Observing a Variable Star – A Recipe for Photometry
COSMOS Astronomy Project, Summer 2004

Kathy Cooksey, UC Santa Cruz

Note: this is a recipe-in-progress, complete as possible but perhaps not comprehenseive. Feedback is welcome.

1 Starting IRAF and Accessing Data

Open IRAF

`monster>` xiraf

Change **d**irectory to the data directory from the IRAF directory (just a note: the seemingly random bold-facing of characters is my attempt to show you why some commands are what they are):

`cl>` cd ../data/szher/jul0804

Start IRAF photometry package

Access **N**ational **O**ptical **A**stronomy **O**bservatory utilities:

`cl>` noao

Access digital (i.e. CCD) photometry utilities:

`no>` digiphot

Access aperture photometry utilities:

`di>` apphot

To exit any of these packages but not from IRAF completely, type 'bye'. To exit IRAF, type 'lo' (**l**og **o**ut).

Configure IRAF to recognize Nickel extensions

`ap>` show imextn

Copy the extensions listed and insert 'ccd' as shown below (I can show you how to cut and paste easily on the computer):

`ap>` reset imextn="oif:imh fxf:fits,fit,**ccd** plf:pl qpf:qp stf:hhh,??h"

`ap>` reset imtype="fits"

`ap>` flpr

2 Renaming, Flat Fielding, & Registering Images

Making a simple image **l**i**s**t

`monster>` ls *.ccd > inname

To make an exact **c**o**p**y:

`monster>` cp inname outname

Editing with *Emacs*
`monster>` emacs -nw outname
In *Emacs* type:
Esc Shift %
then you can enter the expression which you want to replace '.ccd' with the new expression '.fits'.

To replace one occurance, hit *Enter*; to replace all occurances, hit '!'. If you want to undo something in *Emacs*, type:
Control Shift _

Changing names
`ap>` imcopy @inname @outname
Making a complicated image list
Using information in the image **h**eaders, **select** images based on certain keywords:
`ap>` hselect *.fits $I "OBJECT = 'bias"' > bias
`ap>` hselect *.fits $I "OBJECT = 'B flat' && FILTER = 1893" > b_flats
`ap>` hselect *.fits $I "OBJECT = 'V flat' && FILTER = 2373" > v_flats
`ap>` hselect *.fits $I "OBJECT = '*SZ Her*' && FILTER = 1893" > b_raw
`ap>` hselect *.fits $I "OBJECT = '*SZ Her*' && FILTER = 2373" > v_raw
Copy the *f*_raw lists to *f*_flatted (bad grammer, I know) and *f*_aligned, where the *f* represents the filter letter (b or v, in our case). Then edit these lists with *Emacs* so that the extension ('.fits') is 'f.fits' in the **f**lat-fielded list (*f*_flatted) and 'r.fits' in the aligned, or **r**egistered, list (*f*_aligned).

Flat Fielding
Set important parameters:
`ap>` epar imcombine

```
combine = median
reject = minmax
scale = mode
```

(I'm not going to explain what these parameters mean here, but feel free to ask if you like.) Save (**w**rite) and **q**uit by typing ':wq' or run immediately by typing ':go'. If you make a mistake enter, and want to exit without saving, type ':q!'. To run otherwise, type:
`ap>` imcombine @b_flats b_flat
Extract the *mean*, or average, value from the **im**age **stat**istics:
`ap>` imstat b_flat
Then normalize the flat by dividing by the mean, with **im**age **arith**metic:
`ap>` imarith b_flat / *mean*
and make the "Resultant image" b_flat.
Repeat above procedure for the V-band images.

Aligning Images
Registering images takes a bit of finesse, and varies from star field to star field, but for *SZ Her*, the following should work (99.9% guaranteed). We're interested in aligning the images so that the objects in which we're interested appear in the same location in each of our images. Then when we want to know how bright each object is in each image, IRAF knows exactly where to locate the objects.
First, find a *reference* image, one that has a lot of counts in *SZ Her* as well as the comparison stars and has nice, circular stars. For ambiguously good reasons, find a good reference image

for each filter. Display one image by:
ap> display *image* 1
We need to determine the pixel coordinates of *SZ Her* and the comparison stars. You'll have a star chart, and we'll have discussed which objects to use as comparisons. To find the coordinates, type **im**age **exam**, after display the reference image:
ap> imexam
Now the cursor will be over the image with a blinking circle. Place it over *SZ Her* and hit 'a' (**a**perture). The coordinates (and other information) will print to the IRAF window. Continue placing the cursor over the comparisons and typing 'a'. When you're done, just **q**uit ('q') and the cursor will return to normal. Open a new text file in *Emacs*:
monster> emacs -nw b_reg.coor
Enter the coordinates printed to the IRAF window into the text file, one object coordinate per line, either by cutting and pasting or by good, old-fashioned typing. Now prepare the IRAF utility for our needs:
ap> epar imalign

```
boxsize = 15
bigbox = 175
lower = 100
shiftim = yes
interp = linear
boundary = nearest
trimima = yes
```

Write and **q**uit (don't ':go' this). We want to save the stuff that IRAF will print to the screen when we run *imalign*, so run it as follows:
ap> imalign @b_flatted *reference* b_reg.coor @b_aligned > b_imalign.txt
Open up the *f*_imalign.txt file:
monster> more b_imalign.txt
You can move around by hitting the *spacebar* to move down and 'u' to move up. Go to the very end of the file and note the line where it says something like:
#Trim_Section = [10:1024,11:1024]
We're interested in the numbers before the colons (10 and 11 in this case). In this example, 10 is the number of pixels cut off in the x-direction and 11 in the y-direction. We need to adjust our *f*_reg.coor file to account for this so we can use it again:
monster> awk '{print $1-10,$2-11}' b_reg.coor > b_reg.coor2
Do the same for the V-band images.

3 Aperture Photometry and Creating a Lightcurve

QUICK **Phot**OMETRY
ap> epar qphot

```
image = @b_aligned
cbox = 8
annulus = 15
dannulus = 5
aperture = 8,10,12,15
coords = b_reg.coor2
```

```
output = b_mag
zmag = 19.79
exposur = EXPOSURE
filter = FILTER
obstime = TIME
```

Run this (':go' is good). The output *f*_mag is a massively unreadable file, so let's **t**e**x**t **dump** the fields in which we're interested.

`ap>` txdump b_mag > b_mag.txt

The `Fields to be extracted` will be 'image otime itime lid mag merr'. Do the same for the V-band images, and photometry's done.

Creating the Lightcurve

Generating a plot of how *SZ Her*'s brightness varies over time (i.e. lightcurve) is a bit more involved. We now leave the world of IRAF.

We need to gather the needed file with the other data which has been collected for this project (and make sure the names reflect the date of observation):

`monster>` cd /home/kcooksey/data/szher/jul0804

`monster>` cp b_mag.txt ../b_mag.070804

`monster>` cd ..

We need to order the magnitude files according to the time of observation:

`monster>` sort -n -k 2.1,2.4 -k 2.5 b_mag.070804 > b_magsorted.070804

Now generate the data file used to generate the lightcurve.

`monster>` awk -f rearr.awk -v napertures=4 -v useaperture=4 b_magsorted.070804 > b_lightcurve070804.dat

This program is tricky to explain. It essentially accounts for the difference in brightness between *SZ Her* and the comparisons to show how *SZ Her* varies, regardless of observing artifacts (e.g. atmosphere). This is the nature of differential photometry.

Once again, do the same for the V-band images.

We'll use an *I*nteractive *D*ata *L*anguage *pro*cedure (i.e. program) to generate the lightcurve.

`monster>` idl

`idl>` .r lightcurve

`idl>` lightcurve,'v_lightcurve070804.dat','b_lightcurve070804.dat'

This program creates three plots: v_lightcurve070804.eps, b_lightcurve070804.eps, and bv_lightcurve.eps.

This last plot you should immediately rename (by **mov**ing the contents to another filename):

`monster>` mv bv_lightcurve.eps bv_lightcurve070804.eps

View these files and see all your hard work come to fruition:

`monster>` gv bv_lightcurve070804.eps &

Phase Diagram

A phase diagram is another version of a lightcurve, whereby many obseravtions are folded together. It readily displays the shape of the lightcurve, and in generating the phase diagram, we learn about its period.

We have yet another program to generate the data for the phase diagram:

`monster>` awk -f fold.awk -v jd=2453195.5 -v refmag=0 -v hc=0.00296 -v E=916.05 b_lightcurve070804.dat > b_phase070804.dat

Now append, or con**cat**enate, this data onto that from previous observations:

`monster>` cat b_phase.dat b_phase070804.dat > b_phase.dat

Once again (again), do the same for the V-band images. Open up IDL again and generate the phase diagrams:

```
idl> .r phase
idl> phase,'v_phase.dat','b_phase.dat'
```

View the resulting plots (v_phase.eps, b_phase.eps, and bv_phase.eps) with **G**host**v**iew, as you did with the lightcurve.