University of Toronto

# AST326 Lab Report 3

*Brianna Liz Binoy*

Lab Group Z

December 8, 2023

## 1. Abstract

Photometry is used in many fields of astronomy to analyze the intensity of light coming from an astronomical object. In this lab, we applied the standard data reduction process to delete any corrupted data and correct any overflow images. We then conducted differential and aperture photometry to obtain the flux of a supernova explosion and analyzed the results by plotting a light curve of apparent magnitude against time. By analyzing the Signal-To-Noise ratio of all the images of the supernova, the image of the first detection was obtained and was taken on September 22, 2015, at 3:30 AM. The peak apparent magnitude obtained from this light curve is $18.7 \pm 0.1$, which was 16.7 days after the first detection of the supernova. The light curve was further analyzed by doing a power fit curve, and the epoch of the first light was found to be on September 20, 2015, at 4 PM.

## 2. Introduction

In astronomy, photometry is a method used to calculate the flux or intensity of light coming from an astronomical object. Photometry is used to observe astronomical objects that vary in brightness over time, like variable stars and supernova explosions. One branch of photometry is differential photometry, which simultaneously measures the brightness of the target astronomical object against two or three reference stars of constant brightness. Another branch is aperture photometry, which is a technique for calculating the flux of a star through the use of a CCD (Charge-Coupled Device). This method involves summing the pixel counts associated with the star using an aperture and subtracting the sum of corresponding counts from surrounding pixels (noise) using an annulus. The equation for the fluxes is given by:

$$F_{star} = \sum_{aperture} (I(x,y) - I_{bg}) \tag{1}$$

where $I(x, y)$ is the pixels in the aperture and $I_{bg}$ is the local background intensity.

The local background intensity can be calculated using the following equation:

$$I_{bg} = \sum_{annulus} \frac{I(x,y)}{N_{annulus}} \tag{2}$$

where $N_{annulus}$ is the number of pixels in the annulus of the sub-image.

The underlying motivation of this paper is to analyze the flux of a Type 1A supernova explosion over time using aperture and differential photometry and to use that to determine its epoch of peak brightness and its epoch of first light.

Type 1A supernova explosion happens in a binary star system, where one of the stars is a white dwarf. A supernova light curve plots the apparent magnitude of the supernova against time. The B-band light curves of all type 1A supernova explosions follow the same shape, with an initial rapid increase in brightness in the first 15 days and then a fairly rapid decline. In the beginning of a Type 1A supernova, the supernova brightness increases proportional to the following:

$$I_{supernova} \propto C(t - t_1)^2 \tag{3}$$

where $t_1$ is the epock of first light and $C$ is a fitting coefficient.

Astronomers use the term ”apparent magnitude” to describe the brightness of an object in the sky relative to the brightness of another astronomical object. An object's apparent magnitude depends on its flux, its distance away from the observer, and any light blocked by interstellar dust between the object and the observer. The equation for the apparent magnitude is given by:

$$m_1 - m_2 = -2.5log(f_1/f_2) \tag{4}$$

where $m_1$ and $m_2$ are apparent magnitudes of 2 objects in the sky, and $f_1$ and $f_2$ are the fluxes of both astronomical objects respectively. A smaller value of apparent magnitude means a brighter astronomical object.

After taking multiple images over 40 days of the point in space where a Type-1A supernova exploded we reduced the images and data from the images to delete any bad data/images. We then conducted aperture photometry to calculate the flux of 3 reference stars and the supernova explosion. By using standard photometric calibration, the apparent magnitude was obtained. After that, a supernova light curve was plotted and we conducted polynomial fitting to the light curve and power-law fitting to estimate the epoch of its peak brightness and its epoch of first light. Using the Signal-To-Noise ratio, the epoch of first detection was found.

## 3. Data and Observation

The data collection was performed by M. Drout and D.S Moon. 433 CCD images were taken by the Korean Microlensing Telescope Network which consists of 3 telescopes; one in Chile, another in South Africa, and one in Australia. The images were centered at the coordinate (RA, DEC) = (00:57:03.19, -37:02:23.6). They were taken in the span of 40 days, starting from September 20, 2015, to October 30, 2015. The CCD imaging observations were made with a filter located in front of the CCD. The filters transmit photons of certain wavelengths, determined by the filter transmission curve. The images were obtained with a B-band filter of image size 710 x 850 pixels. Some of the images were corrupted due some issues with the instrument.

## 4. Data Reduction

There were several methods used to filter out the bad images and minimize the noise from image data. The first method used was simply hand-picking and deleting any visually bad images from the image files. The code for this is in 7.1.

In the images obtained, the background pixels around the stars do not show a constant value. There is a significant pixel-to-pixel variation in the background pixels even where there are no apparent objects and this is due to the noise originating from sky background, dark and bias noise from CCD and readout noise from the CCD readout electronics. There is also noise from other objects like stars and galaxies. Hence to measure the flux of the reference stars, and the supernova, aperture photometry, described in Section 2 was conducted. For each image file, after zooming into the 3 reference stars, an aperture and annulus were created around the 3 reference stars for all the images. The data from both the aperture and annulus were obtained, and the following equation was used to get the star intensity:

Star Intensity = Entire Intensity inside Object Aperture - Background Intensity inside Object Aperture

Assuming that the stellar images are 2-dimensional Gaussians, the aperture radius was taken to be 3 times the standard deviation (Figure 1). The code for this is under 7.2.

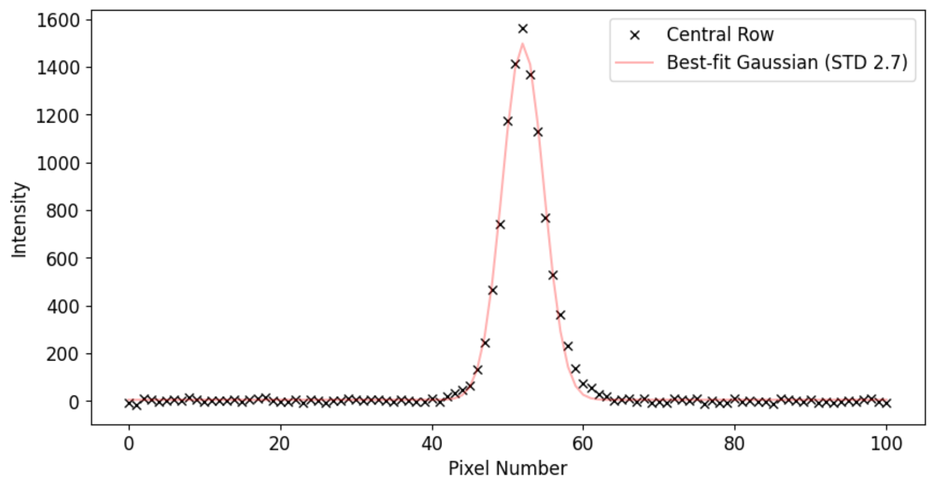



Fig. 1.—: The Gaussian fit for one of the reference stars in one of the image files. The standard deviation, 2.7, was used to find the aperture radius. The image file was randomly selected, however, it was made sure that the image was clear and the star was not too bright, or too dim in the image.

To reduce the noise even further, the SNR ratio of the supernova was taken, and any image that had an SNR of less than 3 was discarded. See Figure 5.

Due to the very high intensities in some of the images of the stars and supernova explosion, there were overflow errors. This was corrected using the code in 7.5. A comparison of the before and after of one of the images is shown below. All the images with very high overflow errors were corrected. Those that had minimal overflow error were discarded (Figure 3).

## 5. Data Analysis

After taking the sub-images of the reference stars and conducting aperture photometry on the reference stars, their fluxes were found. The fluxes were found using the Equation 1. The code for aperture photometry is under 7.8.

Its uncertainties were found using this equation:

$$\Delta F = \sqrt{\frac{F}{g} + N_{aperture}\left(1 + \frac{\pi}{2}\frac{N_{aperture}}{N_{annulus}}\right)\sigma_{bg}} \tag{5}$$

The flux of the supernova explosion was found the same way. The plot of the flux of the supernova explosion against the other reference stars is shown in Figure 4.

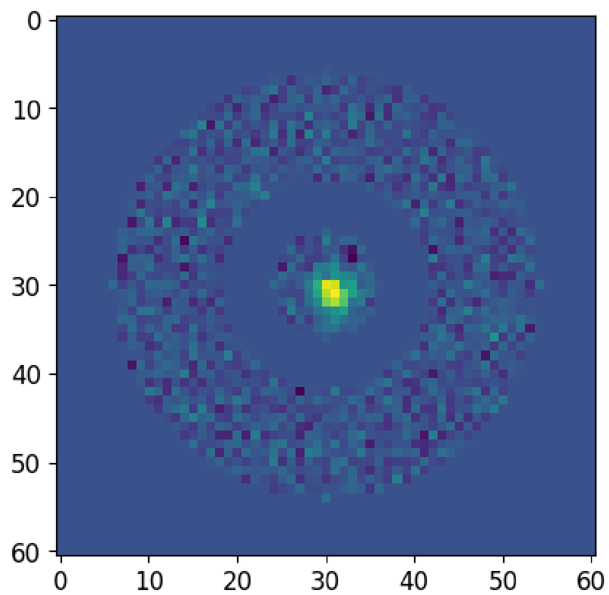



Fig. 2.—: Sub image of one instance of the supernova explosion after conducting aperture photometry. You can see the annulus ring with the background noise. The supernova explosion is the bright yellow dot in the middle

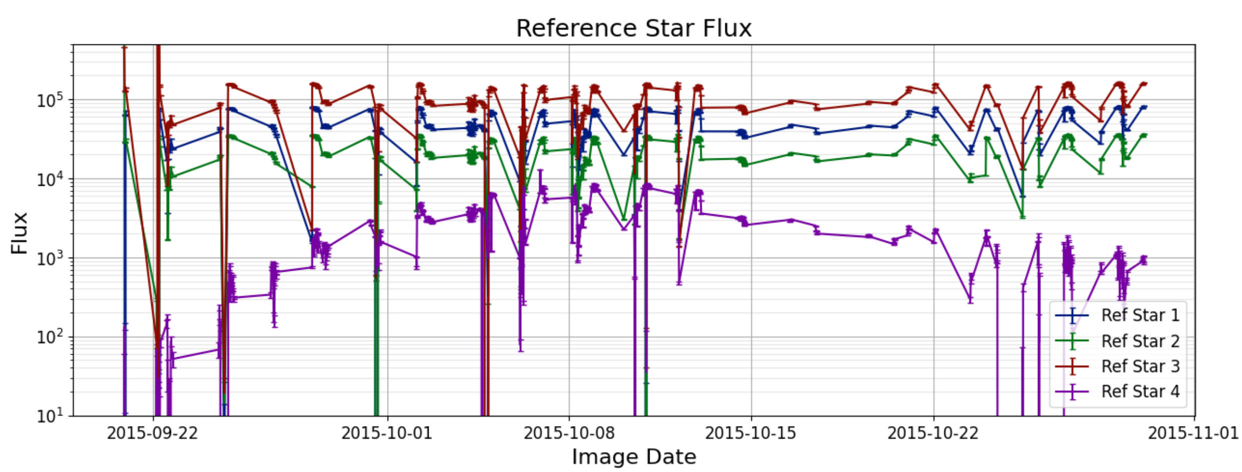
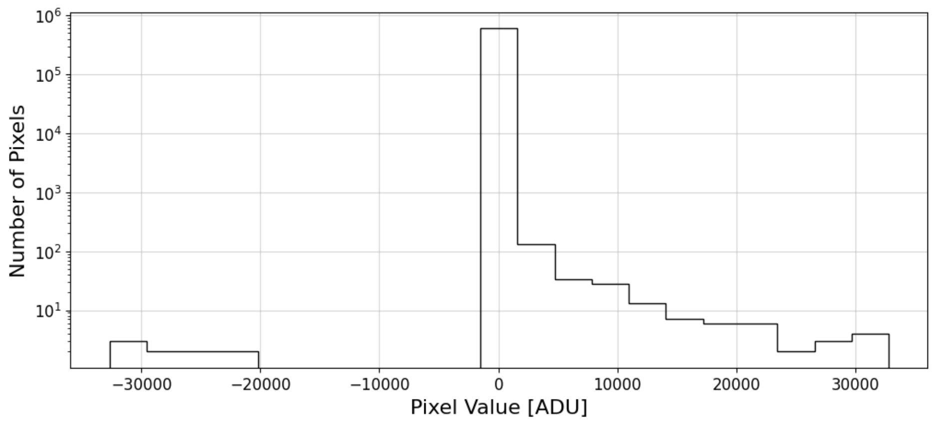
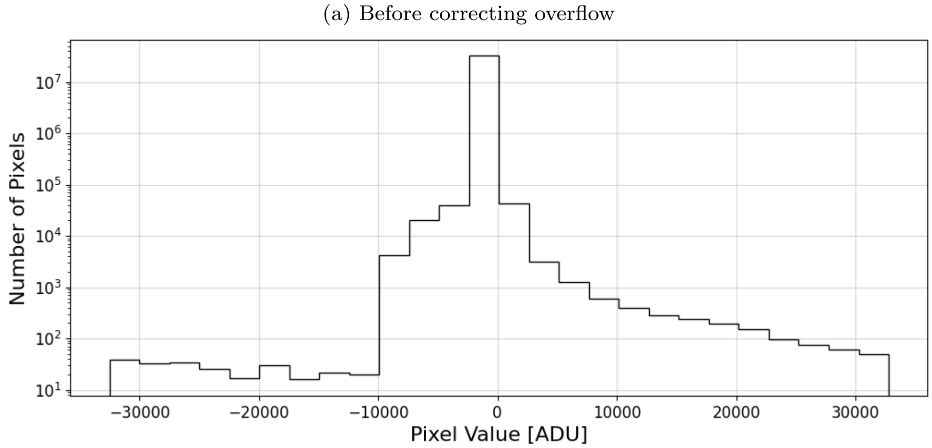



Fig. 4.—: Plot of the fluxes of the 3 reference stars and the supernova against the images, labeled in terms of the date they were taken. For each of the reference stars, the fluxes remain in a certain range, except for dips in between. For the supernova explosion, the flux increased and then decreased, as expected. Since all the stars and the supernova have dips in roughly the same image value, we can conclude that these dips are from images that are bad or have some temporary noise in the background. It is impossible that all the stars light years away from each other simultaneously reduced in flux.

The image of the first detection of the supernova is shown below:




(a) Before correcting overflow




(b) After correcting overflow

Fig. 3.—: Before and after applying correction to one of the overflow images

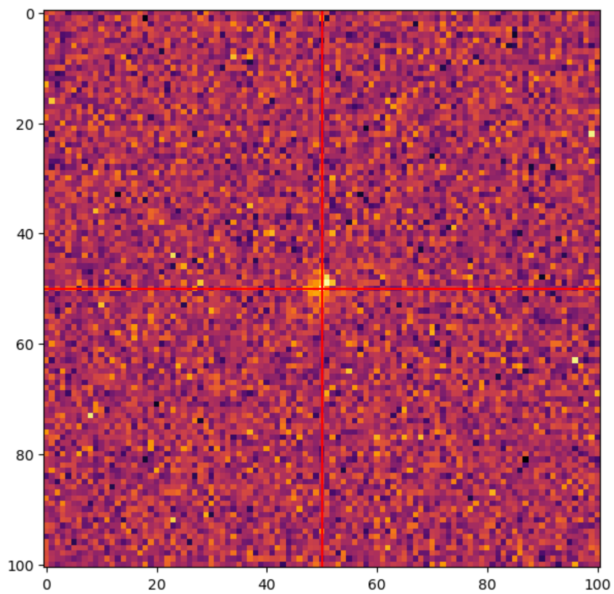



Fig. 5.—: Image of first detection with SNR greater than 3. You can see more brightness in the middle of the image, which is the supernova

.

The image of the last non-detection of the supernova is shown below:

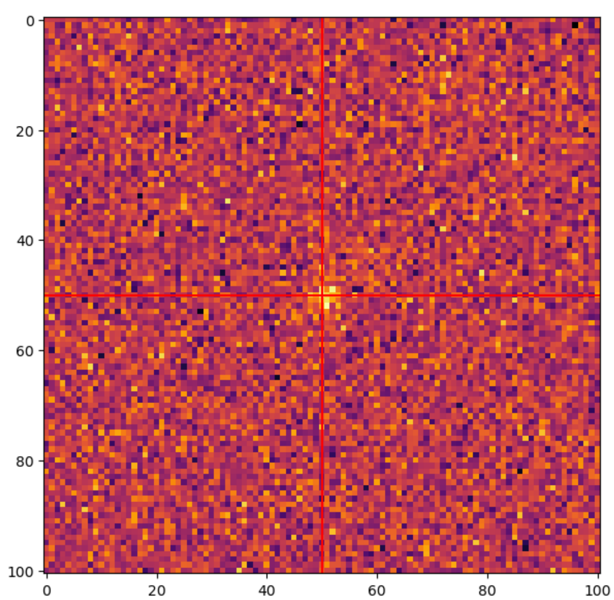



Fig. 6.—: Image of the last non-detection with SNR less than 2.

The code to finding this is under 7.3. This image was taken on September 22 2015 at 3:30 AM. Using the magnitudes of the reference stars, the magnitude of the supernova explosion was found over time. Further reduction was done to remove any images with SNR less than 3 and the light curve was plotted as shown in Figure 7. By converting all the dates of the images into seconds, and then subtracting them from the time of first detection, we plotted the light curve against time since the first detection of supernovae. There are points on the plot whose error bars do not touch the polynomial curve fit, and these points are likely to be from the bad images mentioned in Figure 4. With the removal of this data, a more accurate curve could have been obtained.

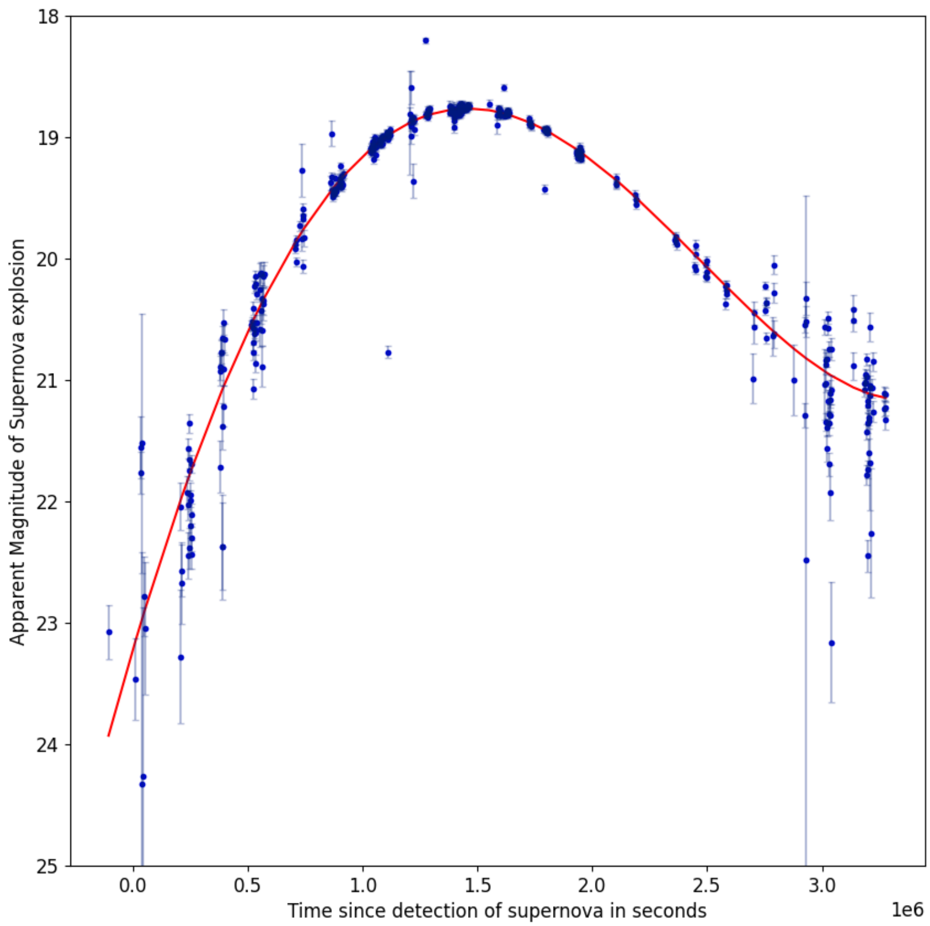



Fig. 7.—: Supernova light curve: From this light curve, it is clear to see that this is a Type-1a supernova explosion, with a rapid increase in brightness over time before peak brightness and a decrease in brightness over time after peak brightness. There are points on the graph whose error bars do not touch the polynomial curve fit.

Using the polynomial curve fit, the peak brightness was found to be $18.7 \pm 0.1$. The uncertainty was calculated using the code in 7.6 . The time this happened is 17 days after the first detection.

To find the epoch of first light, the supernova magnitudes were converted back to fluxes, using Equation 4 and plotted against time. The uncertainties were calculated in the code under 7.7. After limiting the power curve to only include values that follow Equation 3 which are below the 40 percent mark, the following plot was obtained.

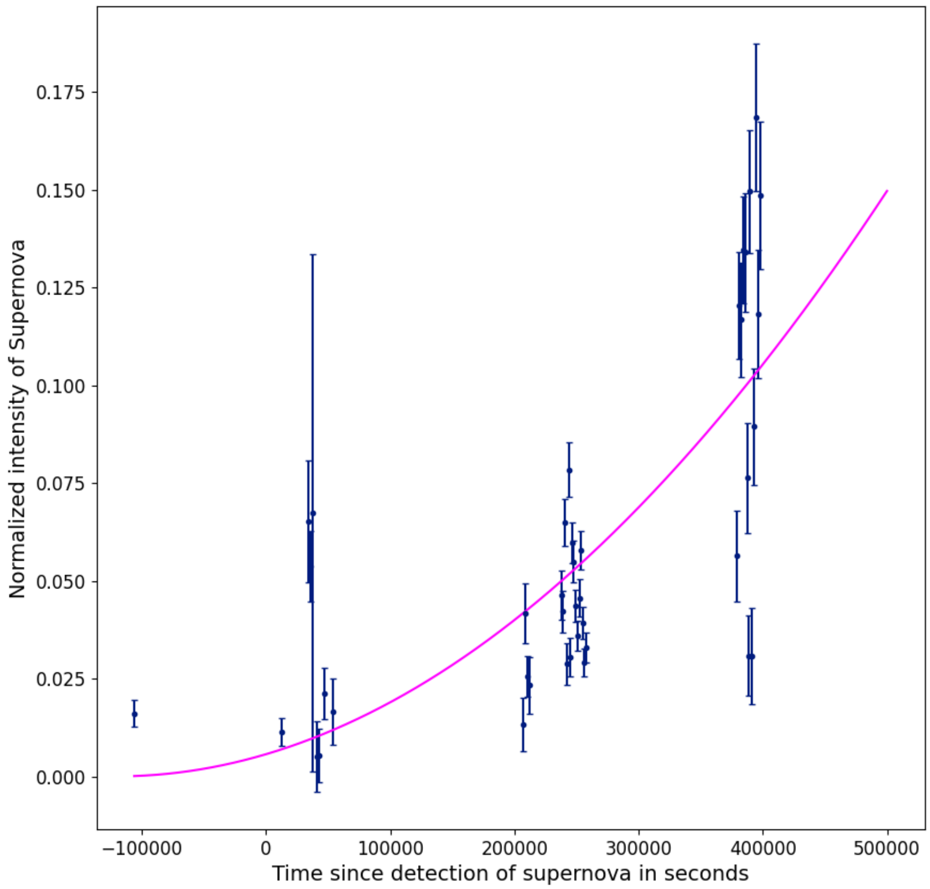



Fig. 8.—: Enter Caption

From this, we calculated the epoch of the first light to be slightly more than a day before the first detection, which is September 20, 2015, at 4 PM.

## 6. Discussion

The supernova light curve produced has a lot of points whose error bars are not touching the polynomial fit. This could be due to not filtering out all the bad images. As mentioned in Figure 4, there were bad images that caused the fluxes to be very low. Filtering those images out would have given a better light curve, with fewer points that are far off from the polynomial fit. Furthermore, the lower the brightness of the supernova, the lower the Signal-To-Noise ratio, hence we must expect larger error bars as the apparent magnitude increases. In the plot we obtained, the error bars did increase in size, but there were still many with small error bars. Centering the images of the supernova and reference stars would have been a more accurate way to get their fluxes using aperture photometry. However, almost all sub-images were nearly centered, hence centering the images would not have made much difference in the results.

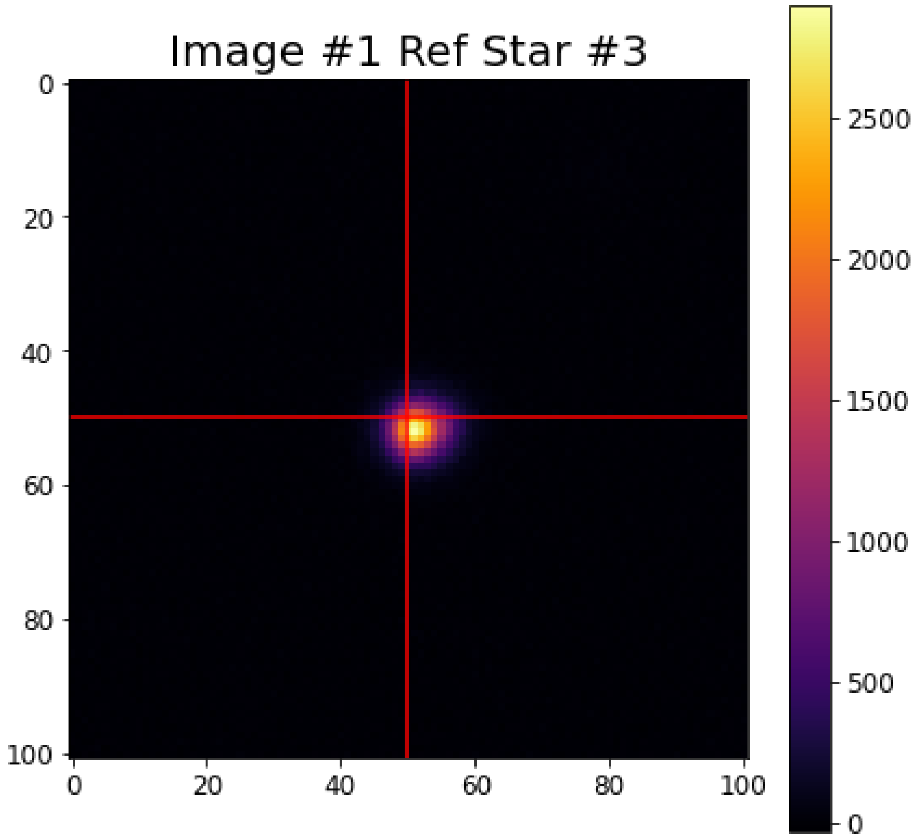



Fig. 9.—: An image of a reference star without centering it. The star is already very close to the center, hence the difference in results would be minimal when conducting aperture photomerty

Instead of discarding images of minimal overflow, they should have been kept, and a new code should have been written to correct them. Adding to that, it would have been better to find the first detection of the supernova after obtaining the power curve, since by then, the bad images that may have affected the Signal-To-Noise ratio would be removed. Since this was done before much filtering was done, the image of the first detection obtained may be wrong. Nevertheless, the light curve obtained does represent a Type-1A supernova explosion; the explosion takes roughly 17 days to reach peak brightness after first detection, which is the time a typical Type-1A supernova takes to reach its peak. After reaching peak brightness, the supernova brightness declines, as expected of a Type-1A supernova. The time of first detection obtained by DS Moon and M Drout, is September 24, at 21:33:00.000, which is 3 days and 8 hours more than the time we obtained. The reason we got a different time is due to lesser data reduction as mentioned.

## 7. Conclusion

The flux of the supernova against time was obtained by reducing the data and using aperture and differential photometry. By obtaining the flux, the magnitude was obtained and a light curve was plotted. From the light curve plot, the peak magnitude of the explosion was found to be $18.7 \pm 0.1$. A power-law fitting was conducted to obtain the epoch of the first light which was September 20, 2015, at 4 PM. This result is slightly off with the result obtained by DS Moon and M Drout.

## Appendix

### 7.1. Handpicking Bad Images

```

#reading the filename and printing the image
def plot_image(filename):
    data1 = read_file(filename)
    plt.figure(figsize=(10,10))
    plt.imshow(data1,
               vmin = np.percentile(data1,5),
               vmax = np.percentile(data1,95),
               origin = 'lower',
               cmap = 'inferno')
    plt.show()

#names of bad images that were handpicked
examples = ['AST325-326-SN-20150924.7465.fits',
            'AST325-326-SN-20150930.5694.fits',
'AST325-326-SN-20150930.6694.fits','AST325-326-SN-20151004.6194.fits',
'AST325-326-SN-20151006.9146.fits', 'AST325-326-SN-20151008.9813.fits',
'AST325-326-SN-20151004.5653.fits', 'AST325-326-SN-20151004.8792.fits',
'AST325-326-SN-20151008.9597.fits', 'AST325-326-SN-20151026.0403.fits',
'AST325-326-SN-20151026.0549.fits', 'AST325-326-SN-20151012.8410.fits',
'AST325-326-SN-20151012.8410.fits', 'AST325-326-SN-20151004.6354.fits',
'AST325-326-SN-20151010.8764.fits','AST325-326-SN-20151011.0174.fits',
'AST325-326-SN-20151011.0479.fits', 'AST325-326-SN-20151017.4861.fits',
'AST325-326-SN-20151029.4160.fits']



#printing the bad images
for f in examples:
    print(f)
    plot_image(f)

#finding the indices of the image files that in an array
indices = np.where(np.isin(img_files, examples))[0]

#removing the values of those indices from the image files, and the file that contains

#the dates the images were taken.
```

```python
file_times= [item for i, item in enumerate(file_times) if i not in indices]
img_files = np.array([x for x in img_files if x not in examples])

print(len(img_files))
print(len(file_times))

```

### 7.2. Handpicking Bad Images

```python
#reading the data file
data  = read_file(img_files[2])
header = fits.open('/content/drive/MyDrive/ASTFitsFiles/'+str(img_files[2]))[0].header
w = WCS(header)
stars = np.array([SkyCoord('00h56m49.70s', '-37d01m38.31s'),  SkyCoord('00h56m46.43s',
 ↪  '-37d02m29.50s'), SkyCoord('00h56m58.27s', '-36d58m16.60s')])
ref_stars = [stars[0], stars[1], stars[2]]
boxsize = 50

#looping over the 3 reference stars
for j, star in enumerate(ref_stars):

        # locate the star
        ref_pos = skycoord_to_pixel(star, w)
        ref_x, ref_y = ref_pos

        # looking at a box around the star to set up our flux code above
        ref_x_int, ref_y_int = ref_x.astype(int), ref_y.astype(int)

        #generating region around star
        sub_im = data[ref_y_int-boxsize: ref_y_int+boxsize+1,
                      ref_x_int-boxsize: ref_x_int+boxsize+1]
        x, y = np.arange(sub_im.shape[0]), sub_im[sub_im.shape[0]//2,:]
        #plotting the gaussian curve to get the standard deviation
        popt, pcov = curve_fit(gaus,
                               x,y, p0=[45, 50,25, 0])
        print(popt)
        plt.figure(figsize=(10,5))
        plt.plot(x,y, 'x', color = 'k', label = 'Central Row')
        plt.plot(x,gaus(x,*popt),color = 'r', alpha = 0.3, label = f'Best-fit Gaussian
         ↪  (STD {round(popt[2],1)})')
        plt.xlabel("Pixel Number")
        plt.ylabel("Intensity")
        plt.legend()
        plt.show()

```

### 7.3. Code for Finding SNR

```python
import numpy as np

def calculate_snr(sub_image):
    # Replace this with your actual SNR calculation
    signal = np.mean(sub_image)  # Example: signal is the mean intensity of the sub-image
    noise = np.std(sub_image)    # Example: noise is the standard deviation of the
    ↪  sub-image

    # Avoid division by zero
    if noise == 0:
        return np.inf

    return signal / noise

# List of sub-images (replace with your actual sub-images)
sub_images = [...]

# Find the first sub-image with SNR greater than 3
for idx, sub_image in enumerate(sub_images):
    snr = calculate_snr(sub_image)
    if snr > 3:
        print(f"The first sub-image with SNR > 3 is at index {idx}.")
        break
else:
    print("No sub-image with SNR > 3 found.")
```

### 7.4. Correcting Overflow

```python
def read_file(filename):
    hdu = fits.open('/content/drive/MyDrive/ASTFitsFiles/'+str(filename))[0]
    data = hdu.data.astype(int)
    #after reading each image, the code below corrects for overlow.
    data[data < -1e3] = ((2**15-1)-np.abs(data[data < -1e3])) + (2**15-1)
    return data
```

### 7.5. Correcting Overflow

```python
def read_file(filename):
    hdu = fits.open('/content/drive/MyDrive/ASTFitsFiles/'+str(filename))[0]
    data = hdu.data.astype(int)
```

```
    #after reading each image, the code below corrects for overlow.
    data[data < -1e3] = ((2**15-1)-np.abs(data[data < -1e3])) + (2**15-1)
    return data

```

### 7.6. Uncertainy in Magnitude Calculation

```
def get_mag_err(SN_flux, SN_flux_err, Ref_flux1, Ref_flux2, Ref_flux3, Ref_flux_err1,
↪ Ref_flux_err2, Ref_flux_err3, Ref_mag_err1, Ref_mag_err2, Ref_mag_err3):
  un_log_SN = -2.5*(SN_flux_err/SN_flux)
  un_log_ref1 = 2.5*(Ref_flux_err1/Ref_flux1)
  mag_error_1 = np.sqrt((Ref_mag_err1)**2 + (un_log_SN)**2 + (un_log_ref1)**2)
  un_log_ref2 = 2.5*(Ref_flux_err2/Ref_flux1)
  mag_error_2 = np.sqrt((Ref_mag_err2)**2 + (un_log_SN)**2 + (un_log_ref2)**2)
  un_log_ref3 = 2.5*(Ref_flux_err3/Ref_flux1)
  mag_error_3 = np.sqrt((Ref_mag_err3)**2 + (un_log_SN)**2 + (un_log_ref3)**2)
  uncertainty_SN = ( mag_error_1**2+ mag_error_2**2 + mag_error_3**2 )/3
  return uncertainty_SN

#every reference star's flux is defined as Ref_flux#
#every reference star's flux error is defined as Ref_flux_err#
#every reference star's magnitude error is defined as Ref_mag_err#
```

This calculation was done using Equation 4 and conducting error propagation.

### 7.7. Uncertainy in Supernova flux

```
def power_curve(t, c, t_1):
  return c*((t-t_1))**2

def flux_err_for_power(mag, mag_err):
  return 0.4*(10**(-0.4*mag))*math.log(10)*mag_err

```

This calculation was done after using the equation in the first function and using error propagation.

### 7.8. Aperture Photometry Code

```
def centroid(image):

    xvals = np.arange(image.shape[1]) - image.shape[1]//2
    centroid_x = np.sum(xvals*image**2)/np.sum(image**2)
    centroid_y = np.sum(xvals*image.T**2)/np.sum(image**2)
```

```
    return np.array([centroid_y, centroid_x])

def get_flux(sub_img, ap, an):
    N = an[an>0].size
    bg = np.sum(sub_img*an)/N
    flux = np.sum((sub_img- bg)*ap)
    return flux

def get_flux_errs(sub_img, ap, an, gain):
    N_ap = ap[ap>0].size
    N_an = an[an>0].size
    bg = np.sum(sub_img*an)/N_an
    flux = np.sum((sub_img- bg)*ap)
    a = flux/gain
    b = (1 + math.pi*0.5*(N_ap/N_an))
    un_bg = np.sqrt(np.sum((an - bg)**2) /(N_an-1))
    flux_err = np.sqrt(a + N_ap*b*un_bg)
    return flux_err


try:
    fluxes, flux_errs = np.load("tmp_fluxes.npy"), np.load("tmp_fluxes_err.npy")
except:
    #Repeating the code from above
    boxsize = 50

    sub_im = data[100-boxsize: 100+boxsize+1, 100-boxsize: 100+boxsize+1]

    #Create an aperture and annulus model
    aperture = get_aperture(sub_im.shape, 12)
    annulus = get_annulus(sub_im.shape, 20, 40)

    fluxes = np.empty((len(ref_stars),len(img_files)))
    flux_errs = np.empty_like(fluxes)


    for i, f in enumerate(img_files):

        # read the image
        data = read_file(str(f))
        header =
        ↪  fits.open('/content/drive/MyDrive/ASTFitsFiles/'+str(img_files[i]))[0].header
        w = WCS(header)

        for j, star in enumerate(ref_stars):
```

```


            ref_pos = skycoord_to_pixel(star, w)
            ref_x, ref_y = ref_pos
            ref_x_int, ref_y_int = ref_x.astype(int), ref_y.astype(int)
            sub_im = data[ref_y_int-boxsize: ref_y_int+boxsize+1,
                        ref_x_int-boxsize: ref_x_int+boxsize+1]
            aperture = get_aperture(sub_im.shape, 6)
            annulus = get_annulus(sub_im.shape, 12, 24)

            #sub_im = ta.center_image(data, sub_im, ref_pos)

            #Get the flux for each of the reference stars
            fluxes[j,i] = get_flux(sub_im, aperture, annulus)
            flux_errs[j,i] = get_flux_errs(sub_im, aperture, annulus, 4)


    np.save("tmp_fluxes", fluxes)
    np.save("tmp_fluxes_err", flux_errs)
```