A Program-Based Method of Plotting the Orbital Data of a Double Star

Nick Smith[1], Chris Foster[1], Blake Myers[1], Barbel Sepulveda[1], and Russell Genet[2, 3, 4]

1. Lincoln High School, Stockton, California
2. Cuesta College, San Luis Obispo, California
3. California Polytechnic State University, San Luis Obispo
4. Concordia University, Irvine, California

Abstract: A program for plotting the orbital data of double stars was developed and tested on the double star WDS01078+0425, previously studied by Adam et al. (2014). This program automates the process in determining the vertical and horizontal components of the separation using the position angle of a given double star. These components are then used to plot a point on the last orbital solution using a program such as MSPaint or ImageJ.

Introduction

Lincoln High School students met to conduct original double star research. They developed a method that uses trigonometry to convert a measurement in arc seconds to pixel distances so that a point can be plotted on an orbital diagram. A program written in Java was developed to accomplish this task. The double star WDS 01078+0425 (Adam et al. 2014) was used to validate the program.

Method

The double star WDS 010878 + 0425 had a separation value of 0.255 arc seconds and a position angle of 227.9 °. The scale on the previously determined orbital solution was used to determine the ratio of pixels to arc seconds. In ImageJ, an equal length interval was measured and found to be 140 pixels, as shown in Figure 1.

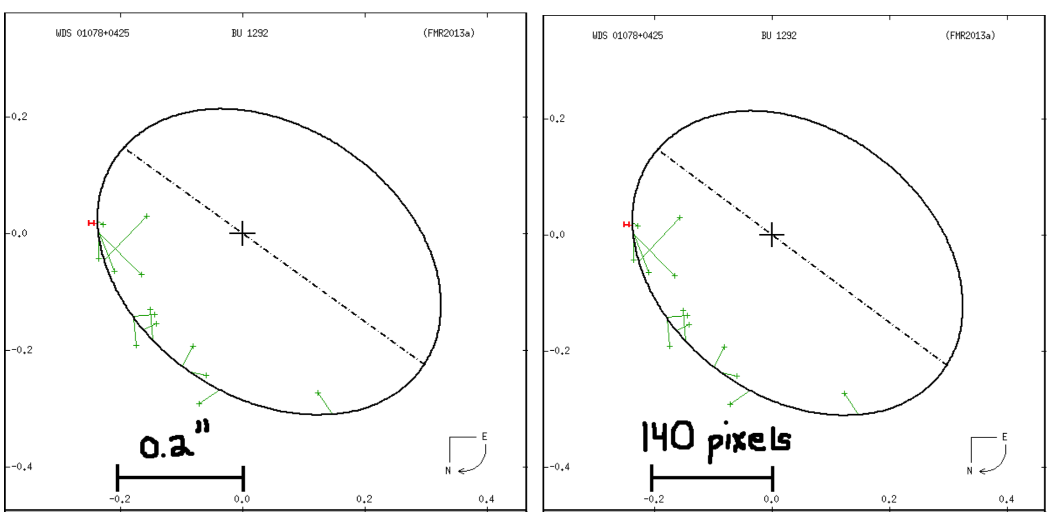

Figure 1: The relationship between arc seconds and pixels.

The initial x coordinate value corresponds to the point located underneath the 0.2 mark in Figure 2. The final x coordinate value corresponds to the point located underneath the 0.0 mark. To obtain the net distance in one interval, the final x-coordinate was subtracted from the initial x-coordinate (See Eq. 1).

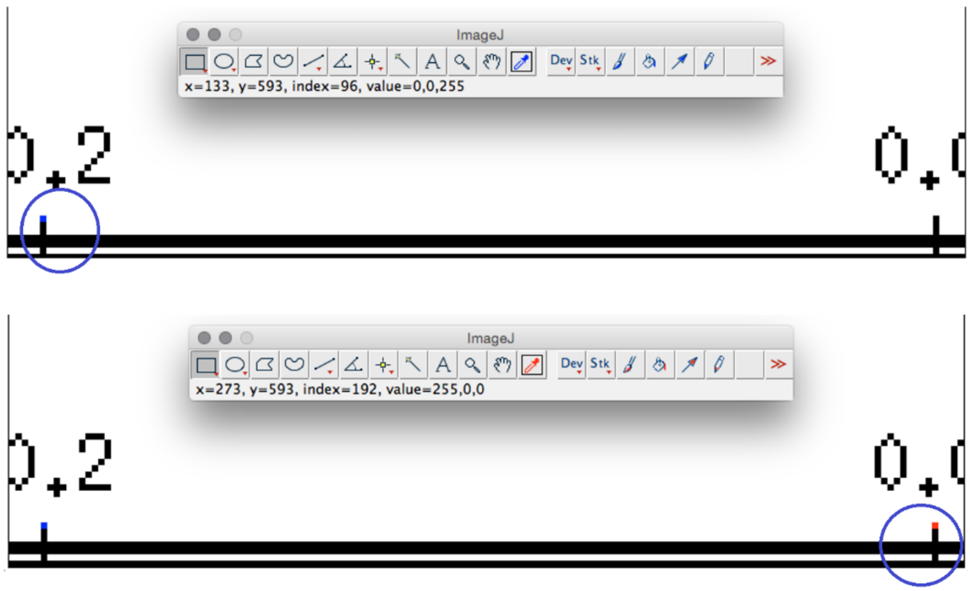

Figure 2: The upper panel shows the initial x-coordinate and the lower panel shows the final x-coordinate of the measurement.

$P_f - P_i = \Delta P$ *Eq. 1 (Equation used to calculate the number of pixels)*

$273 - 133 = \Delta P$

$\mathbf{140 = \Delta P}$

A visual representation of this observation is shown in Figure 3. Using the data from the selected double star, the point was determined to be in the 3rd quadrant from Celestial North. The following logic was used to determine the placement and net position angle of the double star observation:

If θ <= 90° (1st quadrant):
θ′ = θ // No change in the position angle

If θ > 90° and θ <= 180° (2nd quadrant):
θ′ = θ - 90° // New angle is the previous one minus 90°

If θ > 180° and θ <= 270° (3rd quadrant):
θ′ = θ - 180° // New angle is the previous angle minus 180°

If θ > 270° and θ <= 360° (4th quadrant):
θ′ = θ - 270° // New angle is the previous angle minus 270°

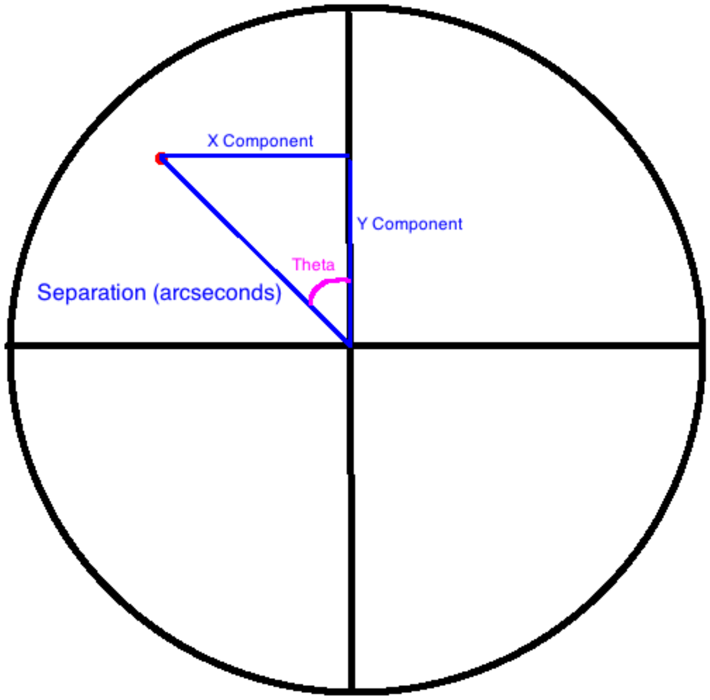

Figure 3: The location of observation in a circle.

Using the formula for the sine of an angle, the x-component of the separation in arc seconds was determined using Eq. 2.

$$\sin\theta = \frac{opposite}{adjacent} = \frac{xcomponent}{separation\rho} \qquad \textit{Eq. 2}$$

The new angle, θ′ in relation to the point, was calculated using Eq. 3.

$$\theta' = \theta - 180° \qquad \textit{Eq. 3}$$

With this equation, the obtained data values were entered to develop an answer for the x-component of this double star using Eq. 4. In order to obtain the correct values, all angles were measured in degrees. Similarly, using the cosine function, the y-coordinate was found.

$$0.255'' * sin\ (\theta°) = xcomponent \qquad \textit{Eq. 4}$$

The previously determined distance ratio was used to set up a proportion to convert the x- and y-components from arc seconds to pixels using Eq. 5. Substituting the value obtained for the x-component of the separation yielded the x-component in pixels. The same concept was applied to find the y-component in pixels.

$$\frac{0.2\ arc\ seconds}{140\ pixels} = \frac{xcomponent}{Pixel\ xcomponent} \qquad \text{Eq. 5}$$

A java based program was then developed to automate these steps. Shown below in Figure 4 is the user interface for the program. The values for the number of arc seconds, pixels, position angle, and separation were entered into the program and the program calculated the x- and y-coordinates.

```
===== O R B I T A L   C O N V E R S I O N   P R O G R A M =====
Created by Nick Smith
**Launched May 07, 2015 6:58 PM, Pacific Daylight Time**
Version 1.4 Release
Enter the number of arcseconds in 1 interval of the image: 0.2
Enter the number of pixels in that interval (you need to count these in a program such as paint): 140
Enter the separation value for this double star: 0.255
Enter the position angle of the star from celestial north [in degrees]: 227.9
===== R E S U L T S =====
Number of pixels left from the center point: 132.44
Number of pixels up from the center point: 119.67
```

Figure 4: The user interface after running the orbital plotting program.

Since the position angle was in the 3rd quadrant from celestial north, the point was found by moving 132 pixels to the left and 120 pixels up from the position of the primary.

To plot the point in ImageJ, the cursor was placed over the center coordinate point and the x- and y-coordinates were displayed in the ImageJ toolbar, as shown in Figure 5. The x-component (in pixels) was subtracted from the center x-coordinate to obtain the final x-coordinate. In a similar manner, the final y-coordinate was found. Since the center is made up of 4 pixels, the bottom leftmost pixel, was arbitrarily chosen to make the measurements.

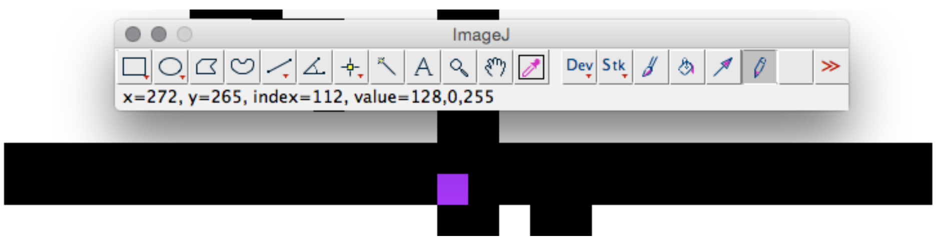

Figure 5: The toolbar for ImageJ and a magnified view of the position of the primary star.

Maintaining a constant y-coordinate of 265, a point was placed in the proper x-coordinate for reference (Figure 6).

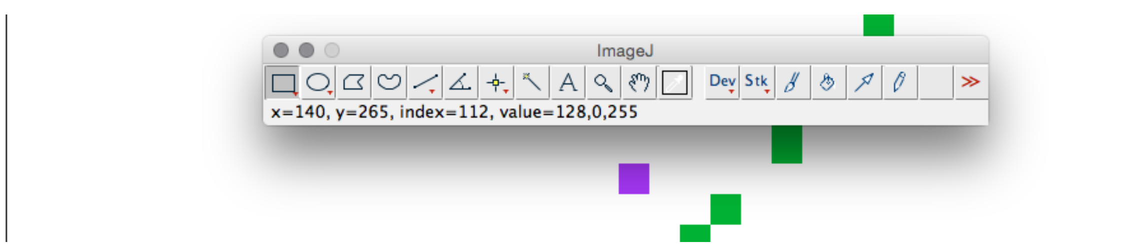

Figure 6: The intermediate points plotted in ImageJ.

Next, maintaining a constant x-coordinate, the cursor was moved up to the coordinate y = 145 and a final point was placed there (Figure 7). Once this point was plotted, the previous preliminary points were erased.

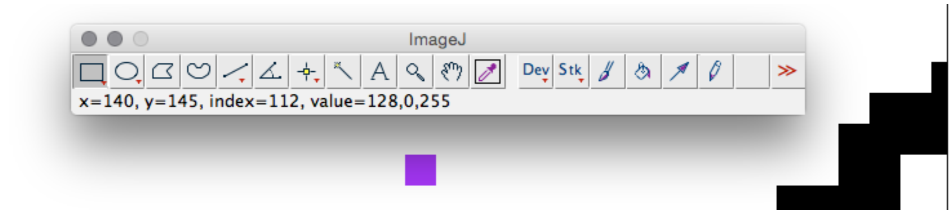

Figure 7: The final plotted position of the target WDS 01078+0425.

Results

Table 1 summarizes the x- and y-components in arc seconds and in pixels used to find the final x- and y-coordinates. Figure 8 shows the final plot.

Position Angle (°)	Separation (″)	x-component in arc seconds	y-component in arc seconds	x-component in pixels	y-component in pixels	Final x-coordinate	Final y-coordinate
227.9	0.255	0.189	0.171	132.4	119.7	140	145

Table 1: Orbital data for WDS 01078+0425.

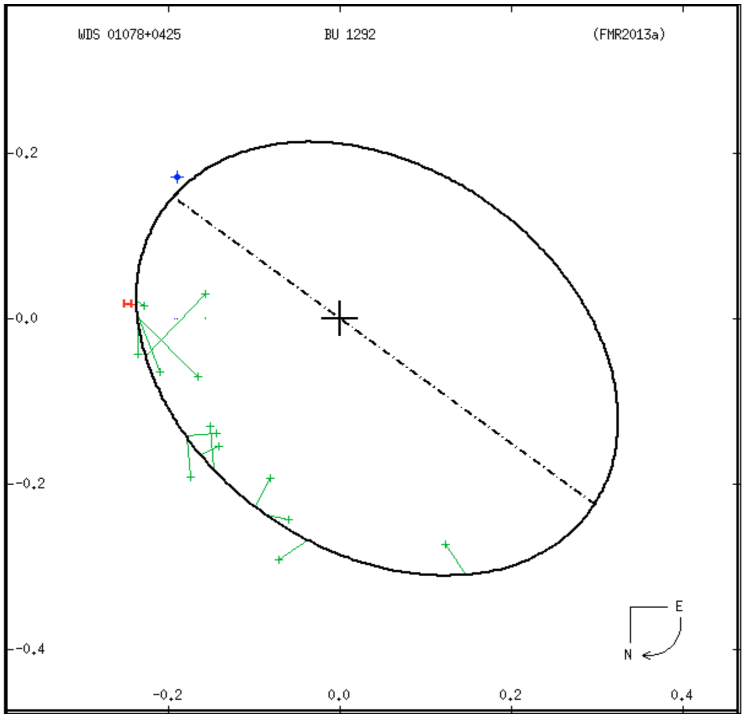

Figure 8: The new position plotted on the previous orbital solution at top-left of the ellipse.

Discussion

The Lincoln High School research team developed and tested a Java-based program that converts separation and position angle measurements into x- and y-components. These components are then converted into x- and y-coordinates which allow the points to be plotted in a program such as ImageJ or MSPaint. A mathematical relationship between the components was found and implemented in a procedure for plotting a point on a previously calculated orbital solution. This program utilized four logic statements along with parameters given by the user to convert separation and position angle into x and y separation values for a specific double star.

Once developed, the program was tested on the double star WDS 01078 + 0425. Figure 8 graphically confirms that the method for plotting data points on an orbital solution is accurate. The observed data point was plotted and found to be within an extremely small margin of error from the predicted ellipse.

One minor source of error could be the small ratio between the numbers of pixels in one interval of the orbital solution. Likewise, when calculating the number of pixels to move in either direction, the components had to be rounded because fractional pixels cannot be graphed in such a program. The program can be downloaded from the website:

http://nicksmith0625.wix.com/orbitalprogram

Conclusion

The Lincoln High School research team used ImageJ and a Java compiler application to develop a program that calculates x- and y-components, in pixels, that allow a data point to be plotted on an orbital solution. Using the program, a known data point was plotted and its accuracy was visually confirmed using the double star WDS 01078+0425. These results proved the validity of the program-based method.

Acknowledgement

We thank the contributors to Adam et al. (2014) for the example for comparison/calibration of our method.

Reference

Adam, Meryl, Roberts, Stephanie, Schenk, Miriam, VanRonk, Carmen, Loayza, Tara, Genet, Russell, Johnson, Bobby, Smith, Thomas C., and Wren, Paul. 2015. First speckle interferometry observation of binary BU 1292. *Journal of Double Star Observations*. 10(4), 245-251.