Photometry of Under-Observed RR Lyrae Variable Stars: GM Orionis

Christine A. Welling

Dickinson College

Follow this and additional works at: http://scholar.dickinson.edu/student_honors

Part of the Stars, Interstellar Medium and the Galaxy Commons

Recommended Citation


This Honors Thesis is brought to you for free and open access by Dickinson Scholar. It has been accepted for inclusion by an authorized administrator. For more information, please contact scholar@dickinson.edu.
Photometry of Under-Observed RR Lyrae Variable Stars: GM Orionis

Submitted in partial fulfillment of honors requirements for the Department of Physics and Astronomy, Dickinson College, by

Christine A. Welling

Advisor: Professor Robert Boyle

Reader: Professor Lars English
Reader: Professor Catrina Hamilton-Drager
Reader: Professor David Jackson
Reader: Professor Windsor Morgan
Reader: Professor Brett Pearson
Reader: Professor Hans Pfister
Reader: Professor Trevor Smith
Reader: Professor Adria Updike

Carlisle, PA
May 15, 2012
Abstract

RR Lyrae stars are pulsating variables that have been used in determining the cosmic distance scale. We have defined under-observed RR Lyrae variable stars to be those RR Lyrae stars that are missing some information in the literature, such as the period, or that have incompatible data in the literature, such as two different light curves. GM Orionis is an RR Lyrae variable star with two inconsistent light curves existent in the literature. I have performed CCD photometry on my own observations of GM Orionis, and have put the photometric measurements on a standard scale in order to compare my measurements to those in the literature. We have observed GM Orionis becoming brighter on average over time, while the amplitude of variability has remained constant.
Acknowledgements

I would like to thank my advisor, Professor Robert Boyle, for guiding me throughout the research process; Professors Catrina Hamilton-Drager, Adria Updike, and Windsor Morgan for their assistance in the details of IRAF and IDL; Professors Lars English and Hans Pfister for translating papers from the original German; Ed Anderson, the staff scientist at Lowell’s National Undergraduate Research Observatory, for explaining how to use the NURO telescope; Dr. Horace Smith at Michigan State University for his expertise on RR Lyrae stars and his research suggestions; Kristen Recine, Matthew Gallentine, Stuart Flury, and Derek Frymark for assisting in data taking during observing runs; and Michael Ryan for invaluable technical support.
Contents

Abstract ................................................. ii

1 Introduction ........................................... 1
  1.1 Historical Background ................................ 1
  1.2 Astrophysical Significance ............................... 2
  1.3 Under-Observed Stars ................................... 3

2 Theory .................................................. 6
  2.1 Stellar Evolution ...................................... 6
  2.2 Opacity .............................................. 8
  2.3 Stellar Pulsation Mechanism ............................. 9
  2.4 Blazhko Effect ........................................ 11
  2.5 Metallicity ........................................... 11
  2.6 Distance Measurements ................................. 12
  2.7 Standard Fields ....................................... 13

3 Observations ............................................ 14

4 Analysis .................................................. 15
  4.1 Data Reduction ....................................... 15
  4.2 Photometry .......................................... 16
  4.3 Standardizing Measurements ............................ 16

5 Discussion ............................................... 19

6 Conclusion ............................................... 24

7 Suggestions for Further Research ......................... 24

A Instructions for Operating the Michael L. Britton Observatory 26
  A.1 Opening the Telescope ................................ 26
  A.2 Observing with the Britton ............................. 27
  A.3 Closing the Britton .................................... 28

B Instructions for Data Reduction .......................... 30
  B.1 General Data Reduction Process ........................ 30
  B.2 I Filter Dithering ...................................... 37
C Instructions for Photometry

C.1 Photometry of Scientific Field .......................................................... 41
C.2 Standard Field Photometry ................................................................. 46

D Instructions for Data Processing and Making Light Curves .......................... 53

E IDL Programs

E.1 centermost ............................................................................................... 56
E.2 centermost_DSON.pro .............................................................................. 58
E.3 mergetest ................................................................................................. 60
E.4 offsets ..................................................................................................... 63
E.5 imcoords ................................................................................................ 65
E.6 mergetwo_NURO .................................................................................... 66
E.7 binstar_B ................................................................................................. 68
E.8 lcds_loop ................................................................................................. 70
E.9 diffmag_B ............................................................................................... 73
E.10 test_lc .................................................................................................... 76
E.11 fig20.pro ............................................................................................... 80
E.12 stand_fit.pro ......................................................................................... 95
E.13 GMOri_stands_B.pro ............................................................................. 99
E.14 GMOri_stands_V.pro ............................................................................. 101
E.15 GMOri_stands_R.pro ............................................................................. 104
E.16 diff_lc.pro ........................................................................................... 107

F Useful Websites for Astronomical Research ........................................... 110

F.1 For Literature Research ........................................................................... 110
F.2 For Weather and Observing ................................................................. 110
## List of Figures

1. Plate VII from Bailey 1902. Example light curves shown, from top to bottom, are class a, class b, and class c cluster variables. .......... 1
2. Example of an RRd type light curve (Debosscher et al. 2009). ........ 3
3. Light curve of GM Ori from Gotz et al. 1957. .................. 4
4. Light curve of GM Ori from Schmidt et al. 1995. ................ 4
6. Plot showing luminosity, radial velocity, and radius as function of time in RR Lyrae stars. This is not to scale, and is only for visualization purposes. ......................................................................................................... 10
7. Field of GM Ori with comparison stars marked. .......................... 17
8. Calculated minus standard V magnitude versus standard V magnitude for the standard fields. Different symbols are used for each star in a given field. .............................................................................................................. 18
9. Light curve of GM Ori in the B filter................................. 20
10. Light curve of GM Ori in the V filter. ................................. 20
11. Light curve of GM Ori in the R filter.............................. 21
12. Minimum and maximum V magnitudes as functions of time........ 22
13. Light curves of V816 Oph from Gotz et al. 1957 (left) and Schmidt et al. 1995 (right)................................................................. 23
14. Final pattern due to fringing in the I filter at NURO. .................... 39
15. Flowchart showing photometry process. ............................... 41
16. Example of format for the standstar file. ............................... 49
17. Example of format for the standobs file............................... 50
18. Example of format for the fstandobs.dat............................ 50
19. Window to interactively fit transformation equations in IRAF........ 52
20. Flowchart showing the data processing method. ....................... 53
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GM Ori data</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Observation data</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Standard field data</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Transformation coefficients</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Standard magnitudes of comparison stars</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Color indices</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Comparison of V Filter</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>File naming conventions</td>
<td>46</td>
</tr>
<tr>
<td>9</td>
<td>Data contained in date filter.sav file</td>
<td>54</td>
</tr>
<tr>
<td>10</td>
<td>IDL Routines</td>
<td>56</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Historical Background

RR Lyrae variables were first discovered in 1893 by Williamina Fleming on photographic plates from the Harvard College Observatory station in Arequipa, Peru. They were found during Solon I. Bailey's survey of globular clusters. Because the first ones were discovered in the globular cluster ω Centauri, RR Lyrae stars were first called cluster variables (Smith 1995).

Bailey divided the variables found in ω Centauri into three classes: a, b, and c. In his 1902 paper, he differentiates between the classifications. Class a stars vary by “a little more than a magnitude,” and have a period between twelve and fifteen hours. The brightness increases to maximum light rapidly, and decreases less rapidly to minimum. He also observed that the brightness was approximately constant at minimum for half of the period (Bailey 1902). This results in a lightcurve that has a sharp turn as it starts to increase brightness. This can be seen in the top two light curves in Fig. 1.

Figure 1: Plate VII from Bailey 1902. Example light curves shown, from top to bottom, are class a, class b, and class c cluster variables.
Class b stars vary by “a little less than a magnitude.” They have a period from fifteen to twenty hours, picking up where the class a stars stop. The brightness of the light curves decreases relatively slowly from maximum, and slows into the minimum brightness. It then climbs more quickly to maximum (Bailey 1902). These light curves look very similar to class a, but they curve more gently into and out of minimum brightness. Class b stars are shown as the middle two light curves in Fig. 1. In contrast to class a, they appear to have a smoother change in brightness.

Class c stars in the cluster vary in a range that “is generally somewhat more than half a magnitude,” (Bailey 1902). The period is between eight and ten hours. This class appears to have more sinusoidally shaped light curves than either class a or b. The bottom two light curves in Fig. 1 are both of class c.

These cluster type variables were initially only found in globular clusters (Smith 1995). However, the same type of star was eventually found outside of globular clusters, making ‘cluster type variable’ a poor name. The brightest of the cluster type variables not in a globular cluster was RR Lyrae, the name we now use for this type of star (Smith 1995).

1.2 Astrophyiscal Significance

RR Lyrae stars are variable because they radially pulsate (Marconi 2009; Smith 1995). Currently, RR Lyrae stars are divided into either two (Smith 1995; Nemec 1991) or three (Feast 1996) classes. The RRab class combines Bailey’s a and b types, due to similarities in the light curves. RRc stars remain a separate type. RRd is the possible third type of RR Lyrae, and an example light curve is shown in Fig 2. The different types of RR Lyraes pulsate in different modes. RRab stars pulsate in the fundamental mode (Feast 1996; Nemec 1991). RRc stars pulsate in the first overtone (Feast 1996; Nemec 1991). RRd pulsate in both the fundamental and first overtone modes (Feast 1996). The period of pulsation is between 0.2 days and 1 day. RRab stars have a longer period (∼0.4-0.9 days) than RRc stars (∼0.2-0.5 days) (Nemec 1991).

RR Lyrae variable stars can be used to study the distance scale. In general, RR Lyrae stars are of similar intrinsic brightness, which we notice by using a Hertzsprung-Russell diagram (discussion in Sec. 2.1). We are able to more accurately determine the absolute magnitude using either a relationship between metallicity\(^1\) and the absolute visual magnitude of the star, or by using a relationship between period and luminosity in the near infrared wavelengths (Marconi 2009). We discuss this further in Sec. 2.6.

\(^{1}\)Metallicity is a measure of the amount of elements that are not hydrogen or helium in a star.
Figure 2: Example of an RRd type light curve (Debosscher et al. 2009).

In the past, RR Lyraes in globular clusters were used to measure the distance to the center of the Galaxy. This was done by Harlow Shapley in 1918. Using a combination of RR Lyrae variables, Cepheid variables, and parallax, Shapley mapped the distances to globular clusters. He surmised that the system of globular clusters should be centered around the center of the Galaxy. He determined that the system of globular clusters was not centered around us, but was centered a large distance away from us (Shapley 1918). This was evidence that we were not at the center of our galaxy.\(^2\)

1.3 Under-Observed Stars

For this project, we have defined under-observed RR Lyrae variable stars to be those RR Lyrae stars that are missing some information in the literature, such as the period, or that have incompatible data in the literature, like two conflicting light curves. The purpose of this research is to add some data to the set of RR Lyrae light curves. In particular, the range of brightness and time in which we observe fills a gap in observations. Brighter variable stars are readily observable by smaller telescopes with charge coupled devices (CCDs). Larger survey programs that will come on line in the future, such as the Large Synoptic Survey Telescope, will map the dimmer stars across the sky over the course of a week. This means that it can take months to obtain enough data to make a light curve for a short period star. We are able to observe stars of intermediate brightness, which fall between the bright stars

\(^2\)As an interesting historical note, Shapley used this as a way to show that so-called spiral nebulae could not be other galaxies, as the distance required for them to appear as small as they do based on the size of our galaxy was huge. We now know that these are, in fact, other galaxies.
observable by smaller telescopes, and the dim stars to be observed by large surveys. Additionally, we are able to observe one star over an entire night, so we are able to more quickly obtain sufficient data for a light curve. This is especially helpful when looking for light curves that change in some way over time.

![Figure 3: Light curve of GM Ori from Gotz et al. 1957.](image)

![Figure 4: Light curve of GM Ori from Schmidt et al. 1995.](image)

We searched for under-observed stars using the Groupe Européen d’Observation Stellaire (GEOS) RR-Lyr database\(^3\) (Le Borgne et al. 2007). GEOS lists only field RR Lyrae stars, those not in globular clusters. This provides us with a less crowded field, making photometry a less difficult process. We only looked for stars in constellations that could easily be observed from Carlisle, PA and Flagstaff, AZ during the fall and

\(^{3}\)Available at [http://rr-lyr.ast.obs-mip.fr/dbrr/dbrr-V1.0_0.php](http://rr-lyr.ast.obs-mip.fr/dbrr/dbrr-V1.0_0.php).
winter. After making a list of potential candidates, we chose the star GM Orionis (GM Ori) as our first candidate. We also contacted Dr. Horace Smith for suggestions of other stars that could be interesting to observe.

GM Ori has two light curves in the literature. The first, based on photographic photometry using photographic plates, is from Gotz et al. (1957), and is shown in Fig. 3. Observations were made at the Sonneberg Observatory in Sonneberg, Germany. The visual magnitude varies from approximately 13.2 to 13.6 with an amplitude of 0.4m. The scatter in the light curve is 0.05m. The shape of the light curve suggests that the star is type RRc, and it is listed as such in the paper.

The second light curve comes from Schmidt et al. (1995), and is shown in Fig. 4. Observations were made at the Behlen Observatory in Lincoln, Nebraska with a CCD. In this light curve, the standard V magnitude ranges from 12.4 to 12.9, with an amplitude of 0.5 magnitudes. We see that the light curves are inconsistent, both in average brightness of the star, and in the amplitude of variability.

Additionally in the literature, Hoffmeister (1943), published a table of 213 new variable stars discovered at the Sonneberg Observatory. Among these is GM Ori (listed in the paper as 98.1943 Orio). The magnitudes are listed as 13.5 and 14, and the star is categorized as short period. There is no light curve in the paper. No error in magnitude is given, so we assume the same scatter as in Gotz et al. (1957).

Table 1 provides data from the literature on GM Ori that is consistent throughout. This includes the period, a Julian date of maximum, position, and the type of RR Lyrae.

<table>
<thead>
<tr>
<th>Period</th>
<th>0.3905135 days$^{a,b}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Julian date of maximum</td>
<td>8580.836$^a$</td>
</tr>
<tr>
<td>Right Ascension</td>
<td>06:23:51.4$^a$</td>
</tr>
<tr>
<td>Declination</td>
<td>+17° 00’ 47.5”$^a$</td>
</tr>
<tr>
<td>Type</td>
<td>RRc$^b$</td>
</tr>
</tbody>
</table>

Table 1: Data from the literature on GM Ori.

$^a$Schmidt et al. (1995)

$^b$Gotz et al. (1957)
2 Theory

2.1 Stellar Evolution

The Hertzsprung-Russell diagram (HR-diagram) plots some measure of a star’s effective temperature against some measure of its luminosity. Figure 5 shows an example of an HR-diagram. In this case, absolute magnitude is the measure of luminosity, and spectral class is the measure of effective temperature. Luminosity measurements can be plotted as absolute magnitude, apparent magnitude, and luminosity. Temperature can be plotted in many ways. Our example uses spectral class as a temperature measurement. Another temperature measurement that can be used is the color index, which is the difference in magnitude between two bandpasses, such as B-V or V-R. This can be used to tell the temperature of the star based on the peak wavelength of the star using a black body curve. The color index can also be used to characterize interstellar reddening, which occurs because the dust scatters more blue light than red light. Positive values of the color index B-V mean the star is redder and negative values of the color index B-V mean the star is bluer (Bohm-Vitense 1992).

The example HR-diagram in Fig. 5 also shows the locations of various types of stars. By knowing where a star is located on the HR-diagram, astronomers are able to determine what type of star it is (Carroll and Ostlie 2007), and then automatically know other physical and evolutionary characteristics.

Stellar evolution describes the structural changes that happen to a star throughout its lifetime. All stars begin their lives somewhere on the main sequence. The main sequence runs from the upper left to lower right of the HR-diagram, and is labeled in Fig. 5. The most massive and luminous stars are at the upper left. Stars are less luminous and less massive further toward the lower right of the HR-diagram. Stars spend most of their lives on the main sequence because this phase of stellar evolution takes the most amount of time. While on the main sequence, stars fuse hydrogen into helium in their cores (Carroll and Ostlie 2007; Zeilik et al. 1992). High mass stars use the carbon-nitrogen-oxygen (CNO) cycle, which turns four hydrogen nuclei into one helium nucleus using carbon, nitrogen and oxygen as catalysts, for fusion. Low mass stars use the proton-proton (p-p) chain, which also turns four hydrogen nuclei into one helium nucleus, to produce fusion reactions in their cores (Carroll and Ostlie 2007).

Stars of different masses evolve in different ways. The most massive stars quickly fuse hydrogen and evolve off of the main sequence. Lower mass stars take a longer time to fuse the hydrogen in their cores, and so spend a longer time on the main sequence. Stars that are approximately 0.8 $M_\odot$ on the main sequence eventually
all stars eventually evolve off the main sequence when they stop fusing hydrogen. In low mass stars, when the hydrogen in the core has been depleted, fusion in the core stops. Fusion of hydrogen then only occurs in a shell around the core. The core begins to contract, releasing gravitational potential energy. The released energy causes the envelope of the star to expand. This process moves the star off the main sequence and into the red giant branch of the HR-diagram (Zeilik et al. 1992; Carroll and Ostlie 2007). This is shown in the area marked “Giants” in Fig. 5.

As the star reaches the red giant branch, the core contracts, increasing its temperature and density. The density of the star’s core becomes high enough to create

---

**Figure 5:** Example of a Hertzsprung-Russell diagram ([http://chandra.harvard.edu/edu/formal/variable_stars/bg_info.html](http://chandra.harvard.edu/edu/formal/variable_stars/bg_info.html)).

 evolve into RR Lyrae stars (Smith 1995), so we will consider only those lower mass stars in the rest of the discussion.
a degenerate gas of electrons (Zeilik et al. 1992). Electrons must occupy the lowest possible energy levels when the gas reaches a sufficiently high temperature. Electrons cannot occupy the same quantum state, because of the Pauli exclusion principle, and become piled on top of one another in energy states. The electrons cannot get any closer together, and this creates the electron degeneracy pressure (Carroll and Ostlie 2007). Having a degenerate gas of electrons means that the ideal gas law no longer applies (Bohm-Vitense 1992), and that an increase in temperature does not increase pressure (Zeilik et al. 1992). The electron degeneracy pressure enables the star to balance the gravitational force inward on the core (Zeilik et al. 1992).

The core continues to contract, increasing the temperature. At some point, the temperature in the core becomes high enough to start helium fusion via the triple-\(\alpha\) process, which converts three helium nuclei into carbon. Since the core is degenerate, when the temperature increases, the pressure does not increase, and thus the core does not expand. Instead, the rate of helium fusion increases, resulting in a thermal runaway called helium flash. Once the temperature is sufficiently high, the electrons are no longer degenerate, and the core expands and cools (Zeilik et al. 1992). As the star cools, the luminosity of the star also decreases (Carroll and Ostlie 2007).

Now, the star proceeds to the horizontal branch of the HR-diagram (Smith 1995). Here, the star fuses helium in its core and hydrogen in a shell around the core (Carroll and Ostlie 2007; Marconi 2009; Smith 1995). It is at this stage in stellar evolution that we may encounter an RR Lyrae star.

The instability strip is a nearly vertical path on the right hand side of the HR-diagram. Pulsating stars tend to be located in this part of the diagram (Carroll and Ostlie 2007). RR Lyrae stars are located at the intersection of the horizontal branch and the instability strip on the HR-diagram (Smith 1995; Zeilik et al. 1992). We see this location in Fig. 5. From the HR-diagram, we can tell that RR Lyrae stars are pulsating, giant stars.

### 2.2 Opacity

The intensity of light that is able to pass through some material depends on the opacity of the material. For stars, we particularly care about the opacity in the stellar interior. Opacity comes from four main sources. All four deal with interactions between photons and particles in the material (Carroll and Ostlie 2007).

Bound-bound transitions occur when an electron bound to atom or ion changes energy levels. The electron absorbs a photon of a specific wavelength to move up an energy level. It then emits a photon(s) in a random direction, scattering the light. In this process, light is unable to go straight through the material, and is instead
bounced around (Carroll and Ostlie 2007). Bound-free absorption causes an electron to be ionized from the atom to which it was previously bound. In this case, the electron absorbs the photon to gain energy. Then, the photon no longer exists to leave the star (Carroll and Ostlie 2007). Free-free absorption allows a free electron near an ion to absorb a photon (Carroll and Ostlie 2007). In electron scattering, a photon is scattered by an electron. The electron can be either free or loosely bound (Carroll and Ostlie 2007).

In general, it is difficult to calculate each of these opacities individually. Instead, we use the Rosseland mean opacity

\[ \kappa = \kappa_0 \rho^n T^{-s} \]  

(1)

where \( \kappa_0 \) is a constant that depends on the stellar composition, \( \rho \) is the density, \( T \) is temperature, and \( s \) and \( n \) are numbers. In regions of high ionization, \( s \) can become small or negative, increasing the opacity (Smith 1995; Carroll and Ostlie 2007).

We consider a Kramers opacity law

\[ \kappa \sim \frac{\rho}{T^{3.5}} \]  

(2)

where \( \rho \) is density and \( T \) is temperature. The equation comes from the functional form for the average bound-free and free-free opacities, which both depend on density and pressure in this way (Carroll and Ostlie 2007).

### 2.3 Stellar Pulsation Mechanism

The pulsation mechanism in RR Lyrae stars is the \( \kappa \)-mechanism. The \( \kappa \)-mechanism is based on the opacity \( \kappa \) (Baker and Kippenhahn 1962). This mechanism can be thought of as a valve that traps and releases radiation (Carroll and Ostlie 2007).

The \( \kappa \)-mechanism works as follows. At maximum compression of the star, the opacity in some layer of the star increases. This traps radiation inside the opaque layer (Baker and Kippenhahn 1962). The trapped radiation provides an outward restoring force, which causes the star to expand. As the star expands, the layer becomes less opaque, and the radiation escapes. Then the layer falls back down and the opacity increases again (Carroll and Ostlie 2007). It is important that the opacity increases as the layer falls back down, or contracts. This gives the system a push to continue pulsations. A decreasing opacity during contraction would damp the pulsations, and they would not continue (Bohm-Vitense 1992).

Partial ionization zones are layers within the star where a species has some atoms that are ionized, while other atoms have not been ionized. This occurs because the
temperature is near that required to ionize the species. In RR Lyrae stars, the valve mechanism is thought to be located in the second ionization region of helium (Baker and Kippenhahn 1962; Marconi 2009; Smith 1995), or in the helium partial ionization zone (Zeilik et al. 1992).

When the partial ionization zone is compressed, the temperature of the zone does not increase. Instead, more of the species in the partial ionization zone is ionized. Because the gas has been compressed, the density increases. Using Kramers opacity (eq. 2), the opacity of the layer increases (Carroll and Ostlie 2007). We know from Sec. 2.2 that ionized regions increase opacity. Thus, it makes sense that increasing the ionization increases the opacity. This increased opacity traps radiation inside the partial ionization zone.

The trapped radiation provides an outward force on the opaque layer, pushing it out. The expanded layer has a decreased density. However, the temperature will not change as much as expected because the ionized layer experiences some recombination of ions and electrons. Thus, the opacity decreases (Carroll and Ostlie 2007). Again, from Sec. 2.2, we know that fewer ions decrease opacity. With a decreased opacity, the light is able to escape. The layer falls back, and the process repeats.

Figure 6: Plot showing luminosity, radial velocity, and radius as function of time in RR Lyrae stars. This is not to scale, and is only for visualization purposes.

The light curve of an RR Lyrae variable star can be interpreted to determine
what is happening in the star’s pulsation (Nemec 1991). At the point of maximum luminosity in the light curve, the expansion of the star is most rapid. The expansion continues as the star dims. When the star has reached its maximum radius, it has zero radial velocity. Then the star starts contracting, making the star's radial velocity positive \(^4\). At minimum luminosity, the contraction of the star is most rapid. The star then begins to brighten because the star’s interior has a higher temperature. The star continues to contract until it reaches a maximum compression where it has no radial velocity. The star then expands, giving it a negative radial velocity. The process then repeats. Figure 6 shows a diagram of this process. Note that the diagram is not necessarily the correct shape, but is meant to illustrate how the luminosity, radial velocity, and radius relate to each other.

Like any oscillating system, a pulsating star can pulsate in different modes. These radial modes are the standing sound waves in the star, much like the standing waves in an organ pipe (Carroll and Ostlie 2007). The classes of RR Lyrae star pulsate in different ways. RRab stars pulsate in the fundamental mode. RRc stars pulsate in the first overtone (Marconi 2009). RRd stars pulsate in a combination of the fundamental and first overtone modes (Smith 1995; Feast 1996).

### 2.4 Blazhko Effect

While the pulsation of an RR Lyrae star is periodic, not all are perfectly periodic. The Blazhko effect is a modulation of the amplitude or period of the RR Lyrae star (Dziembowski 2005). This modulation occurs over a longer time scale than the period, on the order of 20-200 days (Feast 1999). The cause of this modulation is not well understood. Theories that have been put forward include the excitation of nonradial modes (Dziembowski 2004), a strong magnetic dipole field (Shibahashi 2000), or turbulent convective variations in the star (Molnar et al. 2012). However, none of these theories has successfully accounted for all observations.

### 2.5 Metallicity

In astronomy, all elements that are not hydrogen or helium are broadly considered metals. Metals are formed in both stellar interiors and supernova explosions. Metallicity is a measure of the metal content of a star.

Typically, stars can be classified as some spectral type based on the strength of different spectral lines. In RR Lyrae stars, the strength of these spectral lines result

\(^4\)By convention radial velocity is positive when moving away from the observer, and negative when moving toward the observer.
in different spectral classifications. In order to take this difference into account, a spectral index is defined as

\[ \Delta S = 10[Sp(H) - Sp(K)] \]  

where \( Sp(H) \) and \( Sp(K) \) are the spectral types based on the strength of the hydrogen Balmer lines and the calcium K-line, respectively (Preston 1959; Smith 1995). Preston interpreted the spectral index as an indicator of metal abundance (i.e., metallicity) in an RR Lyrae type star. Stars with \( \Delta S = 0 \) have a metal abundance similar to the sun. Stars with higher \( \Delta S \) have fewer metals (Smith 1995).

Metallicity can also be characterized as the ratio of iron to hydrogen

\[ [Fe/H] = \log(\text{iron/hydrogen})_* - \log(\text{iron/hydrogen})_\odot \]  

so that the ratio is compared to the ratio of iron to hydrogen in the sun (Smith 1995). \([Fe/H]\) for the sun is 0 by definition. Metal-poor stars have negative values for \([Fe/H]\) (Smith 1995).

In RR Lyrae stars, there is a correlation between \( \Delta S \) and period. In RRab stars, the average period increases with increasing \( \Delta S \). That is, the lower the metallicity, the longer the average period of the star. This correlation is not present in RRc stars (Smith 1995).

Additionally, there is a relationship between metallicity and absolute visual magnitude\(^5\) (DeMarque et al. 2000; Marconi 2009). This is typically written, assuming a linear relationship, as

\[ M_V(RR) = \mu[Fe/H] + \gamma \]  

where \( \mu \) and \( \gamma \) are constants determined by observation (DeMarque et al. 2000). Thus, if we determine the metallicity, the absolute magnitude can be determined.

### 2.6 Distance Measurements

RR Lyrae stars can be used to measure distance. This can be done in two different ways. The first is a relationship between the star’s pulsation period and luminosity. The second is a relationship between absolute visual magnitude and metallicity (DeMarque et al. 2000; Marconi 2009) (see discussion in Sec. 2.5).

In general, distance is determined using the distance modulus

\[ m - M = 5 \log_{10} \left( \frac{d}{10 \, \text{pc}} \right) + A_V \]  

\(^5\)The magnitude system is a logarithmic measure of flux. Apparent magnitude is what we observe. Absolute magnitude is defined to be the apparent magnitude that a star would have if it were at a distance of 10pc (Birney et al. 2006)
where $M$ is absolute magnitude, $m$ is apparent magnitude, $d$ is distance in parsecs, and $A_\nu$ is extinction\textsuperscript{6} (Carroll and Ostlie 2007). Absolute magnitude and luminosity are related by

$$M_{bol} = M_{bol,Sun} - 2.5 \log_{10} \left( \frac{L}{L_\odot} \right)$$

(7)

where $M_{bol}$ is the absolute bolometric magnitude of the star, $M_{bol,Sun}$ is the bolometric magnitude of the Sun, $L$ is the luminosity of the star, and $L_\odot$ is the solar luminosity (Carroll and Ostlie 2007). Bolometric magnitude is the magnitude over all wavelengths.

The luminosities, and thus absolute magnitudes, of RR Lyrae stars are similar. We can see this by looking at an HR-diagram, and noticing that all of these stars are plotted along a horizontal line. This makes RR Lyrae stars good objects to use for distance measurements (Carroll and Ostlie 2007; Zeilik et al. 1992). We can determine their absolute magnitude using eq. 7. The absolute magnitude measurement can be refined by metallicity if a spectrum of the star has been taken. Then, we can measure the apparent magnitude of a star, and use eq. 6 to determine the distance to the star.

### 2.7 Standard Fields

Standard fields are used in photometry to calibrate measurements of magnitude. This calibration is crucial as it allows us to accurately compare our measurements to those made in the literature. Standard fields include stars for which we know their magnitudes in the standard set of filters\textsuperscript{7}. These standards are found in catalogs, and can be looked up. By observing standard stars over the course of a night, transformation coefficients can be determined. These coefficients are then applied to science images to calibrate the measurements made.

The extinction coefficient accounts for atmospheric extinction. Earth’s atmosphere makes the light we observe dimmer than what we would observe at the top of the atmosphere. The amount that the star is dimmed is dependent on how much atmosphere the light must travel through in order to reach the detector. This amount of atmosphere is called the air mass. We define one air mass to be the smallest thickness of atmosphere the light from a star can pass through. That is, one air mass is the thickness of the atmosphere from the observer directly up to zenith. We then

\textsuperscript{6}Extinction is caused by interstellar gas and dust between observer and object that absorbs or scatters light.

\textsuperscript{7}Generally, the filters used are the Bessel filters, which are made for CCDs to emulate the Johnson–Cousins UBVRI system for photoelectric photometry (Birney et al. 2006)
define a general airmass as

\[ X = \sec z \]  

(8)

where \( z \) is the angle from zenith to the direction of the star (Birney et al. 2006).

The color term takes into account atmospheric effects on the light observed due to the color of light being emitted by the star. We define a color index to be the difference in magnitudes of a star in two different passbands, such as B-V (Birney et al. 2006). In order to properly calibrate the color term, observations must be made of standard stars with different color indices (Birney et al. 2006).

In general, each instrument has a different zero point for its measurements. The zero-point offset provides a correction for the difference in zero points between the instrument being used to take the current measurement and the instrument used to measure the standard star (Birney et al. 2006). To confirm the calibration of a zero point, we need to observe standard stars with a range of magnitudes.

For data taken in the B, V, and R filters, the equations to be solved to determine the transformation coefficients are:

\[ m_B = ((B - V) + V) + b_1 + b_2 \times X_B + b_3 \times (B - V) \]  

(9)

\[ m_V = V + v_1 + v_2 \times X_V + v_3 \times (B - V) \]  

(10)

\[ m_R = (V - (V - R)) + r_1 + r_2 \times X_R + r_3 \times (V - R) \]  

(11)

where \( m_B \), \( m_V \), and \( m_R \) are the instrumental magnitudes; \( ((B - V) + V) \), \( V \), and \( (V - (V - R)) \) are the standard magnitudes; \( B - V \) and \( V - R \) are the color indices of the star; and \( X_B \), \( X_V \), and \( X_R \) are the airmasses during the observations made of the standard field during each filter. We solve for \( b_1 \), \( v_1 \), and \( r_1 \), the zero-point offset in each filter; \( b_2 \), \( v_2 \), and \( r_2 \), the extinction coefficient for each filter; and \( b_3 \), \( v_3 \), and \( r \), the color term for each filter (Massey and Davis 1992). This solution is done interactively in the Image Reduction and Analysis Facility (IRAF). Detailed instructions for the fitting procedure can be found in Section C.2.

3 Observations

Observations were made at Lowell Observatory, under the sponsorship of the National Undergraduate Research Observatory (NURO) in Flagstaff, AZ, and at the Michael L. Britton Observatory (Britton) in Carlisle, PA. NURO has a 31-inch reflecting telescope with stock Bessell filters in UBVRI. The CCD being used on the telescope is NASAcam. The read noise is 6.2 electrons, and the gain is 2.15 electrons per count.
The Britton houses a 24-inch reflecting telescope, also with a Bessel set of UBVRI filters. The CCD used is an Apogee U6, with a size of 1024x1024 pixels. The read noise is 5.8 electrons, and the gain is 1.3 electrons per count. At both observatories, we have only taken images in BVRI filters.

<table>
<thead>
<tr>
<th>Observatory</th>
<th>UT Date of Observation</th>
<th>Used in Light Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURO</td>
<td>01/19/2011</td>
<td>yes</td>
</tr>
<tr>
<td>NURO</td>
<td>01/20/2011</td>
<td>yes</td>
</tr>
<tr>
<td>NURO</td>
<td>01/21/2011</td>
<td>yes</td>
</tr>
<tr>
<td>Britton</td>
<td>02/10/2011</td>
<td>no</td>
</tr>
<tr>
<td>Britton</td>
<td>02/23/2011</td>
<td>no</td>
</tr>
<tr>
<td>Britton</td>
<td>03/28/2011</td>
<td>no</td>
</tr>
<tr>
<td>Britton</td>
<td>11/06/2011</td>
<td>no</td>
</tr>
<tr>
<td>Britton</td>
<td>11/08/2011</td>
<td>no</td>
</tr>
<tr>
<td>NURO</td>
<td>11/16/2011</td>
<td>yes</td>
</tr>
<tr>
<td>NURO</td>
<td>11/17/2011</td>
<td>yes</td>
</tr>
<tr>
<td>NURO</td>
<td>11/18/2011</td>
<td>yes</td>
</tr>
<tr>
<td>Britton</td>
<td>02/06/2012</td>
<td>no</td>
</tr>
<tr>
<td>Britton</td>
<td>02/20/2012</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 2: Observation data for GM Ori using NURO and the Britton.

Table 2 lists all observations made of GM Ori, including the observatory, the date of observation, and whether data from that night is being used. We have only used data from the NURO observatory in making light curves. We chose to process NURO data first because the data from NURO has a better signal to noise ratio than data from the Britton, and because the observing conditions in Flagstaff are better than those in Carlisle. Further, there was not sufficient time to process both sets of data.

4 Analysis

4.1 Data Reduction

Data reduction prepares raw images to be used for scientific research. It corrects for noise in images due to inhomogeneities within the CCD chip, artifacts on the optics, and varying temperatures across the CCD. Data reduction is done using IRAF and Interactive Data Language (IDL) procedures. Detailed step-by-step instructions for data reduction are available in Appendix B.
4.2 Photometry

Photometry measures flux being emitted by a star or some other object. An aperture is placed on the star to measure the light being gathered from that area. However, some of the light collected is due to the nonzero sky background. An estimation of the sky background is made by drawing an annulus around the star. The background is then removed so that only light coming from the star will be counted (Howell, 2000).

During the photometry process, we follow both the object of interest and other stars in the field, called comparison stars. The comparison stars should be near the center of the field and have a range of brightnesses. Most importantly, they cannot be variable stars. The magnitudes of the comparison stars are converted to fluxes. We use the fluxes to make one synthetic comparison star with flux equal to the average flux of the comparison stars. The flux of the synthetic star is then converted back to a magnitude\(^8\). The magnitude of the synthetic comparison star is used as a baseline to measure the changing brightness of the variable star.

Figure 7 shows the field of GM Ori. GM Ori is marked with a vertical line and a horizontal line. All comparison stars used have been circled and labeled with a letter. These stars were photometered in alphabetical order.

Instructions for general photometry can be found in Appendix C.1. Steps for data processing can be found in Appendix D. Copies of all IDL programs used in Appendix E.

4.3 Standardizing Measurements

In order to make meaningful measurements, the relative magnitude must be calibrated by using a standard field (see discussion in Sec. 2.7). This process is carried out using IRAF and IDL.

The observations used for determining the calibration were carried out on one photometric night of high quality, January 21, 2011. We have used the Landolt standard fields RU 149 and PG0231+051 (PG 0231 in the future) for calibration. Table 3 gives relevant data for the standard fields we have used. The second column lists the number of stars in the field. The third column provides the minimum V magnitude of the standard stars. The fourth column lists the maximum V magnitude of the standard stars. Column five provides the range of airmasses at which the field was observed on the photometric night. From the table, we see that we have observed

---

\(^8\)Magnitudes are a logarithmic scale, so we cannot simply compute the average magnitude from the magnitudes given. We must convert to a non-logarithmic scale in order to find the average for the set of stars. We convert back to magnitudes because all of our measurements are in magnitudes.
standards both brighter and dimmer than GM Ori. Additionally, by observing at a range of airmasses, we will obtain a better fit for the transformation coefficients.

<table>
<thead>
<tr>
<th>Field</th>
<th>Number of Stars</th>
<th>Minimum V</th>
<th>Maximum V</th>
<th>Range of Airmass</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG 0231</td>
<td>5</td>
<td>14.735</td>
<td>12.772</td>
<td>1.15-1.94</td>
</tr>
<tr>
<td>RU 149</td>
<td>8</td>
<td>14.495</td>
<td>11.480</td>
<td>1.23-1.46</td>
</tr>
</tbody>
</table>

Table 3: Pertinent data for standard fields used for calibration. The third and fourth columns provide the minimum and maximum V magnitudes of all standard stars in the field. The fifth column provides the range of airmasses at which we observed the field.

The procedures for calibration using standard fields, given in Sec. C.2, were followed to determine transformation coefficients. The values we determined for the coefficients are given in Table 4. For each fit, the scatter was 0.01.

In order to check the accuracy of our fit, we plot the calculated V magnitude of the standard stars minus the standard V magnitude against the standard V magnitude, shown in Fig. 8. Stars from the field PG 0231 are shown in purple, and stars from the field RU 149 are shown in red. A different symbol is used for each star in a given field. The standard deviation of the differences between the calculated and standard V magnitude is 0.05. We expect to see a horizontal line if we have correctly
Table 4: Values determined for the transformation coefficients.

determined the transformation coefficients. We see that sets of observations form horizontal lines, but there is scatter over the course of a night's observations. This means that the photometric quality of the night was not as high as we expected, and that the transparency changed over the course of the night.

Figure 8: Calculated minus standard V magnitude versus standard V magnitude for the standard fields. Different symbols are used for each star in a given field.

The calculated transformation coefficients were applied to the instrumental magnitudes measured on the photometric night. We took the calculated standard magnitudes of the comparison stars to be their accepted magnitude on all other nights. In this way, we are able to calibrate all of our data, even though not all nights of observation were photometric. Table 5 lists the standard magnitude of each comparison star for each filter as calculated for the night of January 21. We notice that all of the comparison stars are brightest in the red filter.
### Table 5: Standard magnitudes of comparison stars in GM Ori field calculated by stand_fit.pro.

<table>
<thead>
<tr>
<th>Comparison Star</th>
<th>B</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>13.908</td>
<td>12.167</td>
<td>11.093</td>
</tr>
<tr>
<td>B</td>
<td>14.517</td>
<td>13.923</td>
<td>13.489</td>
</tr>
<tr>
<td>C</td>
<td>13.754</td>
<td>13.319</td>
<td>12.965</td>
</tr>
<tr>
<td>D</td>
<td>12.705</td>
<td>12.292</td>
<td>11.928</td>
</tr>
<tr>
<td>F</td>
<td>14.768</td>
<td>13.320</td>
<td>12.395</td>
</tr>
<tr>
<td>G</td>
<td>12.057</td>
<td>11.892</td>
<td>11.699</td>
</tr>
<tr>
<td>H</td>
<td>13.149</td>
<td>12.035</td>
<td>11.302</td>
</tr>
<tr>
<td>I</td>
<td>12.724</td>
<td>12.317</td>
<td>11.982</td>
</tr>
<tr>
<td>J</td>
<td>12.457</td>
<td>12.287</td>
<td>12.078</td>
</tr>
<tr>
<td>K</td>
<td>13.324</td>
<td>13.084</td>
<td>12.798</td>
</tr>
</tbody>
</table>

#### 5 Discussion

Light curves of GM Ori have been generated in B, V and R filters, shown in Fig. 9, Fig. 10, and Fig. 11, respectively. We have noticed that the phase does not appear to be well aligned. We tried changing both the last digit of the period given, and adding an extra digit to the period given in order to achieve a more well phased light curve. However, we could not improve the phase. The offset may be due to using the Julian date to calculate phase instead of the heliocentric Julian date. The heliocentric Julian date takes into account the travel time for light as the Earth orbits the sun. Perhaps by using this quantity, we would be able to remove the slight phase offset. Additionally, three data points were excluded from the R filter. All three had a comparison star whose measured instrumental magnitude was significantly different from all other measurements throughout the rest of the observations. We attribute this to the location of GM Ori relative to the full moon during the excluded observations.
Figure 9: Light curve of GM Ori in the B filter.

Figure 10: Light curve of GM Ori in the V filter.
Additionally, we have calculated the color indices B-V and V-R for both the minimum and maximum. This was done by fitting a polynomial to points around phase of 0.5 for minimum and 1.0 for maximum. These values are listed in Table 6. The typical B-V color of an RR Lyrae star is in the range of 0.17-0.42 magnitudes (Nemec 1991). The V-R color of an RR Lyrae star at minimum should be ~0.28 (Kunder et al. 2009). We have observed greater B-V and V-R than expected. This could indicate that our observations have been reddened by interstellar dust.

<table>
<thead>
<tr>
<th></th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-V</td>
<td>0.81</td>
<td>0.61</td>
</tr>
<tr>
<td>V-R</td>
<td>0.51</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 6: Color indices of GM Ori at minimum and maximum.

Table 7 lists the minimum and maximum V magnitudes that were found in the literature and that we have measured. The table from Hoffmeister (1943) only lists a range of magnitudes and does not provide a light curve. The magnitudes from Gotz et al. (1957) are photographic. Magnitudes from Schmidt et al. (1995) are on a standard scale using Landolt standards. Figure 12 shows the minimum and maximum
Minimum $V$ magnitudes from Hoffmeister 1943, Gotz et al. 1957, Schmidt et al. 1995, and this project

<table>
<thead>
<tr>
<th></th>
<th>Minimum $V$</th>
<th>Maximum $V$</th>
<th>Amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hoffmeister 1943</td>
<td>14</td>
<td>13.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Gotz et al. 1957</td>
<td>13.6</td>
<td>13.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Schmidt et al. 1995</td>
<td>12.9</td>
<td>12.4</td>
<td>0.5</td>
</tr>
<tr>
<td>Welling 2012</td>
<td>12.55</td>
<td>12.05</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 7: Minimum and maximum $V$ magnitudes from Hoffmeister 1943, Gotz et al. 1957, Schmidt et al. 1995, and this project

$V$ magnitudes plotted as functions of time. We see that the average magnitude, minimum magnitude, and maximum magnitude appear to be getting steadily brighter.

Figure 12: Minimum and maximum $V$ magnitudes as functions of time.

We assume that the difference between Hoffmeister (1943) and Gotz et al. (1957) should be real because both light curves were made in the same facilities using the same techniques. We compared the Gotz et al. (1957) and Schmidt et al. (1995) papers to determine whether an offset between photographic magnitudes of the first and the CCD magnitudes of the second could be the cause of this phenomenon. We determined that a second object, V816 Oph, also had a light curve in both papers. These light curves are shown in Figure 13, with Gotz et al. (1957) on the left and Schmidt et al. (1995) on the right. We see that both light curves range in magnitude from 13 to 11.8. Additionally, this shows that the photographic plates used by Gotz
et al. (1957) were similar to the V filter. Because we see no offset between the measured magnitudes, we conclude that the observed brightening we have seen in GM Ori between the 1957 and 1995 measurements is real.

Figure 13: Light curves of V816 Oph from Gotz et al. 1957 (left) and Schmidt et al. 1995 (right).

We will consider some possible reasons for the observed brightening. First, we will consider physical mechanisms within the star that may cause changes in the light curve. Stellar evolution will eventually cause the star to become brighter. However, this happens on a time scale of at least several million years. Because our length of time is only 55 years, stellar evolution seems to be an unlikely explanation.

The light curves of RR Lyrae stars that exhibit the Blazhko effect change over longer time scales. The Blazhko effect is observed as period and amplitude modulations of the light curve. We do not observe either of these, so we assume that the Blazhko effect is not the cause of brightening.

We may be able to explain the observed brightening of GM Ori as intervening dust moving from between us and GM Ori. Using the Infrared Science Archive (IRSA) Galactic Dust Reddening and Extinction catalog, at the coordinates of GM Ori we find that $E(B-V)$, the excess B-V that we see due to dust, is 0.595. We determine the intrinsic B-V using

$$E(B - V) = (B - V)_{\text{obs}} - (B - V)_{\text{int}}$$  \hspace{1cm} (12)

where $(B - V)_{\text{obs}}$ is the observed B-V, and $(B - V)_{\text{int}}$ is the intrinsic B-V. We use the middle value of our observed B-V, 0.71, as our observed quantity. Then, if all of the dust measured by is present, the intrinsic B-V for GM Ori is 0.12. This represents the maximum amount of dust that could be between us and GM Ori, and so the maximum amount of reddening we could observe. Based on the estimate of maximum reddening
that we could observe, it is possible that decreasing dust obscuration could be causing GM Ori to appear brighter.

6 Conclusion

We have made new measurements of the RRc star GM Ori, and created light curves in B, V and R. Our data has been compared to that from Hoffmeister (1943), Gotz et al. (1957) and Schmidt et al. (1995). We have noticed a trend in the measurements of GM Ori in the V filter. GM Ori appears to be getting brighter at a rate of approximately 1 magnitude in 50 years, or 0.02 magnitudes per year.

RR Lyrae stars have a generally constant brightness, making them good for distance measurements. Because of this, it is difficult to think of a physical mechanism intrinsic to the star that is causing it to appear brighter. Stellar evolution occurs on time scales of millions of years, which is significantly longer than what we have observed. We do not observe a change in period or amplitude, so we do not believe this can be caused by the Blazhko effect.

One plausible explanation is that a dust cloud is located between us and GM Ori. A decrease in the amount of dust in this region could be responsible for the observed brightening. This would also account for the reddened color indices observed.

7 Suggestions for Further Research

Based on the results of this research, it would be useful to observe GM Ori again in five years to determine whether the star will continue to become brighter on average. Astronomical archives of old photographic plates that can be converted to a standard measurement could be found. These could add to the four data points that we currently have.

The possibility of intervening dust should be further explored in more detail. This could include determining how much dust is there now, and how much dust has left our line of sight. Also, determining a distance to GM Ori would help provide limits to the amount of dust.

Additionally, other under-observed stars exist. We have taken data on the star NSVS 11647538. Kinemuchi et al. (2006) obtained data through the ROTSE telescope on RR Lyrae variable stars. However, they were unable to successfully determine a period for the star. Data that we have taken can be processed in order to determine an accurate period.
Research Acknowledgements

This research has made use of the NASA/ IPAC Infrared Science Archive, which is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.
A Instructions for Operating the Michael L. Britton Observatory

This section details step-by-step instructions for opening, observing with, and closing the Michael L. Britton Observatory at Dickinson College.

A.1 Opening the Telescope

1. Open the dome and doors underneath floor
   (a) Plug in curly cord and open dome. When finished UNPLUG CURLY CORD!!!
   (b) Plug in dome rotation motor (near computer on desk in dome).

2. Go back to warm room and turn on the telescope control computer.
   (a) Log on as DFM

3. Set time on computer using the clock.
   (a) Need 3 rings on service bar on the digital clock (on the shelf)

4. Turn on the drivers. There are 2 switches that should be turned on in order:
   (a) MTR Driver Chassis
   (b) Drives

5. Open up the DFM shortcut on the desktop of the control computer. This is the telescope control panel.
   (a) HA should be 0
   (b) Sidereal time and RA should match

6. Open up The Sky program from the desktop. Under Telescope Menu select Link, then Establish Link

7. Turn off the HALT MOTORS button. The button will be out all the way.

8. Go back to dome and take off all the caps from the telescopes (finder and 24-inch). Plug in camera.
   (a) Use power strip on side of telescope to turn on power to CCD
(b) Make sure the eyepiece is slid all the way to the left.

9. Turn on data taking computer.
   (a) Set time.
   (b) Start ICS on data-taking computer
   (c) Plug in white box next to computer

A.2 Observing with the Britton

1. Take flats, biases, and darks.
   (a) CCD temperature should be at least -20 (-30-ish is ideal)
   (b) Flats
       i. Should have 20,000 counts
       ii. 1 s
       iii. Need to take 5 in each filter
   (c) Biases
       i. 1200 counts is typical
       ii. 0 s
       iii. 25 biases
   (d) Darks
       i. Same exposure length as object image
       ii. 20 darks

2. Turn on tracking, dome track, auto dome

3. Move to a bright star near the zenith. You do this with The Sky program
   (double-click on the object, then click on the telescope icon to slew).

4. Center the bright star in the 24-inch. (It won't be centered in the finder scope.
   There are details in the warm room log.) Once the bright star is centered up in
   the 24-inch, initialize the coordinates with The Sky program.
   (a) Center the object in the telescope with the eyepiece set at 8.35 inches.
   (b) Move the eyepiece back to the left.
   (c) Make sure to check that it is centered in a test image!!!
   (d) You may have to take a number of test images to center. This is perfectly
       fine.
A.3 Closing the Britton

1. Take a few more biases. If you’ve taken biases through the night, you can omit this step.

2. Turn off the tracking.
   (a) In warm room, Track switch goes off.

3. Put the caps on all telescopes.

4. Move to the zenith using the DFM control program (TCS). Go to the Telescope tab and then into the Movement menu. Click on Offset and look for the Zenith tab. A box called Set Zenith Position should appear. The box instructs you to disable the tracking, which is done with the tracking switch on the Motor Driver Chassis. Once the tracking rate on the TCS displays 0.0 you click on the Apply button in the Set Zenith Position box. Start Slew to actually move.

5. Rotate dome to home position and close it.
   (a) Unplug dome rotator motor.
   (b) Plug in the curly cord.
   (c) Close bottom lip first, and then the top.
   (d) Turn off the CCD.
   (e) BE SURE TO UNPLUG ALL CORDS FOR THE DOME.

6. Close all doors and make sure they are locked. Don’t forget about the ones under the telescope!

7. Turn on HALT MOTORS button.

8. In The Sky window, under Telescope Menu, select Link, then Terminate

9. Close The Sky program
   (a) NEVER SAVE CHANGES!!!

10. Close the DFM TCS.

11. Turn off drivers.
    (a) Drives first, then MTR Driver Chassis

12. Turn off the data taking computer
13. Unplug ranger (beige box)

14. Turn off the control computer.

15. You’re done!! GO HOME!
B  Instructions for Data Reduction

Data reduction is the process by which we account for noise in the scientific images. We do this by taking calibration images.

This section provides the process by which we reduced data from NURO and from the Michael L. Britton Observatory. The general steps are given in Section B.1. This process has been slightly adapted from Recine 2011[23]. There are slightly adjusted instructions for the I filter. This information is in Section B.2.

B.1  General Data Reduction Process

1. Check Biases

   -Start IRAF and ds9. Then go into the directory that your data is in. At this point, it can be helpful to have a unix window open and also in your directory.
   -Obtain the logs for the night of data you are looking at.
   -Go through the log and check each bias (or zero) image in ds9. To display the image, in the IRAF window type:

     display image.fits

     The image will be displayed by default in frame 1 by hitting enter a second time. To display in another frame, type a different number and hit enter.
     When checking, the biases should not have white or dark streaks through them.
   -Make a list of usable biases called zeros.lis using vi.
   -Combine the zeros into a master zero using zerocombine (noao >imred >cc-dred). Note in the parameters that those given are for the NURO CCD. For the Britton, gain is 1.4 electrons/ADU and read noise is 5.9 electrons. The following parameters should be used:

     PACKAGE = ccdred
     TASK = zerocombine
     input = @zeros.lis List of zero level images to combine
     (output = Zero) Output zero level name
     (combine= average) Type of combine operation
     (reject = avsigclip) Type of rejection
     (ccdtype= ) CCD image type to combine
     (process= no) Process images before combining?
     (delete = no) Delete input images after combining?
(clobber= no) Clobber existing output image?
(scale = none) Image scaling
(statsec= ) Image section for computing statistics
(nlow = 0) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(Isigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise= 6.2) ccdclip: CCD readout noise (electrons)
(gain = 2.15) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 0.) Value if there are no pixels
(mode = ql)

2. Check Darks
   -As above, display each dark image in ds9.
   -Make a list of usable darks called darks.lis.
   -Use darkcombine (noao >imred >ccdred) to create a master dark for the night.
   The parameters are as follows:

   PACKAGE = ccdred
   TASK = darkcombine
   input = @darks.lis List of dark images to combine
   (output = Dark) Output dark image root name
   (combine= average) Type of combine operation
   (reject = avsigclip) Type of rejection
   (ccdtype= ) CCD image type to combine
   (process= no) Process images before combining?
   (delete = no) Delete input images after combining?
   (clobber= no) Clobber existing output image?
   (scale = none) Image scaling
   (statsec= ) Image section for computing statistics
   (nlow = 0) minmax: Number of low pixels to reject
   (nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(Isigma = 3.) Lower sigma clipping factor
(Hsigma = 3.) Upper sigma clipping factor
(rdnoise= 6.2) ccdclip: CCD readout noise (electrons)
(gain = 2.15) ccdclip: CCD gain (electrons/DN)
(noise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 0.) Value if there are no pixels
(mode = qil)

Again, the values for read noise and gain must be changed based on the CCD being used.
- Also note that depending on the telescope, you may or may not have darks.

3. Flats
- Flats are checked using the display in ds9. In addition to streaks, flats are not usable if they have stars in them. Donuts are okay.
- Make a list of usable flats in each filter. The lists should be filterflats.lis. For example, the usable B flats list should be bflats.lis.
- In order to combine the flats, we use flatcombine. First, we edit ccdproc (noao >imred >ccdred) without running it. When you are finished editing ccdproc, type ctrl+d. The parameters are:

PACKAGE = ccdred
TASK = ccdproc
images = List of CCD images to correct
(output = ) List of output CCD images
(ccdtype= ) CCD image type to correct
(max_cac= 0) Maximum image caching memory (in Mbytes)
(noproc = no) List processing steps only?
(fixpix = no) Fix bad CCD lines and columns?
(oversca= no) Apply overscan strip correction?
(trim = no) Trim the image?
(zerocon= yes) Apply zero level correction?
(darkcor= no) Apply dark count correction?
(flatcor= no) Apply flat field correction?
(illumco= no) Apply illumination correction?
(fringec= no) Apply fringe correction?
(readcor= no) Convert zero level image to readout correction?
(scancor= no) Convert flat field image to scan correction?
(readaxi= line) Read out axis (column—line)
(fixfile=ann) File describing the bad lines and columns
(biassec= ) Overscan strip image section
(trimsec= ) Trim data section
(zero=Zero.fits) Zero level calibration image
(dark= ) Dark count calibration image
(flat= ) Flat field images
(illum= ) Illumination correction images
(fringe= ) Fringe correction images
(minrepl=1.) Minimum flat field value
(scantyp= shortscan) Scan type (shortscan—longscan)
(nscan = 1) Number of short scan lines
(interac= no) Fit overscan interactively?
(function= legendre) Fitting function
(order = 1) Number of polynomial terms or spline pieces
(sample = *) Sample points to fit
(naverag= 1) Number of sample points to combine
(niterat= 1) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_rej= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(mode = ql)

-Run flatcombine (noao >imred >ccdred), which will call ccdproc. The parameters are:

PACKAGE = ccdred
TASK = flatcombine
input = @bflats.lis List of flat field images to combine
(output = bFlat) Output flat field root name
(combine= average) Type of combine operation
(reject= avsigclip) Type of rejection
(ccdtype= ) CCD image type to combine
(process= yes) Process images before combining?
(subsets= yes) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber= no) Clobber existing output image?
(scale = mode) Image scaling
(statsec= ) Image section for computing statistics
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rnoise= 6.2) ccdclip: CCD readout noise (electrons)
(gain = 2.15) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 1.) Value if there are no pixels
(mode = ql)

-The master flats must be normalized. To do this, we divide by the ccdmean using imarith (images >imutil). The mean will be operand2 in the IRAF parameters. The parameters are as follows:

PACKAGE = imutil
TASK = imarith
operand1= bFlat.fits Operand image or numerical constant
op = / Operator
operand2= 16206.14 Operand image or numerical constant
result = bFlat
textunderscore norm.fits Resultant image
(title = ) Title for resultant image
(divzero= 0.) Replacement value for division by zero
(hparams= ) List of header parameters
(pixtype= ) Pixel type for resultant image
(calcyp= ) Calculation data type
(verbose= yes) Print operations?
(noact = no) Print operations without performing them?
(mode = ql)

4. Object Images

-Check each of the object images using ds9. The object images should not have dark or light streaks. Additionally, the images need to be focused, so any images in which the stars look like donuts or eggs should not be used.

-Usable images should be put into lists by filter called filterobjects.lis (for instance objects.lis).

-Subtract biases from the object images. This should be done for each filter. Use the following parameters in ccdproc:

```plaintext
PACKAGE = ccdred
TASK = ccdproc
images = @bobjects.lis List of CCD images to correct
(output = ) List of output CCD images
(ccdtype= ) CCD image type to correct
(max_cac= 0) Maximum image caching memory (in Mbytes)
(noproc = no) List processing steps only?
(fixpix = no) Fix bad CCD lines and columns?
(oversca= no) Apply overscan strip correction?
(trim = no) Trim the image?
(zerocor= yes) Apply zero level correction?
(darkcor= no) Apply dark count correction?
(flatcor= no) Apply flat field correction?
(illumco= no) Apply illumination correction?
(fringec= no) Apply fringe correction?
(readcor= no) Convert zero level image to readout correction?
(scancor= no) Convert flat field image to scan correction?
(readaxi= line) Read out axis (column—line)
(fixfile= ) File describing the bad lines and columns
(biassec= ) Overscan strip image section
(trimsec= ) Trim data section
(zero = Zero.fits) Zero level calibration image
(dark = ) Dark count calibration image
(flat = ) Flat field images
```
(illum = ) Illumination correction images
(fringe = ) Fringe correction images
(minrepl= 1.) Minimum flat field value
(scantyp= shortscan) Scan type (shortscan—longscan)
(nscan = 1) Number of short scan lines
(interac= no) Fit overscan interactively?
(functio= legendre) Fitting function
(order = 1) Number of polynomial terms or spline pieces
(sample = *) Sample points to fit
(naverag= 1) Number of sample points to combine
(niterat= 1) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_re= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(mode = ql)

-Divide the object images by the normalized flat field image. Copy filterobjects.lis to filterobjects.out.lis. Use vi to put ff. in front of each file name to indicate flat fielding of the image. Use the following parameters in imarith:

PACKAGE = imutil
TASK = imarith
operand1= @bobjects.lis Operand image or numerical constant
op = / Operator
operand2= bFlat norm.fits Operand image or numerical constant
result = @bobjects.out.lis Resultant image
(title = ) Title for resultant image
(divzero= 0.) Replacement value for division by zero
(hparams= ) List of header parameters
(pixtype= ) Pixel type for resultant image
(calctyp= ) Calculation data type
(Verbose= yes) Print operations?
(noact = no) Print operations without performing them?
(mode = ql)
B.2 I Filter Dithering

There is a slight modification to flat fielding for I images from NURO. When making the list of I images, make iobjects.i.out.lis. Tack on ff.i. to the beginning of all the file names. This will allow the final image names (in all filters) to have the same format of: ff.filename.fits. This will also make using the IDL programs easier, because nothing will need to be changed.

Steps to remove I filter fringing after flat fielding

1. Make a list of all images taken in the I filter during the night that were dithered. Call this list i_all.lis

2. Use imcombine to put the images together. The parameters are given here:

   PACKAGE = immatch
   TASK = imcombine
   input = @i_all.lis List of images to combine
   output = i_fringetest List of output images
   (headers= ) List of header files (optional)
   (bpmasks= ) List of bad pixel masks (optional)
   (rejmask= ) List of rejection masks (optional)
   (nrejmas= ) List of number rejected masks (optional)
   (expmask= ) List of exposure masks (optional)
   (sigmas = ) List of sigma images (optional)
   (imcmb = $I) Keyword for IMCMB keywords
   (logfile= STDOUT) Log file
   (combine= median) Type of combine operation
   (reject = none) Type of rejection
   (project= no) Project highest dimension of input images?
   (outtype= real) Output image pixel datatype
   (outlimi= ) Output limits (x1 x2 y1 y2 ...)
   (offsets= none) Input image offsets
   (masktyp= none) Mask type
   (maskval= 0) Mask value
   (blank = 0.) Value if there are no pixels
   (scale = none) Image scaling
   (zero = none) Image zero point offset
   (weight = none) Image weights
   (statsec= ) Image section for computing statistics
(expname= ) Image header exposure time keyword
(lthresh= INDEF) Lower threshold
(hthresh= INDEF) Upper threshold
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise= 6.2) ccdclip: CCD readout noise (electrons)
(gain = 2.15) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(sigscal= 0.1) Tolerance for sigma clipping scaling corrections
(pclip = -0.5) pclip: Percentile clipping parameter
(grow = 0.) Radius (pixels) for neighbor rejection
(mode = ql)

3. Now, we normalize the combined I fringe image, i_fringetest. We do this in IRAF. In IRAF command line, type

    imstat i_fringetest

    One of the things IRAF will tell you is the mean of the image. Write this down.

4. Use imarith to normalize i_fringetest. In the parameters, operand2 should be the mean of i_fringetest.

    PACKAGE = imutil
    TASK = imarith
    operand1= i_fringetest Operand image or numerical constant
    op = / Operator
    operand2= 2018. Operand image or numerical constant
    result = i_fringe Norm Resultant image
    (title = ) Title for resultant image
    (divzero= 0.) Replacement value for division by zero
    (hparams= ) List of header parameters
    (pixtype= ) Pixel type for resultant image
    (calctyp= ) Calculation data type
(verbose= yes) Print operations?
(noact = no) Print operations without performing them?
(mode = ql)

5. Copy iobjects.i.out.lis to the file iobjects.out.lis. In the new file, the names should say ff.*.fits. This way, the images that have been fringe-removed have the same name format as the images in other filters that have been reduced. This also differentiates them from the images that have not had the fringes removed (which have ff.i.filename.fits).

![Figure 14: Final pattern due to fringing in the I filter at NURO.](image.png)

6. Use imarith to divide by the normalized I fringe.

    PACKAGE = imutil
    TASK = imarith
operand1= @iobjects.i.out.lis Operand image or numerical constant

op = / Operator

operand2= i fringe Norm Operand image or numerical constant

result = @iobjects.out.lis Resultant image

(title = ) Title for resultant image

(divzero= 0.) Replacement value for division by zero

(hparams= ) List of header parameters

(pixtype= ) Pixel type for resultant image

(calc_typ= ) Calculation data type

(verbose= yes) Print operations?

(noact = no) Print operations without performing them?

(mode = ql)

7. Check the combined I fringe image by displaying it in DS9. Fig. 14 shows an example of the fringe pattern with no stars in it. If there are still stars visible in the image, check each set of dithers to determine if some sets can be removed from the combined image. If so, do not use sets of dithers that have stars when combined. Continue this checking process until the combined image has no stars in it.
C Instructions for Photometry

The photometry process measures the instrumental magnitude of the objects in question. In this section, we describe how to perform aperture photometry on science images and the adjustments that must be made to photometer standard fields. Figure 15 shows the photometry process for science fields. Green boxes indicate files, red boxes indicate programs, orange boxes indicate IRAF routines, and the blue box indicates a UNIX command.

C.1 Photometry of Scientific Field

Figure 15: Flowchart showing photometry process.
1. Find the FWHM of a comparison star.
   -Use IRAF’s imexamine (images >tv) procedure to track one star. Type
     imexamine @filterobjects.out.lis >filter_fwhm_list
   -Center the blinking annulus on the comparison star displayed in ds9. Type ’a’
     to select that pixel. Type ’n’ to move to the next image. When all images have
     been cycled through, type ’q’ to quit the session.
   -Use vi to delete the top lines (starting with #) of filter fwhm_list
   -If you are working with NURO data, run mergetest.pro in IDL. The program
     will prompt you for the filter you are using. It takes the object list and the list
     of FWHM’s and outputs season_test_filter. In this file, the fifth column contains
     the FWHM of the comparison star in each image.
   -If you are working with Britton data, run merge_one_DSON.pro in IDL. The
     program will prompt you for the filter you are using. It takes the object list and
     the list of FWHM’s and outputs season_info_filter. In this file, the fifth column
     contains the FWHM of the comparison star in each image.

2. Find the average position, FWHM, and image closest to the average position.
   -Run centermost.pro in IDL. The program will prompt you for the filter you are
     using. It will print to the screen the mean FWHM of the comparison star, the
     mean x position of the comparison star, the mean y position of the comparison
     star, and the centermost image. These should all be recorded.

3. Determine the coordinates of the comparison stars in the centermost image.
   -Use imexamine to look at the centermost image. Type
     imexamine centerimage.fits >center.coords
   -Center the blinking annulus on each comparison star and the object. Type ’a’
     over each object. You should do this in the same order for each night and filter.
     Additionally, the object being studied should be selected last. Type ’q’ when
     finished.
   -View center.coords in vi and delete the top three lines of text, the indented
     lines, and the repeated numbers.
   -Copy season_test_filter to tracking_filter.coords. In the new file, delete every-
     thing except the last two columns.

4. Determine the coordinates of all comparison stars and object in all images.
- Run offsets.pro in IDL. It will prompt you for the filter being used, and the x and y positions in the centermost image (which you can find in season_test_filter). There are also two pop-up windows. In the first pop-up, click next to get to the third window. Make the following changes in the first pop-up:

FIELD1=x
type=double precision
FIELD2=y
type=double precision

In the second pop-up, click next to the the third window. Make the following changes:

FIELD1=image
type=string

The program offsets.pro outputs offsets_filter.dat, which has the position offset of each image from the centermost image.

-Move center.coords to mastercoords_filter.dat.
- Run imcoords.pro in IDL. It will prompt you for the filter being used. It will also give you a pop-up window. Click next to the second window. Make the following change:

#offields=3

Click next again to the third window. Make the following changes:

FIELD1=x
type=double precision
FIELD2=y
type=double precision
FIELD3=id
type=long integer
The program imcoords.pro will output a file (for each file in the object list) that ends in *.coo.0.

- Check that the coordinates were properly calculated by using tvmark (in IRAF). Type:

```
tvmark 1 centerimage.fits.coo.0
```

You will see small colored dots on the ds9 screen. These should fall in the middle of each of your comparison stars and object.

5. Perform aperture photometry.

- Copy filterobjects.out.lis to filterobjects.mag.lis. In the new file, add .mag.filter to the end of every image name. To do this quickly in vi, type the command:

```
:1,$s/.fits/.fits.mag.filter/g
```

While doing this, you should fill in the appropriate filter.

- Run the phot (noao > digiphot > daophot) procedure in IRAF. To do this, you must edit some other procedures first. Edit photpars to make the aperture value 10. In fitskypars, make the annulus value 1.5 x average FWHM that you wrote down earlier. Make the output of the phot procedure filterobjects.mag.lis. Once these have been edited, run phot.

6. Sort out the instrumental magnitudes

- Use txdump in IRAF to move magnitudes. Epar txdump so the parameters are:

```
PACKAGE = daophot
TASK = txdump
textfile= *.mag.r Input apphot/daophot text database(s)
fields = id,image,mag,merr,cerror,serror,perror Fields to be extracted
expr = yes Boolean expression for record selection
(headers= no) Print the field headers ?
(paramet= yes) Print the parameters if headers is yes ?
(mode = ql) Mode of task
```
-Then type:

```
txdump *.mag.filter >mags1_filter.out
```

If txdump has been properly set up, continue typing enter until txdump runs.

-Run merge_two_NURO.pro (merge_two_DSON.pro for Britton data) which will take in season_test_filter (season_info_filter for Britton data) and mags1_filter.out and will output mags2_filter.out.

-Sort the contents of the mags2_filter.out file to the file mags3_filter.out in the UNIX window. Contents will be sorted based on the star number. If you are using the Snow Leopard OsX, type:

```
sort mags2_filter.out -nk 4 >mags3_filter.out
```

Otherwise, you will need to look up the sort command.

-Run binstar_B.pro. It will take in mags3_filter.out, and output files called star#### filter.dat. There will be as many files as you have comparison stars and objects in the frame. These files contain columns from left to right containing the Julian date (JD), apparent magnitude, error in apparent magnitude (merror), and the star number.
C.2 Standard Field Photometry

The process for standard field photometry was adapted from Massey and Davis (1992), and from K. Recine’s senior thesis. The way in which this process calibrates the magnitude to a standard scale is discussed in Sec. 2.7.

The process we use for photometry of standard fields is similar to that found in Section C.1. However, we change naming conventions slightly in order to differentiate standard fields from each other and from science fields. The naming conventions changes are given in Table 8. Additionally, versions of the IDL programs used in Section C.1 have been written to accommodate specific standard star fields, and to use the new naming conventions. These programs have the same name as those used for science images, but end in ‘.obj’, where obj corresponds to the letters labeling the fields. For instance, in order to run the program centermost.pro on the standard field RU 149, the program centermost_ru.pro should be used.

<table>
<thead>
<tr>
<th>NURO Science Image</th>
<th>NURO Standard Image</th>
<th>Example Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>filterobjects.lis</td>
<td>filter_object.lis</td>
<td>b_pg0231.lis</td>
</tr>
<tr>
<td>filter_fwhm_list</td>
<td>filter_obj_fwhm_list</td>
<td>b_pg_fwhm_list</td>
</tr>
<tr>
<td>season_test_filter</td>
<td>obj_test_filter</td>
<td>pg_test_b</td>
</tr>
<tr>
<td>tracking_filter.coords</td>
<td>tracking_obj_filter.coords</td>
<td>tracking_pg_b.coords</td>
</tr>
<tr>
<td>offsets_filter.dat</td>
<td>offsets_obj_filter.dat</td>
<td>offsets_pg_b.dat</td>
</tr>
<tr>
<td>mastercoords_filter.dat</td>
<td>mastercoords_obj_filter.dat</td>
<td>mastercoords_pg_b.dat</td>
</tr>
<tr>
<td>filterobjects.mag.lis</td>
<td>filter_object.mag.lis</td>
<td>b_pg0231.mag.lis</td>
</tr>
<tr>
<td>* .mag.filter</td>
<td>* .mag.filter.obj</td>
<td>* .mag.b.pg</td>
</tr>
<tr>
<td>mags1_filter.out</td>
<td>mags1_filter_obj.out</td>
<td>mags1_b_pg.out</td>
</tr>
<tr>
<td>mags2_filter.out</td>
<td>mags2_filter_obj.out</td>
<td>mags2_b_pg.out</td>
</tr>
<tr>
<td>mags3_filter.out</td>
<td>mags3_filter_obj.out</td>
<td>mags3_b_pg.out</td>
</tr>
<tr>
<td>star#####_filter.dat</td>
<td>star#####_Filter_obj.dat</td>
<td>star0001_B_pg.dat</td>
</tr>
</tbody>
</table>

Table 8: Changes to file naming conventions when photometering standard fields

Once all of the fields and filters have been photometered, we begin the process of calculating the coefficients. Calculating coefficients will be done using the IRAF package photcal (noao >digiphot >photcal). We need to collect various pieces of data into one place in order for the coefficients to be properly calculated.

1. Correct the airmass. The airmass listed in the header is the airmass at the beginning of the observation. The more correct airmass is that at the middle of the observation. IRAF must know where the observation was made. We update
the header using hedit with the following parameters:

```plaintext
PACKAGE = imutil
TASK = hedit
images = @i.ru149.lis images to be edited
fields = OBSERVAT fields to be edited
value = lowell value expression
(add = no) add rather than edit fields
(addonly= no) add only if field does not exist
(delete = no) delete rather than edit fields
(verify = yes) verify each edit operation
(show = yes) print record of each edit operation
(update = yes) enable updating of the image header
(mode = qI)
```

Once the place of observation is known, we can correct the airmass. This is done using setairmass in IRAF with the following parameters:

```plaintext
PACKAGE = astutil
TASK = setairmass
images = @b.pg0231.lis Input images
(observa= ) _observatory) Observatory for images
(intype = beginning) Input keyword time stamp
(outtype= effective) Output airmass time stamp
(ra = TELRA) Right ascension keyword (hours)
(dec = TELDEC) Declination keyword (degrees)
(equinox= EQUINOX) Equinox keyword (years)
(st = LST-OBS) Local siderial time keyword (hours)
(ut = UTCSTART) Universal time keyword (hours)
(date = date-obs) Observation date keyword
(exposur=exptime)Exposure time keyword(seconds)
(airmass= airmass) Airmass keyword (output)
(utmiddl= utmiddle) Mid-observation UT keyword (output)
(scale = 750.) The atmospheric scale height
(show = yes) Print the airmasses and mid-UT?
(update = yes) Update the image header?
(overrid= yes) Override previous assignments?
```
(mode = ql)

2. Make files containing airmasses. This can be done by:
   `photcal>hselect @filter_object.lis AIRMASS yes >filter_object.AM`
   This pipes the airmass of each image in the list into the .AM file.

3. Create the standstar file. This will look like Fig. 16 (Fig. 17 in Massey and Davis 1992). It contains matched sets of observations (i.e., one set of BVR) of the standard stars. Each star in each field should be labeled for each matched set.

4. Create the standobs file. This file will resemble Fig. 17 (Fig. 20 in Massey and Davis 1992). It has columns from left to right containing each star by field, filter, airmass, x position, y position, magnitude, and magnitude error. The naming conventions for the stars must be the same as those used in the onlandolt catalog, which will be discussed later. It is important that the asterisks are present in the first column for repeated observations of the same star. For the photometric night used in this case, the IDL program fig20.pro was used to collect all of the data into the file. Note that fig20.pro is specifically written for the photometric night used here, and must be modified to be used for other nights.

5. Create the fstandobs.dat file. This file will look like Fig. 18 (Fig. 21 in Massey and Davis 1992). It has the format for the standobs file.

6. Run mkconfig. This will make the configuration file testnuro.cfg. The parameters to be used are:

   ```
   PACKAGE = photcal
   TASK=mkconfig
   config = testnuro.cfg The new configuration file
   catalog = onlandolt The source of the catalog format specification
   observat= standobs The source of the observations file format speci
   transfor= onlandolt The source of the transformation equations
   templat= ) An existing template configuration file
   (catdir = ) _catdir) The standard star catalog directory
   (verify = no) Verify each new entry
   (edit = yes) Edit the new configuration file
   ```
<table>
<thead>
<tr>
<th>PG0231_A</th>
<th>ff.2010121.052.fits ff.2010121.053.fits ff.2010121.054.fits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG0231_A</td>
<td>ff.2010121.056.fits ff.2010121.057.fits ff.2010121.058.fits</td>
</tr>
<tr>
<td>PG0231_A</td>
<td>ff.2010121.111.fits</td>
</tr>
<tr>
<td>PG0231_B</td>
<td>ff.2010121.113.fits ff.2010121.114.fits ff.2010121.115.fits</td>
</tr>
<tr>
<td>PG0231_B</td>
<td>ff.2010121.052.fits ff.2010121.053.fits ff.2010121.054.fits</td>
</tr>
<tr>
<td>PG0231_B</td>
<td>ff.2010121.056.fits ff.2010121.057.fits ff.2010121.058.fits</td>
</tr>
<tr>
<td>PG0231_B</td>
<td>ff.2010121.111.fits</td>
</tr>
<tr>
<td>PG0231_C</td>
<td>ff.2010121.113.fits ff.2010121.114.fits ff.2010121.115.fits</td>
</tr>
<tr>
<td>PG0231_C</td>
<td>ff.2010121.052.fits ff.2010121.053.fits ff.2010121.054.fits</td>
</tr>
<tr>
<td>PG0231_C</td>
<td>ff.2010121.056.fits ff.2010121.057.fits ff.2010121.058.fits</td>
</tr>
<tr>
<td>PG0231_C</td>
<td>ff.2010121.111.fits</td>
</tr>
<tr>
<td>PG0231_D</td>
<td>ff.2010121.113.fits ff.2010121.114.fits ff.2010121.115.fits</td>
</tr>
<tr>
<td>PG0231_D</td>
<td>ff.2010121.052.fits ff.2010121.053.fits ff.2010121.054.fits</td>
</tr>
<tr>
<td>PG0231_D</td>
<td>ff.2010121.056.fits ff.2010121.057.fits ff.2010121.058.fits</td>
</tr>
<tr>
<td>PG0231_D</td>
<td>ff.2010121.111.fits</td>
</tr>
<tr>
<td>PG0231_E</td>
<td>ff.2010121.113.fits ff.2010121.114.fits ff.2010121.115.fits</td>
</tr>
<tr>
<td>PG0231_E</td>
<td>ff.2010121.052.fits ff.2010121.053.fits ff.2010121.054.fits</td>
</tr>
<tr>
<td>PG0231_E</td>
<td>ff.2010121.056.fits ff.2010121.057.fits ff.2010121.058.fits</td>
</tr>
<tr>
<td>PG0231_E</td>
<td>ff.2010121.111.fits</td>
</tr>
<tr>
<td>RUI49</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_A</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_A</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_A</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_A</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_B</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_B</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_B</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_B</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_C</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_C</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_C</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_C</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_D</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_D</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_D</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_D</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_E</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_E</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_E</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_E</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_F</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_F</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
<tr>
<td>RUI49_F</td>
<td>ff.2010121.157.fits ff.2010121.158.fits ff.2010121.159.fits</td>
</tr>
<tr>
<td>RUI49_F</td>
<td>ff.2010121.161.fits ff.2010121.163.fits</td>
</tr>
<tr>
<td>RUI49_G</td>
<td>ff.2010121.133.fits ff.2010121.134.fits ff.2010121.135.fits</td>
</tr>
<tr>
<td>RUI49_G</td>
<td>ff.2010121.137.fits ff.2010121.138.fits ff.2010121.139.fits</td>
</tr>
</tbody>
</table>

Figure 16: Example of format for the standstar file.
Table 1: Example of format for the standobs file.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X₀</td>
<td>x-coordinate in filter B</td>
</tr>
<tr>
<td>y₀</td>
<td>y-coordinate in filter B</td>
</tr>
<tr>
<td>m₀</td>
<td>Instrumental magnitude in filter B</td>
</tr>
<tr>
<td>errorₐ</td>
<td>Magnitude error in filter B</td>
</tr>
</tbody>
</table>

Table 2: Example of format for the fstandobs.dat.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xᵥ</td>
<td>x-coordinate in filter V</td>
</tr>
<tr>
<td>yᵥ</td>
<td>y-coordinate in filter V</td>
</tr>
<tr>
<td>mᵥ</td>
<td>Instrumental magnitude in filter V</td>
</tr>
<tr>
<td>errorᵥ</td>
<td>Magnitude error in filter V</td>
</tr>
</tbody>
</table>

Figure 17: Example of format for the standobs file.

Figure 18: Example of format for the fstandobs.dat.
When mkconfig runs, it will open the configuration file in vi for editing. The bottom section contains the transformation equations to be solved. They should be of the following form:

\[
    mB = (BV + V) + b1 + b2 \times XB + b3 \times BV \\
    mV = V + v1 + v2 \times XV + v3 \times BV \\
    mR = (V - VR) + r1 + r2 \times XR + r3 \times VR
\]

(13)  

(14)  

(15)  

Save the file to exit the editor.

7. Run fitparams. This will perform the actual fitting. The parameters to be used are:

\[
\begin{align*}
    \text{PACKAGE} &= \text{photcal} \\
    \text{TASK} &= \text{fitparams} \\
    \text{observat}= &\text{standobs List of observations files} \\
    \text{catalogs}= &\text{onlandolt List of standard catalog files} \\
    \text{config} = &\text{testnuro.cfg Configuration file} \\
    \text{paramete}= &\text{testnuro.ans Output parameters file} \\
    \text{weightti}= &\text{photometric Weighting type (uniform,photometric,equations)} \\
    \text{addscat}= &\text{yes} \text{ Add a scatter term to the weights ?} \\
    \text{toleran}= &\text{3.0000000000000E-5} \text{ Fit convergence tolerance} \\
    \text{maxiter}= &\text{15} \text{ Maximum number of fit iterations} \\
    \text{nreject}= &\text{2} \text{ Number of rejection iterations} \\
    \text{low_rej}= &\text{3.} \text{ Low sigma rejection factor} \\
    \text{high_re}= &\text{3.} \text{ High sigma rejection factor} \\
    \text{grow} = &\text{0.} \text{ Rejection growing radius} \\
    \text{interac}= &\text{yes} \text{ Solve fit interactively ?} \\
    \text{logfile}= &\text{testlog} \text{ Output log file} \\
    \text{log_unm}= &\text{yes} \text{ Log any unmatched stars ?} \\
    \text{log_fit}= &\text{yes} \text{ Log the fit parameters and statistics ?} \\
    \text{log_res}= &\text{yes} \text{ Log the results ?} \\
    \text{catdir} = &\text{.)_catdir} \text{ The standard star catalog directory}
\end{align*}
\]
Output graphics device
Graphics cursor input
(mode = ql)

Figure 19: Window to interactively fit transformation equations in IRAF.

When fitparams runs, it will open a window that looks like Fig. 19. This window is used to interactively fit the coefficients for the transformation equation. Typing ‘d’ will delete the nearest data point from the fit. Typing ‘f’ will re-fit the coefficients with the changes to the data. Typing ‘u’ will undelete the nearest data point from the fit. The goal is to have a converging solution with low RMS, both of which are shown at the top of the screen. Massey and Davis, 1992, has detailed instructions for running the interactive fit.

8. Obtain the transformation coefficients. They are currently located in the file testnuro.ans under the appropriate date and filter. IRAF keeps track of all of the fits that have been done, so be sure to check the date.

9. Use stand_fits.pro to calculate the calibrated magnitude of the comparison stars in the science field. If another standard night is used, the coefficients will need to be changed inside of the program.
D Instructions for Data Processing and Making Light Curves

Data processing takes the instrumental magnitudes we measured and converts them to a standard scale. These instructions are based on the assumption that the real magnitudes of the comparison stars are known. Figure 20 shows a flowchart outlining this process. Green boxes indicate files, red boxes indicate programs, and the purple box prints to the screen.

Figure 20: Flowchart showing the data processing method.

1. Run lcds_loop.pro. This takes in star#####_filter.dat, snum.lis (that was created by binstar_B.pro), and snum1.lis (that you must create by copying snum.lis and removing the first two zeros from each number) It outputs a file called date_filter.sav. The date_filter.sav file contains the columns shown in Table 9. Because the number of comparison stars can change based on field, we force the object data to be in the last two columns.

2. Run diffmag_B.pro. It takes in date_filter.sav that has just been created. This program creates the synthetic comparison star by computing the average flux for
Table 9: Columns contained in date filter.sav file. We can have varying numbers of comparison stars, so we make the last two columns contain data from the object of interest.

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column n-1</th>
<th>Column n</th>
</tr>
</thead>
<tbody>
<tr>
<td>JD</td>
<td>Mag comp1</td>
<td>Mirror comp1</td>
<td>Mag comp2</td>
<td>Mirror comp2</td>
<td>Obj mag</td>
<td>Obj merror</td>
</tr>
</tbody>
</table>

all of the comparison stars, and then computing the average magnitude from the average flux. It then calculates the differential magnitude of each comparison star and object as the measured magnitude minus the magnitude of the synthetic star. It outputs the differential magnitudes into the file cd_mags_filter.sav. This file contains columns from left to right of the synthetic magnitude, the differential magnitude of the object, and the differential magnitudes of the comparison stars.

The program diffmag_B.pro is currently set up to calculate this for ten comparison stars. However, the program can be easily edited for more or fewer stars. This can be done by changing the number of fluxes being calculated, changing the calculation of average comparison star flux, changing the differential magnitudes, and changing the values being written.

3. Run test lc.pro. It takes in date filter.sav and cd_mags_filter.sav. This program prints two test light curves to the screen. It will let you see quickly if a comparison star is varying when it should not be. If one of the comparison stars is varying, it must be excluded. However, it will probably be a good idea to follow this star as well in order to determine whether it is varying periodically. If you notice that a comparison star is varying while the object of interest is not, check to make sure the data has been correctly written.

4. Run obj_standards_filter.pro. These are a set of programs specifically for a given object and filter. The real magnitudes of the comparison stars, determined by stand_fit.pro, are hard coded into the programs. They will take in date_filter.sav and cd_mags_filter.sav. The standard magnitude of the synthetic comparison star is added to the differential magnitude of the object.

To save the data, IDL prints a prompt to explain what to do. It then enters a stop so that the instructions can be followed. The data must be saved with a name to distinguish it from other nights. The form of the file name will be obj_Filter_Date_real.sav.

5. After all nights have been put onto a standard scale, they should be collected in one directory. We create an array for each filter with three columns, phase, JD,
and standard magnitude. The JD and standard magnitudes will come directly from the obj_Filter_Date_real.sav file made in the previous step for each night. We determine the phase as

\[ \phi = \frac{(JD - JD_0)}{P} \mod P \]  

(16)

where \( JD \) is the julian date of observation, \( JD_0 \) is a specific julian date\(^9\), and \( P \) is the period. This calculation can easily be done with IDL. Make sure to save the array for each filter as a file.

6. To actually create the light curves, plot phase on the x-axis and standard magnitude on the y-axis.

\(^9\) Typically the specific julian date will be either a minimum or maximum, and will come from an ephemeris. In principle, though, any julian date can be used.
E  IDL Programs

This section presents a hard copy of all the IDL routines used in data reduction and processing. The routines have either been updated from previous versions provided by Professor Catrina Hamilton-Drager, or have been written in the past year. Routines were updated in order to avoid making hard changes to code each time a parameter, such as the filter being used, changed. Routines were written in order to avoid needlessly typing into the command line directions that could be easily looped. In both cases, by making mostly minor changes to the IDL code, we have decreased the amount of time needed to go through data processing. Table 10 lists programs edited and written throughout this project.

<table>
<thead>
<tr>
<th>Routines Edited</th>
<th>Routines Written</th>
</tr>
</thead>
<tbody>
<tr>
<td>mergetest.pro</td>
<td>centermost.pro</td>
</tr>
<tr>
<td>offsets.pro</td>
<td>centermost_DSON.pro</td>
</tr>
<tr>
<td>imcoords.pro</td>
<td>lcds_loop.pro</td>
</tr>
<tr>
<td>mergetwo_NURO.pro</td>
<td>test_</td>
</tr>
<tr>
<td>lc.pro</td>
<td>fig20.pro</td>
</tr>
<tr>
<td>binstar B.pro</td>
<td>_stand fit.pro</td>
</tr>
<tr>
<td>diffmag B.pro</td>
<td>GMOri standards Filter.pro</td>
</tr>
</tbody>
</table>

Table 10: IDL routines edited and written during this project.

E.1  centermost

pro centermost

; The purpose of this program is to determine the image in a given
; filter for a given night which has the first comparison star in the
; centered relative to all the images in that filter during that
; night. This image will be used for tracking.

; The program asks for the filter being used.

; Outputs to the command line are the mean x and y positions of the
; star and the centermost image.

; If the program produces an error message because two images both
; count as centermost, there is a commented out workaround at the end
;of the program.

;Determine the filter in use.
filter='string'
print,'What filter are you using? Please use a lowercase letter.'
read,filter

;File containing data
c=season_test_{filter}

;Read in the file and determine the mean full width at half max
readcol,c=file,image,date,time,jd,fwhm,format='(a,a,a,d12.4,d4.2)'
meanfwhm=mean(fwhm)
print,'Mean FWHM=',meanfwhm

;Determine the mean x and y positions of the tracking star
readcol,c=file,x,y,format='(a,x,x,x,x,d8.3,d8.3)'
meanx=mean(x)
meany=mean(y)
print,'Mean x position=',meanx
print,'Mean y position=',meany

;Find the distance from the tracking star to the mean position of the
;tracking star
xx=abs(x-meanx)
yy=abs(y-meany)
minx=min(xx)
miny=min(yy)

;The centermost image is closest to the mean position
checkimage:good=where(xx le minx and yy le miny)

;This is the image that is closest to the center
if good NE -1 then print,im(good)

;None of the images was close enough to the center. Check a bit
;further from the center in each direction.
if good EQ -1 then begin
minx=.1+minx
miny=.1+miny
goto,checkimage
dendif

;Work around if two images are defined as "good"
dummy=good(0)
dummy=fix(dummy)

;if dummy NE -1 then print,im(dummy)

;if dummy EQ -1 then begin
;minx=.1+minx
;miny=.1+miny
;goto,checkimage
;endif

end

**E.2 centermost DSON.pro**

pro centermost DSON

;The purpose of this program is to determine the image in a given
;filter for a given night which has the first comparison star in the
;centered relative to all the images in that filter during that
;night. This image will be used for tracking.

;Determine the filter in use.
filter=’string’
print,’What filter are you using? Please use a lowercase letter.’
read,filter

;File containing data
file=’season_info_’+filter
;Read in the file and determine the mean full width at half max
readcol,file,image,date,time,jd,fwhm,format='(a,a,a,d12.4,d4.2)'
meanfwhm=mean(fwhm)
print,’Mean FWHM=’,meanfwhm

;Determine the mean x and y positions of the tracking star
readcol,file,im,x,y,format='(a,x,x,x,x,d8.3,d8.3)'
meanx=mean(x)
meany=mean(y)
print,’Mean x position=’,meanx
print,’Mean y position=’,meany

;Find the distance from the tracking star to the mean position of the
;tracking star
xx=abs(x-meanx)
yy=abs(y-meany)
minx=min(xx)
miny=min(yy)

;The centermost image is closest to the mean position
checkimage:good=where(xx le minx and yy le miny)

;This is the image that is closest to the center
if good NE -1 then print,im(good)

;None of the images was close enough to the center. Check a bit
;further from the center in each direction.
if good EQ -1 then begin
minx=.1+minx
miny=.1+miny
goto,checkimage
endif
end
E.3  mergetest

pro mergetest,fwhm

;This script is setup to merge information for a season of observing for a single star field on the VVO 24” telescope. This is a pretty specific task and the code is not very smart.

;Two input files are expected.
; 1) Bimlist - a list of the file names of all the shifted images for the field. Note that shifted images are assumed since the code will lop off the first three characters of the file name when making the output file at the end. Those characters should be 'Sh.'.
;
; 2) fwhm_list_B - this list can be named anything but the code is hardwired to look for 'fwhm_list’. Change it if you called it something else. The format that the fwhm_list file should be as follows. It is assumed to be created with the following command performed in IRAF on the shifted images:
;
; imexamine @imlist >fwhm_list
;
; where you use the keys R then N to get radial profile information in a selected star (use the same star in each image) output to the fwhm_list file.
;
;The output file created ('season_info' by default) prints the following information seperated by double spaces:
;
; Filename Date(UT) Time(UT) JD FWHM
;
;The date and time are retrieved from the FITS header of the image so you need to be working in the directory where the shifted images are in order for this code to work.
;
Edited program to work with 0.8-m images from McDonald Observatory.

Edited program to work with NURO images. Changed the call for time.

Edited program to read the whole ASCII file at once.

```fortran
filter='string'
print,”What filter are you using? Use a lowercase letter please.”
read,filter

objoutfile=filter+’objects.out.lis’
:Open input and output files
openr,file1,objoutfile,/getJun
:openr,file2,’temp.a’/getJun

fwhmfile=filter+’fwhm_list’
readfmt,fwhmfile,’(A80)’,line2s ;Original has’b fwhm_list’,’(A80)’,line2s
print,line2s
c=0 ;object counter
i=0 :frame counter
nframe=n_elements(line2s)/3

season_testfile=’season_test_’+filter
openw,out,season_testfile,/getJun

;Initialize string variables and define format of output
line1 = ”
line2 = line1
fwhm=fltarr(nframe)
xx=fltarr(nframe)
yy=fltarr(nframe)
format1 = ’(3(a,2x),f13.4,2x,f4.2,2x,f7.2,2x,f7.2)’

;Step through the filenames of the field
```
while (not eof(file1)) do begin
readf,file1,line1
line1 = strcompress(line1,remove_all) ; remove whitespace
starFile = line1
header = headfits(line1) ; read FITS header
date = strtrim(sxpar(header,'DATE-OBS'),2) ; get date info from header
time = strtrim(sxpar(header,'UTCSTART'),2) ; get time info from header
parseDate = str sep(date, '-') ; parse date info
parseTime = str sep(time,':') ; parse time info
jd=julday(parseDate(1),parseDate(2),(parseDate(0)),
         parseTime(0),parseTime(1),parseTime(2)) ; get Julian Date (double prec)

; Read lines from fwhm file, skipping lines that start with
; the letter z. The process of making this file includes an
; extra line for each image with the z range information.

for j=ct,ct+2 do begin ; read three lines at a time
line2=strsplit(line2s(j),',',/extract) ; strsplit gives array of strings
if n_elements(line2) eq 4 then begin
xx(i)=float(line2(0)) ; convert to a number
print,xx(i)
yy(i)=float(line2(1)) ; convert to a number
print,yy(i)
endif else if (n_elements(line2) gt 8) then begin
print,"I AM HERE!"
fwhm(i)=float(line2(8)) ; note that idl counts from 0, so should be 8
print,fwhm(i)
endif
endfor ; j

printf,format=format1,out,starFile,date,time,jd,fwhm(i),xx(i),yy(i)
ct=ct+3 ; increment counter
i=i+1 ; increment object counter
endwhile

; Clean up and go home...
close,file1,out,/all
E.4 offsets

pro offsets
   ; This program is being created on 28-JUL-2004 to calculate the offsets
   ; for each image for the 2003/2004 observing season.

   ; This program is being edited on 10-NOV-2005 to work in the North 300s
   ; directory.

   ; This program is being edited on 13-OCT-2006 to work with the McDonald data.
   ; KAJ

   ; This program is being edited on 10-Jun-2009 to work with the McDonald
   ; data for 150s exposures. AM

   ; This program is being edited on 21-Jul-2009 to work with Tenagra 300s
   ; NGC2362 EAST exposures. AM

   ; First read in the file that contains the x and y positions of the
   ; tracking star in each image. This file was created in IRAF by using
   ; TXDUMP and piping the image name and x and y positions into a file
   ; called 'north.coords'.

   ; For the McDonald data the x and y positions are in east.coords

   filter='string'
   print,"What filter are you using? Please use a lowercase letter."
   read,filter

centerfile='tracking_'+filter+'.coords'
centers=ascii template(centerfile)
save,centers,file = 'centers.sav'
;restore,'centers.sav'
centers_data = read ascii(centerfile,template = centers)

outobjfile=filter+'objects.out.lis'
images = ascii template(outobjfile)
save,images,file = ’images.sav’
;restore,’images.sav’
images_data = read ascii(outobjfile,template = images)

image = images_data.image
x = centers_data.x
y = centers_data.y

print,”What is the x position of the tracking star in the centermost image?”
read,xcent

print,”What is the y position of the tracking star in the centermost image?”
read,ycent

nimages = n elements(image)

offsetfile=’offsets_ ’+filter+’.dat’
openw, unit1, offsetfile, /get lun

for j=0,nimages-1 do begin
    offx = x(j) - xcent
    offy = y(j) - ycent
    printf,unit1,image(j),offx,offy,format=’(a20,2(2x,d8.3))’
endfor

close,unit1
free_lun,unit1
end
E.5  imcoords

pro imcoords
 ; This file is being created on 28-JUL-2004. The purpose of this program
 ; is to read in the offsets that were calculated for each image, and apply
 ; them to the master list of stars that were found in the reference image.
 ; The program then writes these modified coordinates out to a file that will
 ; then be used by PHOT when performing the photometry on these stars. Let’s
 ; hope I can make it work!

 ;Determine the filter to read files
filter='string'
print,"What filter are you using? Lowercase letter please.”
read,filter

 ; First read in the file that has all the x and y coordinates for the stars
 ; in the reference image.

 mastercoordsim='mastercoords_'+filter+'.dat'
stars = ascii template(mastercoordsim)
save,stars, file = 'stars.sav'
;restore,'stars.sav'
stars_data = read ascii(mastercoordsim,template = stars)

 x = stars_data.x ; The first column in mastercoords.dat
 y = stars_data.y ; The second column in mastercoords.dat
 id = stars_data.id ; The last column in mastercoords.dat
n = n_elements(x) ; How many stars are there?

 format = '(3x,d8.3,2x,d8.3,47x,I3)'

 offset_file='offsets_'+filter+'.dat'
readcol,offset_file,image,xoff,yoff,format='(a,d7.3,d7.3)'

 file = n_elements(image)
format2 = '(3x,d8.3,2x,d8.3,47x,I4)'

 for i = 0,file-1 do begin
openw, out, image(i)+’coo.0’,/get_lun ; Open the file for adjusted xs

for j = 0, n-1 do begin
xx = x(j) + xoff(i)
yy = y(j) + yoff(i)
printf, out, xx, yy, id(j), format=format2
endfor

close, out
free_lun, out
endfor
end

E.6 mergetwo NURO

pro merge_two_NURO
 ;This program takes in information from season_test_filter and
 ;mags1_filter.out files and outputs mags2_filter.out

 ;Determine the filter to use
filter = ’string’
print, ”What filter are you using? Please tell me a lowercase letter.”
read, filter

season_test_filter = ’season_test ’ + filter
print, season_test_filter
mags1_filter = ’mags1 ’ + filter + ’.out’
mags2_filter = ’mags2 ’ + filter + ’.out’

opentr, IN1, season_test_filter, /get_lun
opentr, IN2, mags1_filter, /get_lun
openw, OUT, mags2_filter, /get_lun
line1 = ""
line2 = line1
form1 = '(f12.4,2x,f6.3,2x,f5.3,2x,a4,3(2x,a))'

readf,IN1,line1
line1 = strtrim(strcompress(line1),2)
parseLine1 = strsep(line1,' ')
filename1 = parseLine1[0]
jd = parseLine1[3]

while (not eof(in2)) do begin
readf,IN2,line2
line2 = strtrim(strcompress(line2),2)
parseLine2 = strsep(line2,' ')
id = parseLine2[0]
filename2 = parseLine2[1]
mag = parseLine2[2]
merr = parseLine2[3]
cerr = parseLine2[4]
serr = parseLine2[5]
perr = parseLine2[6]

if (filename1 ne filename2) then begin
readf,IN1,line1
line1 = strtrim(strcompress(line1),2)
parseLine1 = strsep(line1,' ')
filename1 = parseLine1[0]
jd = parseLine1[3]
endif

if (mag eq 'INDEF') and (merr eq 'INDEF') then $
printf,format=form1,OUT,jd,'99.999','9.999',id,cerr,serr,perr
else printf,format=form1,OUT,jd,mag,merr,id,cerr,serr,perr
endwhile

close,in1,in2,out,/all
E.7 binstar B

pro binstar_B
;
;PURPOSE: This program takes a txdump output file (stm.out) and sorts it into new star###.dat files, one for each of the component stars. (This is an IDL version of the Fortran program BINSTAR.f)
;
;INPUT: mags3_B.out -the TXDUMP output (with julian dates replacing image names) sorted according to ID# in the format (julian date, mag, magerr, id)
; stars -an integer value = the number of component stars
;
;OUTPUT: snum.lis -a list of 4-digit star numbers that binstar will use to name the output files.
; star###.dat -a listing of output from the IRAF PHOT task for each star with the same column structure as stm.out The #### gives the object number.
;
;Create snum.lis file:
k=1
print,"How many images do you have?"
read,images
print,"How many stars are in your images?"
read,stars
openw,unitc,"snum.lis",/get Jun
w=""
format=’(a)’
while k le stars do begin
w=string(k)
v=getwrd(w,0)
if k lt 10 then printf,unitc,’0’+’0’+’0’+v
if k ge 10 and k lt 100 then printf,unitc,’0’+’0’+v,$
format=format
if k ge 100 and k lt 1000 then printf,unitc,'0'+v,format=format
if k gt 999 then printf,unitc,v,format=format
k=k+1
end
close,unitc
free_lun,unitc

;Determine which filters are being used to read out files later
littlefilter='string'
print,"What filter are you using? Please use a lowercase letter."
read,littlefilter

bigfilter='string'
print,"What filter are you using? Please use a capital letter. And yes, I know this is repetitive."
read,bigfilter

; ;Create star###.dat files:
magfile='mags3_'+littlefilter+'.out'
openr,unita,magfile,/get_lun
openr,unitb,"snum.lis",/get_lun
string=""
num=""
line=""\nm=""
m='1'
k='0'

starstring='star0001 _'+bigfilter+'.dat'

openw,unit,starstring,/get_lun
readf,unitb,num
print,num
while k lt images+1 and not eof(unita) do begin
readf,unita,line
k=getwrd(line,3)
print,k
if k ne m then begin
  close,unit
  free_lun,unit
end
if k ne m then readf,unitb,num
string='star'+num+' '+bigfilter+'.dat'
if k ne m then openw,unit,string./get lun
printf,unit,line
m=k
end
close,unit
free_lun,unit
close,unita
close,unita
free_lun,unita
end

E.8 lcds loop

pro lcds_loop

;The purpose of this program is to quickly move data from the
;star#### Filter.dat files into a large array.
;The array has columns from left to right of
;JD, mag comp star 1, mag error comp star 1, ..., mag comp star n, mag
;error comp star n, obj mag, obj mag error
;This program should work for any number of comparison stars.

;First, we tell IDL the filter we have used.
;Because different files have capital or lowercase letters, we tell it both things.

  littlefilter='string'
print,"What filter are you using? Please use a lowercase letter."
read,littlefilter

  bigfilter='string'
print,"What filter are you using? Please use a capital letter."
read,bigfilter
;Each image will be a row in the array we are making.
;We find the number of images by looking in the filterobjects.out.lis file.

nrowfile=littlefilter+'objects.out.lis'
readcol,nrowfile,files,format='a'
nrow=n_elements(files)
print, 'number of rows=',nrow

;Determine the number of columns in the array.
;Each comp star and the object need two columns (for magnitude and magnitude error).
;We need one more column than that for the Julian date.
print,"How many comparison stars do you have?"
read,cstars
cstars=fix(cstars) ;Makes the number of comp stars an integer
ncolumn=2*cstars+3
print,'number of columns=',ncolumn

;This tells IDL the date both for looking for files and naming the saved file at the end.
date='string'
print,"What is the date? Please write as month day."
read,date

;Creating the array
star_struc=dblarr(nrow,ncolumn)
print, star_struc

;Read the Julian date for the array.
starfile='star0001_ '+bigfilter+'.dat'
print,starfile
readcol,starfile,jd,format='(d12.4,x,x)'
print,jd
star_struc[*,0]=jd
print, 'jd=',jd
;The snum file is used to tell the program which file it will be reading. This was created by binstar_B.pro
;The snum1 file should be a copied version of snum, with the beginning two zeros removed. It is used to number the mags and merrs later on.
readcol,'snum.lis',snum,format='a'
readcol,'snum1.lis',num,format='a'

;Now we start reading in all of the data from the star#### filter.dat files
;The while loop reads in all of the comp stars.

i=0

while i lt cstars do begin

  filestring='star'+snum(i)+'_'+bigfilter+'.dat'
magstring='mag'+num(i)
merrstring='merr'+num(i)
nmag=(2*(1+i))-1
nmerr=2*(1+i)
readcol,filestring,magstring,merrstring,format='(x,d6.3,d5.3)'
print,'file',filestring
print,'i=',i
print,'snum=',snum(i)
print,'num=',num(i)
print,'nmag=',nmag
print,'magstring=',magstring
print,'nmerr=',nmerr
print,'merrstring=',merrstring
star_struc[*,nmag]=magstring
star_struc[*,nmerr]=merrstring
i=1+i
print,'i=',i
end

;This reads in the object magnitude and magnitude error
filestring='star'+snum(i)+'_'+bigfilter+'.dat'
magstring='obj'
merrstring='objerror'
nmag=(2*(i+1)-1
nmerr=2*(1+i)
print,filestring
readcol,filestring,magstring,merrstring,format='(x,d6.3,d5.3)'
star_struc[*,*nmag]=magstring
star_struc[*,*nmerr]=merrstring
;Name we would like the array to have. Theoretically this would be unique.
;However we do not need the name to be unique until the last step.
shortname=date+'_isbigfilter'

;Uses the name shortname has been given to name the array we have just made
d=execute(shortname+”=star_struc")

;Save the data structure with format Date_Filter.sav
;This will still be called star_struc when it is restored.
savefile=date+'_isbigfilter'+'.sav'
save,star_struc,filenam=savefile

print,'Saved ',savefile
end

E.9 diffmagB

pro diffmag_B

;This program calculates the differential magnitude of the object.
;It combines the fluxes of the comparison stars to create a synthetic
;star, and subtracts the synthetic star’s magnitude from the
;comparison stars and object to obtain differential magnitudes.

;Determine file to restore. This should be the output from lcs_loop.pro
filter='string'
print,”What filter are you using? Please use a capital letter.”
read,filter
date='string'
print,”What is the date? Please write in the form month day.”
read,date
restore_file=date+’_’+filter+’.sav’
restore,restore_file

    print,”How many images do you have?”
read,nim
nims=nim-1

;Create the output data structure
print,”How many comp stars do you have?”
read,ncomp
ncol=ncomp+2 ;Need one column for the synthesized star, one for the object, and
one for each comp star

    rowfile=filter+’objects.out.lis’
readcol,rowfile,files,format=’a’
nrow=n_elements(files) ;Gives the same number of rows as star_struc

    cd_mags=dblarr(nrow,ncol)

    print, ’Created data structure’

;Calculate differential mags, one for each row (aka for each image). ’i’ indexes the
number of images

    for i=0,nims do begin

;Calculate the flux for each comparison star
flux1=10.0^(0.4*star_struc(i,1))
flux2=10.0^(0.4*star_struc(i,3))
flux3=10.0^(0.4*star_struc(i,5))
flux4=10.0^(0.4*star_struc(i,7))
flux5=10.0^(0.4*star_struc(i,9))
flux6=10.0^(0.4*star_struc(i,11))
flux7=10.0^(0.4*star_struc(i,13))
flux8=10.0^(0.4*star_struc(i,15))
flux9=10.0**(0.4*star_struc(i,17))
flux10=10.0**(0.4*star_struc(i,19))

;Calculate the average flux of the comparison stars
avgflux=(flux1+flux2+flux3+flux4+flux5+flux6+flux7+flux8+flux9+flux10)/10.0

print,'average flux=',avgflux
;Calculate the magnitude of the combined comp star from the average flux
compmag=2.5*alog10(avgflux)

print,'synthetic magnitude=',compmag

;Determine the differential magnitudes

diffmagobj=star_struc(i,21)-compmag ;Differential magnitude for object
diffmag1=star_struc(i,1)-compmag ;Differential magnitude for comp star1
diffmag2=star_struc(i,3)-compmag ;Differential magnitude for comp star2
diffmag3=star_struc(i,5)-compmag ;Differential magnitude for comp star3
diffmag4=star_struc(i,7)-compmag ;Differential magnitude for comp star4
diffmag5=star_struc(i,9)-compmag
diffmag6=star_struc(i,11)-compmag
diffmag7=star_struc(i,13)-compmag
diffmag8=star_struc(i,15)-compmag
diffmag9=star_struc(i,17)-compmag
diffmag10=star_struc(i,19)-compmag ;Differential magnitude for comp star10

print,diffmagobj
print,diffmag1
print,diffmag2
print,diffmag3
print,diffmag4
print,diffmag5
print,diffmag6
print,diffmag7
print,diffmag8
print,diffmag9
print,diffmag10
;Write data into the new data structure.
cd_mags[i,0]=compmag
cd_mags[i,1]=diffmagobj
cd_mags[i,2]=diffmag1
cd_mags[i,3]=diffmag2
cd_mags[i,4]=diffmag3
cd_mags[i,5]=diffmag4
cd_mags[i,6]=diffmag5
cd_mags[i,7]=diffmag6
cd_mags[i,8]=diffmag7
cd_mags[i,9]=diffmag8
cd_mags[i,10]=diffmag9
cd_mags[i,11]=diffmag10
endfor

savefile='cd_mags_’+filter+’’.sav’
save,cd_mags,filename=savefile
print,’Saved’,savefile
print,cd_mags
end

E.10 test lc

pro test_lc

;The purpose of this program is to create a test light curve in order
;to make sure none of the comparison stars is varying and to make sure
;that the object of interest is varying. It is currently set up for
;10 comparison stars but can be adjusted for different numbers of
;comparison stars.

;Set data structure containing Julian date and errors
filter=’string’
print,”What filter are you using? Please use a capital letter.”
read, filter
date='string'
print,"'What is the date? Please put in the form month_day.'"
read, date
lcds_file=date+'.'+filter+'.sav'
restore, lcds_file
jd_structure=star_struc

; Set data structure containing differential magnitudes
cd_mags_file='cd_mags.'+filter+'.sav'
restore, cd_mags_file
diff_mags=cd_mags

print, 'Can I get the differential magnitudes?', diff_mags(0,0)

; Determine maximum y value for initial plot
maxobj=max(diff_mags(*,1))
maxcomp1=max(diff_mags(*,2))
maxcomp2=max(diff_mags(*,3))
maxcomp3=max(diff_mags(*,4))
maxcomp4=max(diff_mags(*,5))
maxcomp5=max(diff_mags(*,6))
maxcomp6=max(diff_mags(*,7))
maxcomp7=max(diff_mags(*,8))
maxcomp8=max(diff_mags(*,9))
maxcomp9=max(diff_mags(*,10))
maxcomp10=max(diff_mags(*,11))

maxarray=[maxobj,maxcomp1,maxcomp2,maxcomp3,maxcomp4,$
maxcomp5,maxcomp6,maxcomp7,maxcomp8,maxcomp9,maxcomp10]
maxmag=max(maxarray)+0.1

print, 'Maxima are', maxarray
print, 'Maximum Differential Magnitude=', maxmag

; Determine minimum y value for initial plot
minobj=min(diff_mags(*,1))
mincomp1=min(diff_mags(*,2))
mincomp2=min(diff_mags(*,3))
mincomp3=min(diff_mags(*,4))
mincomp4=min(diff_mags(*,5))
mincomp5=min(diff_mags(*,6))
mincomp6=min(diff_mags(*,7))
mincomp7=min(diff_mags(*,8))
mincomp8=min(diff_mags(*,9))
mincomp9=min(diff_mags(*,10))
mincomp10=min(diff_mags(*,11))

minarray=[minobj,mincomp1,mincomp2,mincomp3,mincomp4,$
mincomp5,mincomp6,mincomp7,mincomp8,mincomp9,mincomp10] minmag=min(minarray)-0.1

print, 'Minima are', minarray
print, 'Minimum Differential Magnitude=',minmag

device,decompose=0
loadct,13

;Make an initial plot of the comp stars and object

print, 'About to start plotting...'  

plot,jd_structure(*,0),diff_mags(*,2),ystyle=1,psym=8,yr=[minmag,maxmag],color=150
print,'Comp star 1 Filled circle Green'
oplot,jd_structure(*,0),diff_mags(*,3),psym=4,color=80
print,'Comp star 2 Diamond Bright blue'
oplot,jd_structure(*,0),diff_mags(*,4),psym=5,color=120
print, 'Comp star 3 Triangle Teal’
oplot,jd_structure(*,0),diff_mags(*,5),psym=6,color=40
print, 'Comp star 4 Square Purple'
oplot,jd_structure(*,0),diff_mags(*,1),psym=7,color=200
print, 'Object X Yellow’
oplot,jd_structure(*,0),diff_mags(*,6),psym=2,color=100
print, 'Comp star 5 Asterisk Lighter blue’
print, 'Comp star 6 diamond Red'
oplot,jd_structure(*,0),diff_mags(*,7),psym=4,color=250
print,'Comp star 7 Triangle Goldenrod'
oplot,jd_structure(*,0),diff_mags(*,8),psym=5,color=225
print,'Comp star 8 Square Lighter blue'
oplot,jd_structure(*,0),diff_mags(*,9),psym=6,color=100
print,'Comp star 9 Filled Circle Blue'
oplot,jd_structure(*,0),diff_mags(*,10),psym=8,color=50
print,'Comp star 10 Plus sign Red'

print, 'Finished plotting.'

print, 'When finished looking at this, type .c to look at only the comparison stars.'

stop

maxcomparray=[maxcomp1,maxcomp2,maxcomp3,maxcomp4,maxcomp5,$
maxcomp6,maxcomp7,maxcomp8,maxcomp9,maxcomp10]
maxmagcomp=max(maxcomparray)+0.1

mincomparray=[mincomp1,mincomp2,mincomp3,mincomp4,mincomp5,$
mincomp6,mincomp7,mincomp8,mincomp9,mincomp10]
minmagcomp=min(mincomparray)-0.1

plot,jd_structure(*,0),diff_mags(*,2),ystyle=1,psym=8,yr=[minmagcomp,maxmagcomp],color=
print,'Comp star 1 Filled circle Green'
oplot,jd_structure(*,0),diff_mags(*,3),psym=4,color=80
print,'Comp star 2 Diamond Bright blue'
oplot,jd_structure(*,0),diff_mags(*,4),psym=5,color=120
print,'Comp star 3 Triangle Teal'
oplot,jd_structure(*,0),diff_mags(*,5),psym=6,color=40
print, 'Comp star 4 Square Purple'
oplot,jd_structure(*,0),diff_mags(*,1),psym=7,color=200
print, 'Object X Yellow'
oplot,jd_structure(*,0),diff_mags(*,6),psym=2,color=100
print, 'Comp star 5 Asterisk Lighter blue'
E.11  fig20.pro

pro fig20

;The purpose of this program is to create a file with a format similar
;to Figure 20 in Massey and Davis 1992

;Combine data from various files for the following format
;Column 1:  Field
;Column 2:  Filter
;Column 3:  Airmass (from filter_object.AM)
;Column 4:  Xcenter (from *.coo.0 column 1)
;Column 5:  Ycenter (from *.coo.0 column 2)
;Column 6:  instrumental magnitude (column 2 in star#### file)
;Column 7:  merr (column 3 in star#### file)

;Need one row for each star and each filter
;Total 10 files for PG0231 (5 stars)
;Total 11 files for RU149 (8 stars)
;TOTAL NUMBER OF ROWS=138

;star#### Filter_obj.dat has
; JD apparent_magnitude merr star#

; pg_allfiles.lis and ru_allfiles.lis have been made by:
; concatenating the filter_object.out.lis files and then
; sorting the concatenated file and piping that into these files
; They list all of the observations of the field in numerical order

; Define an data structure to have ALL of this stuff!!
array=field:strarr(138),filter:strarr(138),am:dblarr(138),xcentst:dblarr(138),$
ycentst:dblarr(138),mag:dblarr(138),merr:dblarr(138)

; Inserting the field/star names
for i=0,9 do array.field(i)='PG0231+051A'
for i=10,19 do array.field(i)='PG0231+051B'
for i=20,29 do array.field(i)='PG0231+051C'
for i=30,39 do array.field(i)='PG0231+051D'
for i=40,49 do array.field(i)='PG0231+051E'
for i=50,60 do array.field(i)='RU 149'
for i=61,71 do array.field(i)='RU 149A'
for i=72,82 do array.field(i)='RU 149B'
for i=83,93 do array.field(i)='RU 149C'
for i=94,104 do array.field(i)='RU 149D'
for i=105,115 do array.field(i)='RU 149E'
for i=116,126 do array.field(i)='RU 149F'
for i=127,137 do array.field(i)='RU 149G'

; Insert the filters...
array.filter=['B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$
'B','V','R','B','V','R','B','V','R','B','V','R',$$

81
;Get magnitudes, merrors, airmasses from files

;PG B filter
readcol,'star0001_B_pg.dat',mag_pgA_B,merr_pgA_B,format='(x,d6.3,d5.3)'
readcol,'star0002_B_pg.dat',mag_pgB_B,merr_pgB_B,format='(x,d6.3,d5.3)'
readcol,'star0003_B_pg.dat',mag_pgC_B,merr_pgC_B,format='(x,d6.3,d5.3)'
readcol,'star0004_B_pg.dat',mag_pgD_B,merr_pgD_B,format='(x,d6.3,d5.3)'
readcol,'star0005_B_pg.dat',mag_pgE_B,merr_pgE_B,format='(x,d6.3,d5.3)'
readcol,'b_pg0231.AM',am_pg_B,format='D'

;PG V filter
readcol,'star0001_V_pg.dat',mag_pgA_V,merr_pgA_V,format='(x,d6.3,d5.3)'
readcol,'star0002_V_pg.dat',mag_pgB_V,merr_pgB_V,format='(x,d6.3,d5.3)'
readcol,'star0003_V_pg.dat',mag_pgC_V,merr_pgC_V,format='(x,d6.3,d5.3)'
readcol,'star0004_V_pg.dat',mag_pgD_V,merr_pgD_V,format='(x,d6.3,d5.3)'
readcol,'star0005_V_pg.dat',mag_pgE_V,merr_pgE_V,format='(x,d6.3,d5.3)'
readcol,'v_pg0231.AM',am_pg_V,format='D'

;PG R filter
readcol,'star0001_R_pg.dat',mag_pgA_R,merr_pgA_R,format='(x,d6.3,d5.3)'
readcol,'star0002_R_pg.dat',mag_pgB_R,merr_pgB_R,format='(x,d6.3,d5.3)'
readcol,'star0003_R_pg.dat',mag_pgC_R,merr_pgC_R,format='(x,d6.3,d5.3)'
readcol,'star0004_R_pg.dat',mag_pgD_R,merr_pgD_R,format='(x,d6.3,d5.3)'
readcol,'star0005_R_pg.dat',mag_pgE_R,merr_pgE_R,format='(x,d6.3,d5.3)'
readcol,'r_pg0231.AM',am_pg_R,format='D'

;RU B filter
readcol,'star0001_B_ru.dat',mag_ru_B,merr_ru_B,format='(x,d6.3,d5.3)'
readcol,'star0002_B_ru.dat',mag_ruA_B,merr_ruA_B,format='(x,d6.3,d5.3)'
readcol,'star0003_B_ru.dat',mag_ruB_B,merr_ruB_B,format='(x,d6.3,d5.3)'
readcol,'star0004_B_ru.dat',mag_ruC_B,merr_ruC_B,format='(x,d6.3,d5.3)'
readcol,'star0005_B_ru.dat',mag_ruD_B,merr_ruD_B,format='(x,d6.3,d5.3)'
readcol,'star0006_B_ru.dat',mag_ruE_B,merr_ruE_B,format='(x,d6.3,d5.3)'
readcol,'star0007_B_ru.dat',mag_ruF_B,merr_ruF_B,format='(x,d6.3,d5.3)'
readcol,'star0008_B_ru.dat',mag_ruG_B,merr_ruG_B,format='(x,d6.3,d5.3)'
readcol,'b_ru149.AM',am_ru_B,format='D'

;RU V filter
readcol,'star0001_V_ru.dat',mag_ru V,merr_ru V,format='(x,d6.3,d5.3)'
readcol,'star0002_V_ru.dat',mag_ruA V,merr_ruA V,format='(x,d6.3,d5.3)'
readcol,'star0003_V_ru.dat',mag_ruB V,merr_ruB V,format='(x,d6.3,d5.3)'
readcol,'star0004_V_ru.dat',mag_ruC V,merr_ruC V,format='(x,d6.3,d5.3)'
readcol,'star0005_V_ru.dat',mag_ruD V,merr_ruD V,format='(x,d6.3,d5.3)'
readcol,'star0006_V_ru.dat',mag_ruE V,merr_ruE V,format='(x,d6.3,d5.3)'
readcol,'star0007_V_ru.dat',mag_ruF V,merr_ruF V,format='(x,d6.3,d5.3)'
readcol,'star0008_V_ru.dat',mag_ruG V,merr_ruG V,format='(x,d6.3,d5.3)'
readcol,'v_ru149.AM',am_ru V,format='D'
;RU R filter
readcol,'star0001_R_ru.dat',mag_ru R,merr_ru R,format='(x,d6.3,d5.3)'
readcol,'star0002_R_ru.dat',mag_ruA R,merr_ruA R,format='(x,d6.3,d5.3)'
readcol,'star0003_R_ru.dat',mag_ruB R,merr_ruB R,format='(x,d6.3,d5.3)'
readcol,'star0004_R_ru.dat',mag_ruC R,merr_ruC R,format='(x,d6.3,d5.3)'
readcol,'star0005_R_ru.dat',mag_ruD R,merr_ruD R,format='(x,d6.3,d5.3)'
readcol,'star0006_R_ru.dat',mag_ruE R,merr_ruE R,format='(x,d6.3,d5.3)'
readcol,'star0007_R_ru.dat',mag_ruF R,merr_ruF R,format='(x,d6.3,d5.3)'
readcol,'star0008_R_ru.dat',mag_ruG R,merr_ruG R,format='(x,d6.3,d5.3)'
readcol,'v_ru149.AM',am_ru R,format='D'

;Inserting the magnitudes, merrors and airmasses in the correct places
;PG A B filter
mpl=where((array.field EQ 'PG0231+051A') and (array.filter EQ 'B'))
n=n_elements(mag_pgA_B)
print,n
for i=0,n-1 do begin
  stop
  array.mag(mpl(i))=mag_pgA_B(i)
  array.merr(mpl(i))=merr_pgA_B(i)
  array.am(mpl(i))=am_pg_B(i)
endfor
;PG B B filter
mpl=where((array.field EQ 'PG0231+051B') and (array.filter EQ 'B'))
n=n_elements(mag_pgB_B)
for i=0,n-1 do begin
  array.mag(mpl(i))=mag_pgB_B(i)
  array.merr(mpl(i))=merr_pgB_B(i)
array.am(mpl(i))=am pg_B(i)
endfor

;PG C B FILTER
mpl=where((array.field EQ 'PG0231+051C') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag pg_C_B(i)
array.merr(mpl(i))=merr pg_C_B(i)
array.am(mpl(i))=am pg_B(i)
endfor

;PG D B FILTER
mpl=where((array.field EQ 'PG0231+051D') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag pg_D_B(i)
array.merr(mpl(i))=merr pg_D_B(i)
array.am(mpl(i))=am pg_B(i)
endfor

;PG E B FILTER
mpl=where((array.field EQ 'PG0231+051E') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag pg_E_B(i)
array.merr(mpl(i))=merr pg_E_B(i)
array.am(mpl(i))=am pg_B(i)
endfor

;PG A V filter
mpl=where((array.field EQ 'PG0231+051A') and (array.filter EQ 'V'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag pg_A_V(i)
array.merr(mpl(i))=merr pg_A_V(i)
array.am(mpl(i))=am pg_V(i)
endfor

;PG B V filter
mpl=where((array.field EQ 'PG0231+051B') and (array.filter EQ 'V'))
n=n_elements(mag_pgB_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_pgB_V(i)
array.merr(mpl(i))=merr_pgB_V(i)
array.am(mpl(i))=am_pg_V(i)
endfor
;PG C V FILTER
mpl=where((array.field EQ 'PG0231+051C') and (array.filter EQ 'V'))
n=n_elements(mag_pgC_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_pgC_V(i)
array.merr(mpl(i))=merr_pgC_V(i)
array.am(mpl(i))=am_pg_V(i)
endfor
;PG D V FILTER
mpl=where((array.field EQ 'PG0231+051D') and (array.filter EQ 'V'))
n=n_elements(mag_pgD_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_pgD_V(i)
array.merr(mpl(i))=merr_pgD_V(i)
array.am(mpl(i))=am_pg_V(i)
endfor
;PG E V FILTER
mpl=where((array.field EQ 'PG0231+051E') and (array.filter EQ 'V'))
n=n_elements(mag_pgE_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_pgE_V(i)
array.merr(mpl(i))=merr_pgE_V(i)
array.am(mpl(i))=am_pg_V(i)
endfor
;PG A R filter
mpl=where((array.field EQ 'PG0231+051A') and (array.filter EQ 'R'))
n=n_elements(mag_pgA_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_pgA_R(i)
array.merr(mpl(i))=merr_pgA_R(i)
array.am(mpl(i))=am Pg_R(i)
endfor
;PG B R filter
mpl=where((array.field EQ 'PG0231+051B') and (array.filter EQ 'R'))
n=n_elements(mag pgB_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag pgB_R(i)
array.merr(mpl(i))=merr pgB_R(i)
array.am(mpl(i))=am pg_R(i)
endfor
;PG C R FILTER
mpl=where((array.field EQ 'PG0231+051C') and (array.filter EQ 'R'))
n=n_elements(mag pgC_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag pgC_R(i)
array.merr(mpl(i))=merr pgC_R(i)
array.am(mpl(i))=am pg_R(i)
endfor
;PG D R FILTER
mpl=where((array.field EQ 'PG0231+051D') and (array.filter EQ 'R'))
n=n_elements(mag pgD_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag pgD_R(i)
array.merr(mpl(i))=merr pgD_R(i)
array.am(mpl(i))=am pg_R(i)
endfor
;PG E R FILTER
mpl=where((array.field EQ 'PG0231+051E') and (array.filter EQ 'R'))
n=n_elements(mag pgE_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag pgE_R(i)
array.merr(mpl(i))=merr pgE_R(i)
array.am(mpl(i))=am pg_R(i)
endfor
;RU B FILTER
mpl=where((array.field EQ 'RU J49') and (array.filter EQ 'B'))
n=n_elements(mag_ru_B)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ru_B(i)
array.merr(mpl(i))=merr_ru_B(i)
array.am(mpl(i))=am_ru_B(i)
endfor

;RU A B FILTER
mpl=where((array.field EQ 'RU J49A') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruA_B(i)
array.merr(mpl(i))=merr_ruA_B(i)
array.am(mpl(i))=am_ru_B(i)
endfor

;RU B B FILTER
mpl=where((array.field EQ 'RU J49B') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruB_B(i)
array.merr(mpl(i))=merr_ruB_B(i)
array.am(mpl(i))=am_ru_B(i)
endfor

;RU C B FILTER
mpl=where((array.field EQ 'RU J49C') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruC_B(i)
array.merr(mpl(i))=merr_ruC_B(i)
array.am(mpl(i))=am_ru_B(i)
endfor

;RU D B FILTER
mpl=where((array.field EQ 'RU J49D') and (array.filter EQ 'B'))
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruD_B(i)
array.merr(mpl(i))=merr_ruD_B(i)
array.am(mpl(i))=am_ru_B(i)
endfor
;RU E B FILTER
mpl=where((array.field EQ 'RU J49E') and (array.filter EQ 'B'))
n=n _elements(mag _ruE _B)
for i=0,n-1 do begin
array.mag(mpl(i))=mag _ruE _B(i)
array.merr(mpl(i))=merr _ruE _B(i)
array.am(mpl(i))=am _ru _B(i)
endfor
;RU F B FILTER
mpl=where((array.field EQ 'RU J49F') and (array.filter EQ 'B'))
n=n _elements(mag _ruF _B)
for i=0,n-1 do begin
array.mag(mpl(i))=mag _ruF _B(i)
array.merr(mpl(i))=merr _ruF _B(i)
array.am(mpl(i))=am _ru _B(i)
endfor
;RU G B FILTER
mpl=where((array.field EQ 'RU J49G') and (array.filter EQ 'B'))
n=n _elements(mag _ruG _B)
for i=0,n-1 do begin
array.mag(mpl(i))=mag _ruG _B(i)
array.merr(mpl(i))=merr _ruG _B(i)
array.am(mpl(i))=am _ru _B(i)
endfor
;RU V FILTER
mpl=where((array.field EQ 'RU J49') and (array.filter EQ 'V'))
n=n _elements(mag _ru _V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag _ru _V(i)
array.merr(mpl(i))=merr _ru _V(i)
array.am(mpl(i))=am _ru _V(i)
endfor
;RU A V FILTER
mpl=where((array.field EQ 'RU J49A') and (array.filter EQ 'V'))
n=n _elements(mag _ruA _V)
for i=0, n-1 do begin
  array.mag(mpl(i)) = mag_RuA_V(i)
  array.merr(mpl(i)) = merr_RuA_V(i)
  array.am(mpl(i)) = am_Ru_V(i)
endfor
;RU B V FILTER
mpl = where((array.field EQ 'RU 149B') and (array.filter EQ 'V'))
n = n_elements(mag_RuB_V)
for i=0, n-1 do begin
  array.mag(mpl(i)) = mag_RuB_V(i)
  array.merr(mpl(i)) = merr_RuB_V(i)
  array.am(mpl(i)) = am_Ru_V(i)
endfor
;RU C V FILTER
mpl = where((array.field EQ 'RU 149C') and (array.filter EQ 'V'))
n = n_elements(mag_RuC_V)
for i=0, n-1 do begin
  array.mag(mpl(i)) = mag_RuC_V(i)
  array.merr(mpl(i)) = merr_RuC_V(i)
  array.am(mpl(i)) = am_Ru_V(i)
endfor
;RU D V FILTER
mpl = where((array.field EQ 'RU 149D') and (array.filter EQ 'V'))
n = n_elements(mag_RuD_V)
for i=0, n-1 do begin
  array.mag(mpl(i)) = mag_RuD_V(i)
  array.merr(mpl(i)) = merr_RuD_V(i)
  array.am(mpl(i)) = am_Ru_V(i)
endfor
;RU E V FILTER
mpl = where((array.field EQ 'RU 149E') and (array.filter EQ 'V'))
n = n_elements(mag_RuE_V)
for i=0, n-1 do begin
  array.mag(mpl(i)) = mag_RuE_V(i)
  array.merr(mpl(i)) = merr_RuE_V(i)
  array.am(mpl(i)) = am_Ru_V(i)
endfor
;RU F V FILTER
mpl=where((array.field EQ 'RU J49F') and (array.filter EQ 'V'))
n=n_elements(mag_ruF_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruF_V(i)
array.merr(mpl(i))=merr_ruF_V(i)
array.am(mpl(i))=am_ru_V(i)
endfor

;RU G V FILTER
mpl=where((array.field EQ 'RU J49G') and (array.filter EQ 'V'))
n=n_elements(mag_ruG_V)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruG_V(i)
array.merr(mpl(i))=merr_ruG_V(i)
array.am(mpl(i))=am_ru_V(i)
endfor

;RU R FILTER
mpl=where((array.field EQ 'RU J49') and (array.filter EQ 'R'))
n=n_elements(mag_ru_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ru_R(i)
array.merr(mpl(i))=merr_ru_R(i)
array.am(mpl(i))=am_ru_R(i)
endfor

;RU A R FILTER
mpl=where((array.field EQ 'RU J49A') and (array.filter EQ 'R'))
n=n_elements(mag_ruA_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruA_R(i)
array.merr(mpl(i))=merr_ruA_R(i)
array.am(mpl(i))=am_ru_R(i)
endfor

;RU B R FILTER
mpl=where((array.field EQ 'RU J49B') and (array.filter EQ 'R'))
n=n_elements(mag_ruB_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_rub R(i)
array.merr(mpl(i))=merr_rub R(i)
array.am(mpl(i))=am_ru R(i)
endfor
;RU C R FILTER
mpl=where((array.field EQ 'RU 149C') and (array.filter EQ 'R'))
n=n_elements(mag_ruC R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruC R(i)
array.merr(mpl(i))=merr_ruC R(i)
array.am(mpl(i))=am_ru R(i)
endfor
;RU D R FILTER
mpl=where((array.field EQ 'RU 149D') and (array.filter EQ 'R'))
n=n_elements(mag_ruD R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruD R(i)
array.merr(mpl(i))=merr_ruD R(i)
array.am(mpl(i))=am_ru R(i)
endfor
;RU E R FILTER
mpl=where((array.field EQ 'RU 149E') and (array.filter EQ 'R'))
n=n_elements(mag_ruE R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruE R(i)
array.merr(mpl(i))=merr_ruE R(i)
array.am(mpl(i))=am_ru R(i)
endfor
;RU F R FILTER
mpl=where((array.field EQ 'RU 149F') and (array.filter EQ 'R'))
n=n_elements(mag_ruF R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruF R(i)
array.merr(mpl(i))=merr_ruF R(i)
array.am(mpl(i))=am_ru R(i)
endfor
;RU G R FILTER
mpl=where((array.field EQ 'RU J49G') and (array.filter EQ 'R'))
n=n_elements(mag_ruG_R)
for i=0,n-1 do begin
array.mag(mpl(i))=mag_ruG_R(i)
array.merr(mpl(i))=merr_ruG_R(i)
array.am(mpl(i))=am_ru_R(i)
endfor

;Get x and y coordinates of stars in each image

;Obtain coordinates for stars in PG0231 field
readcol,'pg_allfiles.lis',pg_coordfiles,format='A'

    pgA=where(array.field EQ 'PG0231+051A')
    pgB=where(array.field EQ 'PG0231+051B')
    pgC=where(array.field EQ 'PG0231+051C')
    pgD=where(array.field EQ 'PG0231+051D')
    pgE=where(array.field EQ 'PG0231+051E')

    n=n_elements(pg_coordfiles)
print,'Number of *.coo.0 files is',n

j=0
for i=0,(n-1) do begin
readcol,pg_coordfiles(i),xcent,ycent,format='D'
array.xcentst(pgA(j))=xcent(0)
array.ycentst(pgA(j))=ycent(0)
array.xcentst(pgB(j))=xcent(1)
array.ycentst(pgB(j))=ycent(1)
array.xcentst(pgC(j))=xcent(2)
array.ycentst(pgC(j))=ycent(2)
array.xcentst(pgD(j))=xcent(3)
array.ycentst(pgD(j))=ycent(3)
array.xcentst(pgE(j))=xcent(4)
array.ycentst(pgE(j))=ycent(4)
j=j+1
endfor
Obtain coordinates for stars in RU149 field
readcol,'ru_allfiles.lis',ru_coordfiles,format='A'

ru=where(array.field EQ 'RU_149')
ruA=where(array.field EQ 'RU_149A')
ruB=where(array.field EQ 'RU_149B')
ruC=where(array.field EQ 'RU_149C')
ruD=where(array.field EQ 'RU_149D')
ruE=where(array.field EQ 'RU_149E')
ruF=where(array.field EQ 'RU_149F')
ruG=where(array.field EQ 'RU_149G')

n=n_elements(ru_coordfiles)
print,'Number of *.coo.0 files is',n

j=0
for i=0,(n-1) do begin
readcol,ru_coordfiles(i),xcent,ycent,format='D'
array.xcentst(ru(j))=xcent(0)
array.ycentst(ru(j))=ycent(0)
array.xcentst(ruA(j))=xcent(1)
array.ycentst(ruA(j))=ycent(1)
array.xcentst(ruB(j))=xcent(2)
array.ycentst(ruB(j))=ycent(2)
array.xcentst(ruC(j))=xcent(3)
array.ycentst(ruC(j))=ycent(3)
array.xcentst(ruD(j))=xcent(4)
array.ycentst(ruD(j))=ycent(4)
array.xcentst(ruE(j))=xcent(5)
array.ycentst(ruE(j))=ycent(5)
array.xcentst(ruF(j))=xcent(6)
array.ycentst(ruF(j))=ycent(6)
array.xcentst(ruG(j))=xcent(7)
array.ycentst(ruG(j))=ycent(7)
j=j+1
endfor
;Now the names need to be reformatted since we’ve put everything else in!
;Inserting the field/star names
array.field(0)=’PG0231+051A’
for i=1,9 do array.field(i)=’*’
array.field(10)=’PG0231+051B’
for i=11,19 do array.field(i)=’*’
array.field(20)=’PG0231+051C’
for i=21,29 do array.field(i)=’*’
array.field(30)=’PG0231+051D’
for i=31,39 do array.field(i)=’*’
array.field(40)=’PG0231+051E’
for i=41,49 do array.field(i)=’*’
array.field(50)=’RU_149’
for i=51,60 do array.field(i)=’*’
array.field(61)=’RU_149A’
for i=62,71 do array.field(i)=’*’
array.field(72)=’RU_149B’
for i=73,82 do array.field(i)=’*’
array.field(83)=’RU_149C’
for i=84,93 do array.field(i)=’*’
array.field(94)=’RU_149D’
for i=95,104 do array.field(i)=’*’
array.field(105)=’RU _149E’
for i=106,115 do array.field(i)=’*’
array.field(116)=’RU _149F’
for i=117,126 do array.field(i)=’*’
array.field(127)=’RU _149G’
for i=128,137 do array.field(i)=’*’

openw,fig20,’standobs’,/getLun
for i=0,137 do printf,fig20,array.field(i),array.filter(i),array.am(i),$
array.xcentst(i),array.ycentst(i),array.mag(i),$
array.merr(i),format=’(A,2x,A,D,D,D,D,D,D)’
close,fig20
E.12  stand fit.pro

;This program calculates the standard magnitudes of the comparison stars on the
;photometric night (Jan 21, 2011) based on the transformation
;coefficients calculated by IRAF

pro stand_fit

;Write in the parameters we obtained from fitting the standard
;solutions

;B filter b1=-
2.096385 b2=-
0.1855268 b3=-
0.05669202

;V filter v1=-
2.312695
v2=-0.005434302
v3=0.06113653

;R filter r1=-
2.532774 r2=0.163554
r3=0.1612175

;Need airmass files for the comp stars
;This is the airmass for each observation
readcol,’bobjects.AM’,XB
readcol,’vobjects.AM’,XV
readcol,’robjects.AM’,XR

;Finally need instrumental magnitudes
readcol,’star0001 _ B.dat’,mB1,format=('(x,d6.3)’
readcol,’star0002 _ B.dat’,mB2,format=('(x,d6.3)’

end
readcol,'star0003_B.dat',mB3,format='(x,d6.3)'
readcol,'star0004_B.dat',mB4,format='(x,d6.3)'
readcol,'star0005_B.dat',mB5,format='(x,d6.3)'
readcol,'star0006_B.dat',mB6,format='(x,d6.3)'
readcol,'star0007_B.dat',mB7,format='(x,d6.3)'
readcol,'star0008_B.dat',mB8,format='(x,d6.3)'
readcol,'star0009_B.dat',mB9,format='(x,d6.3)'
readcol,'star0010_B.dat',mB10,format='(x,d6.3)'

readcol,'star0001_V.dat',mV1,format='(x,d6.3)'
readcol,'star0002_V.dat',mV2,format='(x,d6.3)'
readcol,'star0003_V.dat',mV3,format='(x,d6.3)'
readcol,'star0004_V.dat',mV4,format='(x,d6.3)'
readcol,'star0005_V.dat',mV5,format='(x,d6.3)'
readcol,'star0006_V.dat',mV6,format='(x,d6.3)'
readcol,'star0007_V.dat',mV7,format='(x,d6.3)'
readcol,'star0008_V.dat',mV8,format='(x,d6.3)'
readcol,'star0009_V.dat',mV9,format='(x,d6.3)'
readcol,'star0010_V.dat',mV10,format='(x,d6.3)'

readcol,'star0001_R.dat',mR1,format='(x,d6.3)'
readcol,'star0002_R.dat',mR2,format='(x,d6.3)'
readcol,'star0003_R.dat',mR3,format='(x,d6.3)'
readcol,'star0004_R.dat',mR4,format='(x,d6.3)'
readcol,'star0005_R.dat',mR5,format='(x,d6.3)'
readcol,'star0006_R.dat',mR6,format='(x,d6.3)'
readcol,'star0007_R.dat',mR7,format='(x,d6.3)'
readcol,'star0008_R.dat',mR8,format='(x,d6.3)'
readcol,'star0009_R.dat',mR9,format='(x,d6.3)'
readcol,'star0010_R.dat',mR10,format='(x,d6.3)'

; We find the real magnitudes of the stars in order.

; COMP star 1
BB1=(mB1-b1-(b2*XB)+(b3*mV1/(1-v3))-(b3*v1/(1-v3))-(b3*v2*XV/(1-v3)))*((1-v3)/(((1-v3)*(1+b3)))+b3*v3))
VV1=(1/(1-v3))*(mV1-v1-(v2*XV)-(v3*BB1))
RR1 = (1/(1-r3))*(mR1-r1-(r2*XR)-(r3*VV1))
print, mean(BB1), mean(VV1), mean(RR1)
stop

;Comp star 2
BB2 = (mB2-b1-(b2*XB)+(b3*mV2/(1-v3))-(b3*v1/(1-v3))-(b3*v2*XV/(1-v3)))*((1-v3)/(((1-v3)*(1+b3))+(b3*v3))
VV2 = (1/(1-v3))*(mV2-v1-(v2*XY)-(v3*BB2))
RR2 = (1/(1-r3))*(mR2-r1-(r2*XR)-(r3*VV2))
print, mean(BB2), mean(VV2), mean(RR2)
stop

;Comp star 3
BB3 = (mB3-b1-(b2*XB)+(b3*mV3/(1-v3))-(b3*v1/(1-v3))-(b3*v2*XV/(1-v3)))*((1-v3)/(((1-v3)*(1+b3))+(b3*v3))
VV3 = (1/(1-v3))*(mV3-v1-(v2*XY)-(v3*BB3))
RR3 = (1/(1-r3))*(mR3-r1-(r2*XR)-(r3*VV3))
print, mean(BB3), mean(VV3), mean(RR3)
stop

;Comp star 4
BB4 = (mB4-b1-(b2*XB)+(b3*mV4/(1-v3))-(b3*v1/(1-v3))-(b3*v2*XV/(1-v3)))*((1-v3)/(((1-v3)*(1+b3))+(b3*v3))
VV4 = (1/(1-v3))*(mV4-v1-(v2*XY)-(v3*BB4))
RR4 = (1/(1-r3))*(mR4-r1-(r2*XR)-(r3*VV4))
print, mean(BB4), mean(VV4), mean(RR4)
stop

;Comp star 5
BB5 = (mB5-b1-(b2*XB)+(b3*mV5/(1-v3))-(b3*v1/(1-v3))-(b3*v2*XV/(1-v3)))*((1-v3)/(((1-v3)*(1+b3))+(b3*v3))
VV5 = (1/(1-v3))*(mV5-v1-(v2*XY)-(v3*BB5))
RR5 = (1/(1-r3))*(mR5-r1-(r2*XR)-(r3*VV5))
print, mean(BB5), mean(VV5), mean(RR5)
stop

;Comp star 6
$BB6 = (mB6 - b1 - (b2 * XB) + (b3 * mV6 / (1 - v3)) - (b3 * v1 / (1 - v3)) - (b3 * v2 * XV / (1 - v3))) \times ((1 - v3) / (((1 - v3) * (1 + b3)) + b3 * v3))$

$VV6 = (1 / (1 - v3)) \times (mV6 - v1 - (v2 * XV) - (v3 * BB6))$

$RR6 = (1 / (1 - r3)) \times (mR6 - r1 - (r2 * XR) - (r3 * VV6))$

print, mean(BB6), mean(VV6), mean(RR6)

stop

; Comp star 7

$BB7 = (mB7 - b1 - (b2 * XB) + (b3 * mV7 / (1 - v3)) - (b3 * v1 / (1 - v3)) - (b3 * v2 * XV / (1 - v3))) \times ((1 - v3) / (((1 - v3) * (1 + b3)) + b3 * v3))$

$VV7 = (1 / (1 - v3)) \times (mV7 - v1 - (v2 * XV) - (v3 * BB7))$

$RR7 = (1 / (1 - r3)) \times (mR7 - r1 - (r2 * XR) - (r3 * VV7))$

print, mean(BB7), mean(VV7), mean(RR7)

stop

; Comp star 8

$BB8 = (mB8 - b1 - (b2 * XB) + (b3 * mV8 / (1 - v3)) - (b3 * v1 / (1 - v3)) - (b3 * v2 * XV / (1 - v3))) \times ((1 - v3) / (((1 - v3) * (1 + b3)) + b3 * v3))$

$VV8 = (1 / (1 - v3)) \times (mV8 - v1 - (v2 * XV) - (v3 * BB8))$

$RR8 = (1 / (1 - r3)) \times (mR8 - r1 - (r2 * XR) - (r3 * VV8))$

print, mean(BB8), mean(VV8), mean(RR8)

stop

; Comp star 9

$BB9 = (mB9 - b1 - (b2 * XB) + (b3 * mV9 / (1 - v3)) - (b3 * v1 / (1 - v3)) - (b3 * v2 * XV / (1 - v3))) \times ((1 - v3) / (((1 - v3) * (1 + b3)) + b3 * v3))$

$VV9 = (1 / (1 - v3)) \times (mV9 - v1 - (v2 * XV) - (v3 * BB9))$

$RR9 = (1 / (1 - r3)) \times (mR9 - r1 - (r2 * XR) - (r3 * VV9))$

print, mean(BB9), mean(VV9), mean(RR9)

stop

; Comp star 10

$BB10 = (mB10 - b1 - (b2 * XB) + (b3 * mV10 / (1 - v3)) - (b3 * v1 / (1 - v3)) - (b3 * v2 * XV / (1 - v3))) \times ((1 - v3) / (((1 - v3) * (1 + b3)) + b3 * v3))$

$VV10 = (1 / (1 - v3)) \times (mV10 - v1 - (v2 * XV) - (v3 * BB10))$

$RR10 = (1 / (1 - r3)) \times (mR10 - r1 - (r2 * XR) - (r3 * VV10))$

print, mean(BB10), mean(VV10), mean(RR10)
E.13 GM Ori standards B.pro

pro GM Ori standards B

;The purpose of this program is to put measurements of GM Ori in the B filter onto a standard scale. Real magnitudes were calculated from the night of January 21, 2011.

;Real magnitudes of the comp stars calculated using stand_fit.pro and the transformation coefficients.
;These do have to be typed by hand :
realmag1=13.908345
realmag2=14.517664
realmag3=13.754667
realmag4=12.705737
realmag5=14.768410
realmag6=12.057684
realmag7=13.149097
realmag8=12.724018
realmag9=12.457963
realmag10=13.324614

;Calculate the flux of the real magnitudes
realflux1=10.0^0.4*realmag1
realflux2=10.0^0.4*realmag2
realflux3=10.0^0.4*realmag3
realflux4=10.0^0.4*realmag4
realflux5=10.0^0.4*realmag5
realflux6=10.0^0.4*realmag6
realflux7=10.0^0.4*realmag7
realflux8=10.0^0.4*realmag8
realflux9 = 10.0^((0.4 * realmag9))
realflux10 = 10.0^((0.4 * realmag10))

; Calculate the average flux of the comp stars
avg_cflux = (realflux1 + realflux2 + realflux3 + realflux4 + realflux5 + realflux6 + realflux7 + realflux8 + realflux9 + realflux10) / 10.0

; Calculate the real average comp magnitude
avg_comp_mag = 2.5 * alog10(avg_cflux)
print, 'Real average comp magnitude=', avg_comp_mag

; Find the filter being used
filter = 'string'
print, "What filter are you using? Please use a capital letter."
read, filter

sfilter = 'string'
print, "What filter are you using? Please use a lowercase letter."
read, sfilter

; Find date of data
date = 'string'
print, "What is the date? Please write in the form month day."
read, date

; Restore the important files

; Restore the star struc save file made by lcds loop
restore_ss = date + '_' + filter + '.sav'
restore, restore_ss

; Restore the cd mags save file made by diffmag B
restore_cdm = 'cd_mags_-' + filter + '.sav'
restore, restore_cdm

; Determine differential magnitude

object_real = cd_mags(*,1) + avg_comp_mag
rowfile=sfilter+'objects.out.lis'
readcol,rowfile,files,format='a'
nrow=n_elements(files)
print,’number of rows=’,nrow

obj_filter_real=dblarr(nrow,2)
obj_filter_real[*,0]=star.struc[*,0]
obj_filter_real[*,1]=object.real

print,’The program will now enter a stop.’
print,’You should use this to save the data structure with a unique name.’
print,’First, let something of the form obj_filter.date=obj_filter.real’
print,’In this case, you should FILL IN the appropriate object, filter, and date yourself.’
print,’Then use the save function:’
print,’save,obj_filter.date,filename=obj_filter.date.real.sav’
print,’MAKE SURE that you put the file name in single quotes!!!’
print,’Once you have saved, type .c to continue running the program.’

stop

print,’Continuing from stop.’

print,’The program has reached its end.’

end

E.14 GMOri standards V.pro

pro GMOri_standards_V

;The purpose of this program is to put measurements of GM Ori in the V
;filter onto a standard scale. Real magnitudes were calculated from
;the night of January 21, 2011.
;Real magnitudes of the comp stars calculated using stand_fit.pro and
;the transformation coefficients.
;These do have to be typed by hand :(
realmag1=12.167273
realmag2=13.923384
realmag3=13.319086
realmag4=12.292449
realmag5=13.320176
realmag6=11.892270
realmag7=12.035643
realmag8=12.317532
realmag9=12.287636
realmag10=13.084006

;Calculate the flux of the real magnitudes
realflux1=10.0^(0.4*realmag1)
realflux2=10.0^(0.4*realmag2)
realflux3=10.0^(0.4*realmag3)
realflux4=10.0^(0.4*realmag4)
realflux5=10.0^(0.4*realmag5)
realflux6=10.0^(0.4*realmag6)
realflux7=10.0^(0.4*realmag7)
realflux8=10.0^(0.4*realmag8)
realflux9=10.0^(0.4*realmag9)
realflux10=10.0^(0.4*realmag10)

;Calculate the average flux of the comp stars
avg_cflux=(realflux1+realflux2+realflux3+realflux4+realflux5+$
realflux6+realflux7+realflux8+realflux9+realflux10)/10.0

;Calculate the real average comp magnitude
avg_comp_mag=2.5*aalog10(avg_cflux)

;Find the filter being used
filter='string'
print,"What filter are you using? Please use a capital letter."
read, filter

sfilt='string'
print("What filter are you using? Please use a lowercase letter.")
read,sfilt

;Find date of data
date='string'
print("What is the date? Please write in the form month day.")
read, date

;Restore the important files

;Restore the star struct save file made by lcls_loop
restore ss=\date+’_'+sfilter+'.sav'
restore,restore ss

;Restore the cd mags sasve file made by diffmag B
restore cdm=’cd mags ’+sfilter+'.sav'
restore,restore cdm

;Determining differential magnitude

object_real=cd mags(*,1)+avg comp mag

rowfile=sfilt+’objects.out.lis’
readcol,rowfile,files,format=’a’
nrow=n_elements(files)
print,’number of rows=’,nrow

obj_filter_real=dblarr(nrow,2)
obj_filter_real[*,0]=star struct[*,0]
obj_filter_real[*,1]=object real

print,’The program will now enter a stop.’
print,’You should use this to save the data structure with a unique name.’
print,’First, let something of the form obj_filter_date=obj filter real’
print,’In this case, you should FILL IN the appropriate object, filter, and date your-
self.
print,'Then use the save function:
print,'save, obj_filter_date, filename=obj_filter_date_real.sav'
print,'MAKE SURE that you put the file name in single quotes!!!'
print,'Once you have saved, type .c to continue running the program.'

stop

print,'Continuing from stop.'

print,'The program has reached its end.'

end

E.15 GM Ori standards R.pro

pro GM Ori standards_R

;The purpose of this program is to put measurements of GM Ori in the R
;filter onto a standard scale. Real magnitudes were calculated from
;the night of January 21, 2011.

;Real magnitudes of the comp stars calculated using stand_fit.pro and
;the transformation coefficients.
;These do have to be typed by hand :(
realmag1=11.093669
realmag2=13.489464
realmag3=12.965399
realmag4=11.928791
realmag5=12.395316
realmag6=11.699708
realmag7=11.302171
realmag8=11.982786
realmag9=12.078742
realmag10=12.798767
;Calculate the flux of the real magnitudes

realflux1=10.0^(0.4*realmag1)
realflux2=10.0^(0.4*realmag2)
realflux3=10.0^(0.4*realmag3)
realflux4=10.0^(0.4*realmag4)
realflux5=10.0^(0.4*realmag5)
realflux6=10.0^(0.4*realmag6)
realflux7=10.0^(0.4*realmag7)
realflux8=10.0^(0.4*realmag8)
realflux9=10.0^(0.4*realmag9)
realflux10=10.0^(0.4*realmag10)

;Calculate the average flux of the comp stars
avg_cflux=(realflux1+realflux2+realflux3+realflux4+realflux5+$
realflux6+realflux7+realflux8+realflux9+realflux10)/10.0

;Calculate the real average comp magnitude
avg_comp_mag=2.5*alog10(avg_cflux)

;Find the filter being used
filter='string'
print,”What filter are you using? Please use a capital letter.”
read, filter

sfilt='string'
print,”What filter are you using? Please use a lowercase letter.”
read,sfilter

;Find date of data
date='string'
print,”What is the date? Please write in the form month day.”
read, date

;Restore the important files

;Restore the star struct save file made by lcls loop
restore ss=date+'_'+filter+'.sav'
restore,restore ss
;Restore the cd mags sasve file made by diffmag B
restore_cdm='cd mags '+'filter+'+.sav'
restore,restore_cdm

;Determining differential magnitude

object_real=cd_mags(*,1)+avg_comp_mag

rowfile=sfilter+'objects.out.lis'
readcol,rowfile,files,format='a'
nrow=n_elements(files)
print,'number of rows=',nrow

obj_filter_real=dblarr(nrow,2)
obj_filter_real[*,0]=star_struct[*,0]
obj_filter_real[*,1]=object_real

print,'The program will now enter a stop.'
print,'You should use this to save the data structure with a unique name.'
print,'First, let something of the form obj_filter_date=obj_filter_real'
print,'In this case, you should FILL IN the appropriate object, filter, and date yourself.'
print,'Then use the save function:'
print,'save,obj_filter_date,filename=obj_filter_date_real.sav'
print,'MAKE SURE that you put the file name in single quotes!!!'
print,'Once you have saved, type .c to continue running the program.'

stop

print,'Continuing from stop.'

print,'The program has reached its end.'
The purpose of this program is to create light curves for GM Ori from the January and October 2011 data. While this program is not terribly useful for anyone else as such it provides a template for how the program can be set up.

;Restore ALL the files
restore, 'GMOri_B_Jan19_real.sav'
restore, 'GMOri_B_Jan20_real.sav'
restore, 'GMOri_B_Jan21_real.sav'
restore, 'GMOri_B_Oct16_real.sav'
restore, 'GMOri_B_Oct17_real.sav'
restore, 'GMOri_B_Oct18_real.sav'
restore, 'GMOri_V_Jan19_real.sav'
restore, 'GMOri_V_Jan20_real.sav'
restore, 'GMOri_V_Jan21_real.sav'
restore, 'GMOri_V_Oct16_real.sav'
restore, 'GMOri_V_Oct17_real.sav'
restore, 'GMOri_V_Oct18_real.sav'
restore, 'GMOri_R_Jan19_real.sav'
restore, 'GMOri_R_Jan20_real.sav'
restore, 'GMOri_R_Jan21_real.sav'
restore, 'GMOri_R_Oct16_real.sav'
restore, 'GMOri_R_Oct17_real.sav'
restore, 'GMOri_R_Oct18_real.sav'

;Determine the range of julian dates that will be used
minjdB=min(GMOri_B_Jan19(*,0)),format='(d12.4)'
maxjdB=max(GMOri_B_Oct18(*,0)),format='(d12.4)'

;Set up data structure for B filter

;Number of observations in the B filter will be how many elements each
; part of the data structure must have
nobsB=n_elements(GMOri_B_Jan19(*,0))+n_elements(GMOri_B_Jan20(*,0)) +
           n_elements(GMOri_B_Jan21(*,0)) +$
           n_elements(GMOri_B_Oct16(*,0))+n_elements(GMOri_B_Oct17(*,0)) +$
           n_elements(GMOri_B_Oct18(*,0))
; Define the data structure
GMOri_final_B=phase:dblarr(nobsB),jd:dblarr(nobsB),realmags:dblarr(nobsB)
; JD inserted
GMOri_final_B.jd=[GMOri_B_Jan19(*,0), GMOri_B_Jan20(*,0), GMOri_B_Jan21(*,0),$
                 GMOri_B_Oct16(*,0), GMOri_B_Oct17(*,0), GMOri_B_Oct18(*,0)]
; Magnitudes inserted
GMOri_final_B.realmags=[GMOri_B_Jan19(*,1),$
                    GMOri_B_Jan20(*,1), GMOri_B_Jan21(*,1),$
                    GMOri_B_Oct16(*,1), GMOri_B_Oct17(*,1), GMOri_B_Oct18(*,1)]
; Calculate phase
period=0.3905135
specJD=8580.836
pha=((GMOri_final_B.jd-specJD) mod period)/period
GMOri_final_B.phase=pha

; Determine the range of julian dates that will be used
minjdV=min(GMOri_V_Jan19(*,0)),format='(d12.4)'
maxjdV=max(GMOri_V_Oct18(*,0)),format='(d12.4)'

; Set up data structure for V filter

; Number of observations in the V filter will be how many elements each
; part of the data structure must have
nobsV=n_elements(GMOri_V_Jan19(*,0))+n_elements(GMOri_V_Jan20(*,0)) +$
           n_elements(GMOri_V_Jan21(*,0)) +$
           n_elements(GMOri_V_Oct16(*,0)) +$
           n_elements(GMOri_V_Oct17(*,0))+n_elements(GMOri_V_Oct18(*,0))
; Define the data structure
GMOri_final_V=phase:dblarr(nobsV),jd:dblarr(nobsV),realmags:dblarr(nobsV)
; JD inserted
GMOri_final_V.jd=[GMOri_V_Jan19(*,0), GMOri_V_Jan20(*,0), GMOri_V_Jan21(*,0),$
GMOri_V.Oct16(*,0), GMOri_V.Oct17(*,0), GMOri_V.Oct18(*,0)]
; Magnitudes inserted
GMOri_final_V.realmags=[GMOri_V Jan19(*,1), GMOri_V Jan20(*,1), GMOri_V Jan21(*,1),$
GMOri_V.Oct16(*,1), GMOri_V.Oct17(*,1), GMOri_V.Oct18(*,1)]
; Calculate phase
period=0.3905135
specJD=8580.836
pha=((GMOri_final_V.jd-specJD) mod period)/period
GMOri_final_V.phase=pha

; Determine the range of julian dates that will be used
minjdR=min(GMOri_R Jan19(*,0)), format='(d12.4)'
maxjdR=max(GMOri_R Oct18(*,0)), format='(d12.4)'

; Set up data structure for R filter

; Number of observations in the R filter will be how many elements each
; part of the data structure must have
nobsR=n_elements(GMOri_R Jan19(*,0))+n_elements(GMOri_R Jan20(*,0))+n_elements(GMOri_R Jan21(*,0))+n_elements(GMOri_R Oct16(*,0))+n_elements(GMOri_R Oct17(*,0))+n_elements(GMOri_R Oct18(*,0))
; Define the data structure
GMOri_final_R=phase:dblarr(nobsR), jd:dblarr(nobsR), realmags:dblarr(nobsR)
; JD inserted
GMOri_final_R.jd=[GMOri_R Jan19(*,0), GMOri_R Jan20(*,0), GMOri_R Jan21(*,0),$
GMOri_R.Oct16(*,0), GMOri_R.Oct17(*,0), GMOri_R.Oct18(*,0)]
; Magnitudes inserted
GMOri_final_R.realmags=[GMOri_R Jan19(*,1), GMOri_R Jan20(*,1), GMOri_R Jan21(*,1),$
GMOri_R.Oct16(*,1), GMOri_R.Oct17(*,1), GMOri_R.Oct18(*,1)]
; Calculate phase
period=0.3905135
specJD=8580.836
pha=((GMOri_final_R.jd-specJD) mod period)/period
GMOri_final_R.phase=pha
end
F Useful Websites for Astronomical Research

Several websites exist that are useful for astronomical research. This section gives some of the more useful websites for this purpose.

F.1 For Literature Research

In order to find journal articles related to a given topic, the SAO/NASA Astrophysics Data System, is useful. It can be found at the following address: http://adsabs.harvard.edu/abstract_service.html. The website allows both general searches by topic, and searches for specific papers.

In order to find other observational data, use SIMBAD.

F.2 For Weather and Observing

Before observing, a finder chart of the field is needed. It allows the observer to determine whether the image taken by the telescope matches the field in which the object of interest is located. The finder chart can be obtained using the Space Telescope Science Institute’s Digitized Sky Survey at http://archive.stsci.edu/cgi-bin/dss_form.

To determine the viability of an object for a given observing time and location, an hourly airmass table is used. One such table is available at http://www.briancasey.org/artifacts/astro/airmass.cgi. When using this website, be sure to change the epoch before searching. When determining the airmass at NURO, use Kitt Peak as the location. When determining the airmass in Carlisle, use the Black Moshannon Observatory at Penn State.

For weather checking, use the Clear Sky Clock. http://cleardarksky.com/c/BrittonObPAkey.html?1 This is specialized for use at the Britton. A second chart can be accessed for the NURO telescope at Anderson Mesa Station. http://cleardarksky.com/c/AdrMsStAZkey.html?1 Also helpful for weather is from the University of Illinois. http://ww2010.atmos.uiuc.edu/(Gh)/wx/satellite.rxml The continental United States Infrared Imagery shows clouds.
References


