FRI Astronomy Lab #2

Goal: In this lab you will learn about the observations we do and how we reduce the data from raw image form. You will also become familiar with some of the software that we will be using throughout this course, and will extend your proficiency with Mathematica.

1. Before you can begin, you need to configure your account a bit so that you can easily access the software the you will be using in this lab. By default, when you open a new terminal it looks for a file called ".bashrc" to learn what your terminal preferences are, including where on the computer different programs that you may want to access are located. The period before the name means that that is a hidden file. By default, `ls` commands do not list hidden files. Open a terminal and type `ls -a` to list *all* of the files in your home directory. Since you have a new account for the lab machines, you probably do not have a ".bashrc" file. Type `touch .bashrc` to create one. Use your favorite text editor to place the following line of instruction into your new file:

   ```
   . /etc/bashrc
   ```

   This essentially turns your local .bashrc file into a shortcut to an existing setup file that tells your terminal where software that we will be using in this lab is located. Save your file and type `. .bashrc` (notice the space between the periods) to load your new settings. These setting will be conveniently applied to all of your terminal sessions from now on.

2. Login to `rocky` using your personalized login account. If your EID is "smack" then your login name on `rocky` is `fri.smack`. Next, `cd` into the "smaug" directory.

3. Type `ls`. This is the place where all the data we take is stored. `cd` into the "sep08b" directory and then into "A1748" (this is the run name). The A stands for argos which is the camera used to take the data. Now type `pwd` (for print working directory).

4. Logout (control + d) and make a new directory in your home directory called A1748. Use a single `scp` command to copy only the files A1748.2600.fits through A1748.2699.fits into that directory. (hint: lookup regular expressions – in particular the * symbol). Also copy the files "filter.py" and "a1748.dat" (You can use additional scp commands for this). If you haven't already, `cd` into the A1748 directory you created.

5. We will be using the software package *IRAF* in this lab (that stands for **I**mage **R**eduction and **A**nalysis **F**acility) to examine the sequence of raw image files that you just transferred over. *IRAF* is notorious for being quite user unfriendly, but you will become familiar with its quirks.

   For starters, *IRAF* doesn't play nice with the normal terminal window, so we need to open a different type of terminal. From your open terminal window in your A1748 directory, type `xgterm&` . A new terminal window of type "xgterm" will open (the ampersand tells the

terminal to run the given command in the background, leaving the first terminal available to execute more commands).

The following steps will all be completed in the new xgterm until otherwise specified.

6. You will need a display window open to view the images in, so type `ds9&` to open a data visualization application named from Star Trek.

7. In order to begin using *IRAF*, you must have a "login.cl" file in your working directory. This file keeps track of your *IRAF* preferences and history. To create it, type `mkiraf` and inform the prompt that you are using `xgterm`. If you type `ls` you should see your "login.cl" file and a new directory called "uparm".

8. Begin *IRAF* by typing `cl` into the xgterm window. You will now be able to manipulate and execute *IRAF* "tasks," which are commands built for data reduction and analysis. You will become familiar with many tasks in future labs, but will focus on one this week called "imexamine."

   Each *IRAF* task requires a set of input parameters to be defined. Since you are using a freshly made login.cl file, the current parameters should be set to default. Type `lpar imexamine` to list the parameters of the task imexamine (you can instead type "imexam" for short). To learn more about what these parameters mean, type `help imexam`

   Parameters remember and store the last values they were set to under the same login.cl file, and these values will be reused behind the scenes whenever the parameters are not explicitly defined in an *IRAF* command. If things are not going as expected, you may wish to restore a task to its default parameters by typing `unlearn [task name]`

9. Before we get started analyzing this whole set of images, let's get a sense for what one in the series looks like. Type `display A1748.2600.fits 1` to load the first image file into frame 1 of the *ds9* window. You should see the star field. The "whiteness" of the pixels represents the relative amount of light that was measured at those pixel locations. This type of observation is useful for a process called "relative photometry" wherein the brightness of a star of interest is compared to the brightness of a (hopefully) stable comparison star to determine how much the brightness of the target may be changing. Can you think of why we must compare two stars in the same image rather than just keep track of the apparent brightness of our target star alone?

   These "fits" images contain more information than just the raw images. They also carry descriptions of the objects and field being observed, as well as details of the observing instruments, facilities, and settings. Type `imheader A1748.2600.fits longheader=yes` to view this information for the first image. Make note of the coordinates in the image where the target and comparison stars are located. Identify these stars in the *ds9* window (hint: if you mouse over the ds9 window the coordinates of the mouse position show up, as well as the photon count for that pixel). It is important that you remember which stars these are

through the rest of the lab, so sketch the star field in your lab notebook, circle the target and comparison star, and write down their coordinates.

10. Rather than run imexamine separately on your images one by one, you can save a lot of time by defining and sending to imexamine a list of images that you wish to examine. Create a list of the image (fits) files in your working directory by typing `ls *.fits > list.txt` This sends the output of `ls *.fits` to the file "list.txt". Check the contents of this file by typing `less list.txt`

11. You are now ready to run imexamine! Do this by typing:
    `imexam @list.txt 1 logfile=target.log`

    This executes the imexamine task with three parameters defined explicitly. The "at" sign before the name of the list of image filenames tells the task that we are giving it a list rather than a single image for the input. The "1" tells *IRAF* to load the images in the first frame of the *ds9* window through the required "frame" parameter. The optional "logfile" parameter is set to the name of a file that imexamine will output measurement information to as we explore the images. Use the name "target.log" for this call since you will be measuring the brightness of the target first, before separately measuring the brightness of the comparison star in step 14.

12. In order to use imexamine, you need to have your mouse focus on the *ds9* window. Click in the *ds9* window to focus on it. Your cursor should be an alternating black dot and ring over the displayed image. When your cursor looks like this, imexamine allows you to explore the displayed image in numerous ways using the mouse and keyboard together. Type a question mark to print your options to the xgterm window. Press "q" in the xgterm window when you're done reading the command options and refocus on the *ds9* window.

13. Explore some of the options on the displayed image. Put your cursor over different stars and get a feel for how their measured starlight is distributed on the detector using the "e" (contour), "s" (surface), and "r" (radial) plot key commands (Note: After the first of these, a new plot window will open and you will need to refocus your cursor on the *ds9* window. Be sure that the xgterm, *ds9* and plot windows are all fully visible throughout this exercise). The "r" command is particularly useful for gaining a quantitative understanding of a stellar image. While most of the values provided are more technical than we currently have need for, the third number from the left on the bottom of the plot window gives the total number of light photons detected from the star. This is the sum of photon counts from pixels that the stellar image appears on. You will be recording this value for the target and comparison stars for the entire sequence of images.

14. To record the measured values from the "r" command to the log file that we defined in our call to imexamine, we must tell toggle the "write" option to *on* by typing "w". Since our logfile "target.log" is only meant to contain measurements for the target star, be sure not to

run an "r" command on any other stars while the logfile is open. You can always press "w" again to close and later reopen the logfile again (as confirmed in the xgterm window).

With the log file open, press "r" over the target star in the first image. The information is now saved in the logfile to be plotted later. Proceed to the next image file with the "n" (next) hotkey ("p" will take you back to the previous image). Press the "r" key over the target star in the second image, sending a plot of the target profile to the plot window and outputting the measured values to the logfile. Continue in this fashion until you reach the last file (the "File" field in the *ds9* window will read "A1748.2699.fits" – be careful: they will wrap and repeat). Pay attention to how the displayed image and the radial plot of the target profile change visually as you flip through the observations.

Press "q" to quit imexamine.

15. Run imexamine one more time, setting the logfile parameter to "comparison.log" this time. Record the primary comparison star brightness by opening the logfile and flipping through the images again, pressing the "r" key command over the *comparison* star this time. Again, be sure to glance at each image and plot as you pass through them.

16. Good work, Sport. You have completed a tedious version of aperture photometry. In the next lab you will learn to appreciate faster tools for performing more accurate photometry, but with more automation comes less transparency as to how the measurements are actually made. Hopefully this exercise has armed you with a better visual concept for what photometry accomplishes.

    As it was in life, *IRAF* is also persnickety in death. Be sure to always **log out** of IRAF by typing "lo" in the xgterm window rather than just closing the window. This will prevent future headaches.

    You can then close the xgterm and *ds9* windows.

17. Now that you have measured the target and comparison star brightnesses over time, you ought to visualize the results in the meaningful way. You can use Mathematica to do this.

    We have provided a tool to help you to convert the information contained in the logs output by the imexamine routine to a more manageable format. Use a `less` command to see what your .log files look like. In your A1748 folder type `./filter.py` to create new files "target.dat" and "comparison.dat" (Note: Your .log files will need to be named exactly "target.log" and "comparison.log" for this to work. Rename them with `mv` if necessary). Use `less` to look at how the data is formatted in the new files. The first column of numbers lists the image frame number and the second column contains the stellar flux measurements. (Note that if you accidentally missed a frame while you were measuring, the flux value listed will be -1. If this is that case, you can either redo the measurements or come up with a way to ignore that data point when you are plotting your results).

Open Mathematica and import the data from each of these files as a "Table" using the `Import` command (You will need to specify the entire path to these files, or relative to your home directory using `~/`). Split the frame numbers from the brightness measurements using the `Part` command.

Make the following plots with `ListPlot` (Note: You may find that the options "Axes -> False, Frame -> True, Joined -> True" are more scientific looking):

- target brightness versus frame number
- comparison brightness versus frame number
- normalized target brightness (brightness divided by mean brightness value) versus frame number
- normalized comparison brightness versus frame number
- normalized target brightness divided by normalized comparison brightness

Write up short descriptions of the general features that you notice for each of these plots as comments in your Mathematica notebook. Why is it useful to consider the target brightness relative to comparison stars? What are the units on the y-axes of these plots?

18. Now it's time to look at the file "a1748.dat" that you transferred from Rocky. Use a `less` command to see its contents. This file contains the results of a more precise extraction of the light curve for this object over the complete set of images captures. The first four columns contain the time (in seconds), target brightness, comparison brightness, and second comparison star brightness. You won't need to worry about the other columns.

    Import these data and separate the columns as you did in the last step. Plot each star's normalized brightness against time. Plot the normalized target brightness divided each of the normalized comparison star brightnesses against time, and also one of the normalized comparison star's brightness divided by the other's. **Answer** the following questions as comments in your Mathematica notebook: *What similarities are present in any of these plots? What differences? Why?*

19. Please save your Mathematica notebook and submit it online via Canvas by Friday's lecture.

20. Bask in the glory of a job well done...