
M	25,000 < N < 250,000	$10^{-5} < X < 10^{-4}$
X	250,000 < N	$10^{-4} < X$

A class C flare is too weak to affect Earth. An M class flare can cause brief radio blackouts at the poles and minor radiation storms that might endanger astronauts. The X class flares are capable of producing long lasting radiation storms which can harm our satellites. Flight passengers too may get exposed to small amounts of radiation.

The following figure shows relative strengths of class C, M and X flares.



Class C Class M Class X

Figure 30: Relative strengths of different classes.

Consider a Solar Flare with Peak count rate 1000 Photons/second. From the table, we see that The flare belongs to class **B**.

We can further divide the solar flares into subclass. Countrate 250 corresponds to B0 and Countrate 2499 almost to B9.9 and intermediate values are equally partitioned on the logarithmic scale. The general expression for subclass of a Flare with peak count rate x becomes:

$$Subclass = 10 \cdot \log_{10}(\frac{x}{250}) \tag{14}$$

where x corresponds to the peak intensity of the solar flare in count rate. For the present case

$$Subclass = 10 \cdot \log_{10}(\frac{1000}{250}) = 10 \cdot \log_{10}(4) \approx 6.02 \tag{15}$$

So the flare belongs to class B6.0

7 Conclusion

The advent of modern techniques of computation and outstanding instruments of observation, has made astronomical data analysis much easier. In this project, we saw the general methods used in spectroscopy, exoplanet detection and solar flare analysis.

What's next?

The XUVI team plans to participate in the Lunarathon being conducted by ISSDC on the ocasion of international moon day. We'll be using Chandrayaan-2 data to carry forward what we learned in the project.

JUX

The tools we used and developed during the project, will be made available with the next version of JUX package. The pre-processing methods and filters used during Exoplanet detection and Solar Flare analysis were kept as general as possible, hence, these can be applied to a wide variety of data by anyone by installing the package from PyPI.

8 References

- **Automated Solar Flare Statistics in soft X-Rays**, *Aschwanden and Freeland, 2012*
- **SphinX catalogue of small flares and brightenings**, *Grycuik et al., 2014*
- **Flare Characteristics from X-Ray Light curves**, *Grycuik et al., 2017*
- **Light Curves - an Introduction**, *NASA*
- **Transit Light Curve**, *Andrew Vanderburg*
- **Astrometry**, *Wikipedia*