Differential Photometry with IRAF

Once you have fully reduced your images for the night, you will need to measure the brightness of your target compared to some reference stars in the field to search for flux variations.

1. imexam

Investigate the stars in the image with your target and measure the average seeing of each image

First, change a few settings:
epar rimexam (yes, that is an “r” in front of “imexam”)

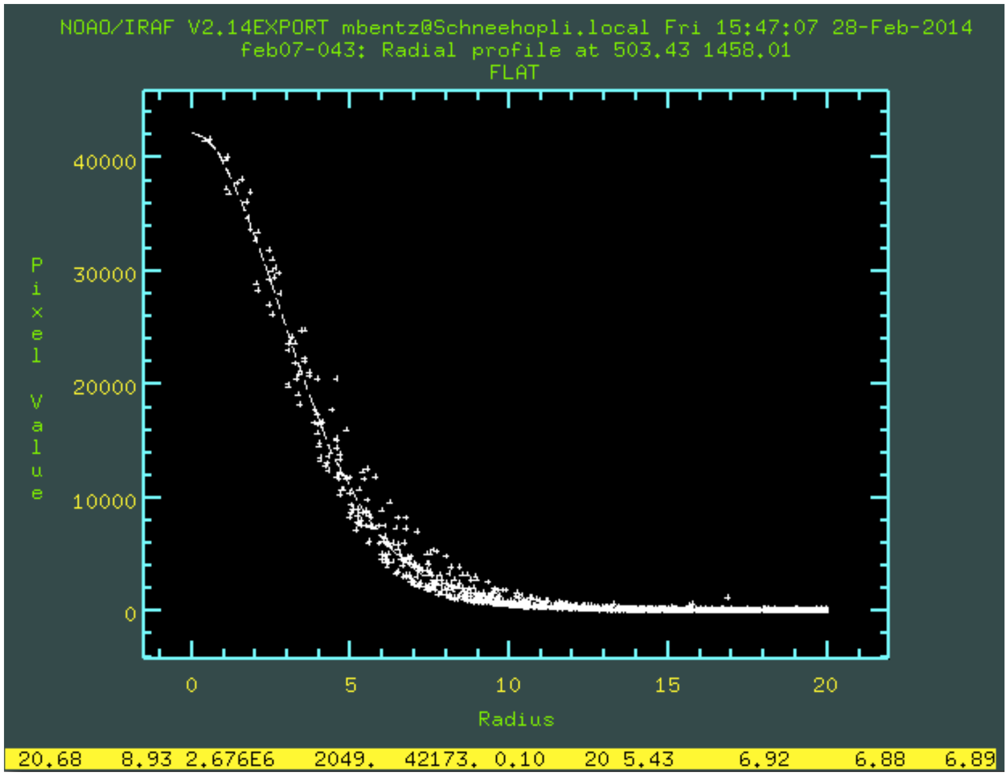
```
      (banner = yes)            Standard banner
       (title = "")             Title
      (xlabel = "Radius")       X-axis label
      (ylabel = "Pixel Value")  Y-axis label
     (fitplot = yes)            Overplot profile fit?
     (fittype = "moffat")       Profile type to fit
      (center = yes)            Center object in aperture?
  (background = yes)            Fit and subtract background?
      (radius = 5.)             Object radius
      (buffer = 10.)             Background buffer width
       (width = 5.)             Background width
  (iterations = 3)              Number of radius adjustment iterations
      (xorder = 0)              Background x order
      (yorder = 0)              Background y order
     (magzero = 25.)            Magnitude zero point
        (beta = INDEF)          Moffat beta parameter
       (rplot = 20.)            Plotting radius
```
... don't need to change other parameters ...

Make sure your reduced image is displayed (bias-subtracted, dark-subtracted, and flat-fielded), then type **imexam.** Your cursor will turn into a blinking annulus. Hover the cursor over a star and press the “r” button on the keyboard (don’t click the mouse). The radial plot for that star will be generated. Remember that you may need to click on the top of the ds9 window to make that window active first. To quit out of imexam, type “q” while hovering over the image in an active ds9 window.

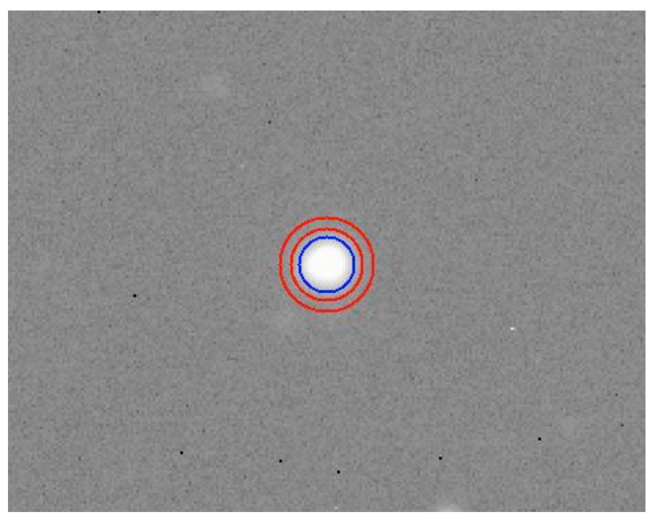
Investigate the radial plots for several stars and choose 3 stars in the image that are fairly bright but are not saturated. Be wary of any stars that get over 60000 counts in their centers, they could be saturated in the central one or two pixels. Remember that the detector saturates at 65,000 counts and you already subtracted off some counts for the bias and for the dark current.

These same 3 stars will be used in all your images, so you should check them in the other filters as well to be sure they are not saturated there either. You also want to be sure that they do not fall off the edge of the detector in any of your images after slewing and dithering. Be sure you can remember which stars you chose. Mark them and take a screenshot, or whatever you need to do.

Record the full-width at half-maximum (FWHM) value that prints at the very bottom right of the screen (the very last number). Do this for all 3 reference stars you've chosen. The median of these measurements is the seeing for that image and should be recorded in your logsheet. Note that the FWHM from IRAF is in pixels, so you need to convert that value to arcseconds using the plate scale of the detector (see the observing manual).



For each of your stars, note the radius at which most of the starlight is enclosed (here, approximately 10 pixels). Also note the range of radii outside of 10 pixels where there is no information from any sources, just sky flux (subtracted off in the example above). Your goals are to determine: 1) a **single** radius within which you would collect most of the light from all of your reference stars in **all** the images for that field, but to minimize the noise, you want to keep it as small as possible; and 2) a range of radii outside of the object aperture that is empty of everything except sky flux (for example, 13-17 pixels in the example above).



In practice, we will measure the brightness of our sources within your chosen aperture (10 pixels in this example), and we will put down two additional apertures to create an annulus or ring around this (as above, 13 pix and 17 pix in radius). Inside that annulus or ring, we will determine the local sky brightness per pixel to subtract off the pixels in our object aperture. Visually, we will do something like the following:

blue circle = aperture to add up star light

red circles = ring of sky brightness to measure (per pixel) and subtract from star light pixels in blue circle

You need to determine the radius of the blue circle (your aperture radius) and the radius of the inner red circle and the width of the annulus (difference between the radii of the two red circles).

2. phot

To measure the brightness of the target and the 3 reference stars, we will use the package **phot** (stored in **noao.digiphot.apphot**)
Remember that by default, IRAF only loads a few packages when it starts up. You have to manually

load other packages when you want to use them (but you only have to load them once per session). To verify where a task lives, you can type

phelp taskname

where taskname in this case would be "phot", and read the very top middle line (in this case "noao.digiphot.apphot"). To load the packages in which phot lives, type

noao <enter>
digiphot <enter>
apphot <enter>

You can type "?" to see the list of tasks available in the package you most recently loaded, and for a short description of each, type "help".

You will need to edit the phot parameters before running it.

Phot parameters:

```
        image = ""                The input image(s)
      skyfile = ""                The input sky file(s)
      (coords = "")               The input coordinate files(s) (default: image.c
      (output = "default")        The output photometry file(s) (default: image.m
    (plotfile = "")               The output plots metacode file
    (datapars = "")               Data dependent parameters
  (centerpars = "")               Centering parameters
  (fitskypars = "")               Sky fitting parameters
    (photpars = "")               Photometry parameters
 (interactive = yes)              Interactive mode ?
    (radplots = yes)              Plot the radial profiles in interactive mode ?
   (icommands = "")               Image cursor: [x y wcs] key [cmd]
   (gcommands = "")               Graphics cursor: [x y wcs] key [cmd]
       (wcsin = )_.wcsin)         The input coordinate system (logical,tv,physica
      (wcsout = )_.wcsout)        The output coordinate system (logical,tv,physic
       (cache = )_.cache)         Cache the input image pixels in memory ?
      (verify = )_.verify)        Verify critical parameters in non-interactive m
      (update = )_.update)        Update critical parameters in non-interactive m
     (verbose = )_.verbose)       Print messages in non-interactive mode ?
    (graphics = )_.graphics)      Graphics device
     (display = )_.display)       Display device
        (mode = "ql")
```

Be sure "interactive" and "radplots" are both "yes". There are additional hidden parameters in **datapars, centerpars, fitskypars, and photpars**. To get to them, arrow down to that line and type ":e" (colon followed by "e").

datapars parameters:

```
       (scale = 1.)               Image scale in units per pixel
     (fwhmpsf = 4.)               FWHM of the PSF in scale units
    (emission = yes)              Features are positive ?
       (sigma = INDEF)            Standard deviation of background in counts
     (datamin = INDEF)            Minimum good data value
```

```
    (datamax = INDEF)          Maximum good data value
      (noise = "poisson")      Noise model
    (ccdread = " ")            CCD readout noise image header keyword
       (gain = " ")            CCD gain image header keyword
  (readnoise = 6.5)            CCD readout noise in electrons
      (epadu = 2.3)            Gain in electrons per count
   (exposure = "")             Exposure time image header keyword
    (airmass = "")             Airmass image header keyword
     (filter = "")             Filter image header keyword
    (obstime = "")             Time of observation image header keyword
      (itime = 1.)             Exposure time
   (xairmass = INDEF)          Airmass
    (ifilter = "INDEF")        Filter
      (otime = "INDEF")        Time of observation
       (mode = "ql")
```

datapars notes:
The keywords for “exposure” and “airmass” and “filter” should match what is printed in the headers of the images, and the values for “readnoise” and “epadu” (gain) are important here so be sure that you fill them in properly. The others aren't that important and can be left alone. Also, change the “fwhmpsf” value to be the same as your average seeing measurement (in pixels) for the average image on an average night. You can then leave it the same throughout. Type “:q” to exit the editor for datapars.

centerpars parameters:

```
 (calgorithm = "centroid")     Centering algorithm
       (cbox = 5.)             Centering box width in scale units
 (cthreshold = 0.)             Centering threshold in sigma above background
 (minsnratio = 1.)             Minimum signal-to-noise ratio for centering alg
   (cmaxiter = 10)             Maximum number of iterations for centering algo
   (maxshift = 10.)            Maximum center shift in scale units
      (clean = no)             Symmetry clean before centering ?
     (rclean = 1.)             Cleaning radius in scale units
      (rclip = 2.)             Clipping radius in scale units
     (kclean = 3.)             Rejection limit in sigma
   (mkcenter = no)             Mark the computed center on display ?
       (mode = "ql")
```

centerpars notes:
Make sure you edit “maxshift” to have a value of “10” so that you can be a bit sloppy about how well you center the mouse on an object when you mark its location. The other parameters are less important.

fitskypars parameters:

```
 (salgorithm = "median")       Sky fitting algorithm
    (annulus = 13.)            Inner radius of sky annulus in scale units
   (dannulus = 4.)             Width of sky annulus in scale units
   (skyvalue = 0.)             User sky value
   (smaxiter = 10)             Maximum number of sky fitting iterations
    (sloclip = 0.)             Lower clipping factor in percent
    (shiclip = 0.)             Upper clipping factor in percent
   (snreject = 50)             Maximum number of sky fitting rejection iterati
  (sloreject = 3.)             Lower K-sigma rejection limit in sky sigma
```

```
   (shireject = 3.)              Upper K-sigma rejection limit in sky sigma
       (khist = 3.)              Half width of histogram in sky sigma
     (binsize = 0.1)             Binsize of histogram in sky sigma
      (smooth = no)              Boxcar smooth the histogram
       (rgrow = 0.)              Region growing radius in scale units
       (mksky = no)              Mark sky annuli on the display
        (mode = "ql")
```

fitskypars notes:
Here, you need to set the region for determining the local sky background. “Annulus” should be the radius for the inner part of your sky ring (13 pixels in the previous example) and “dannulus” should be the width of the sky ring in pixels (17-13 = 4 pixels in the previous example). Set them to what YOU determined for your data.

photpars parameters:

```
   (weighting = "constant")      Photometric weighting scheme for wphot
   (apertures = "10")            List of aperture radii in scale units
        (zmag = 25.)             Zero point of magnitude scale
     (mkapert = yes)             Draw apertures on the display
        (mode = "ql")
```

photpars notes:
Here is where you set the size of the aperture within which you will add up all the flux from the star under the “apertures” keyword (10 pixels in the previous example).

Once all these parameters are set, run **phot** on a single image (bias-subtracted, dark-subtracted, and flat-fielded).

Always in the same order, place the cursor over the center of your target and your reference stars (you don't need to click the mouse button) and press the space bar. You should see a window pop up that shows a radial plot and your aperture radius marked with a vertical line, as well as the sky annulus inner and outer radii.

When you have finished measuring sources in an image, press “q” twice to quit **phot**. You should have a file now that has the name of the image file and ends with “mag.1”. If you run **phot** on the same image again, you will get a file called “mag.2”, and so on.

Run **phot** on the rest of your images of your target, remembering to always measure the target and reference stars in the same order.

The mag files can be opened with any text editor (e.g., vi, emacs, pico) and look like this:

```
#K IRAF       = NOAO/IRAFV2.12.2a-EXPOR version    %-23s
#K USER       = bentz                   name       %-23s
#K HOST       = saha.chara.gsu.edu      computer   %-23s
#K DATE       = 2010-08-30              yyy-mm-dd  %-23s
#K TIME       = 14:41:01                hh:mm:ss   %-23s
#K PACKAGE    = apphot                  name       %-23s
#K TASK       = phot                    name       %-23s
#
#K SCALE      = 1.                      units      %-23.7g
```

```
#K FWHMPSF    = 4.5                     scaleunit  %-23.7g
#K EMISSION   = yes                     switch     %-23b
#K DATAMIN    = INDEF                   counts     %-23.7g
#K DATAMAX    = INDEF                   counts     %-23.7g
#K EXPOSURE   = ""                      keyword    %-23s
#K AIRMASS    = ""                      keyword    %-23s
#K FILTER     = ""                      keyword    %-23s
#K OBSTIME    = ""                      keyword    %-23s
#
#K NOISE      = poisson                 model      %-23s
#K SIGMA      = INDEF                   counts     %-23.7g
#K GAIN       = ""                      keyword    %-23s
#K EPADU      = 1.                      e-/adu     %-23.7g
#K CCDREAD    = ""                      keyword    %-23s
#K READNOISE  = 0.                      e-         %-23.7g
#
#K CALGORITHM = centroid                algorithm  %-23s
#K CBOXWIDTH  = 5.                      scaleunit  %-23.7g
#K CTHRESHOLD = 0.                      sigma      %-23.7g
#K MINSNRATIO = 1.                      number     %-23.7g
#K CMAXITER   = 10                      number     %-23d
#K MAXSHIFT   = 10.                     scaleunit  %-23.7g
#K CLEAN      = no                      scaleunit  %-23b
#K RCLEAN     = 1.                      scaleunit  %-23.7g
#K RCLIP      = 2.                      scaleunit  %-23.7g
#K KCLEAN     = 3.                      sigma      %-23.7g
#
#K SALGORITHM = median                  algorithm  %-23s
#K ANNULUS    = 15.                     scaleunit  %-23.7g
#K DANNULUS   = 10.                     scaleunit  %-23.7g
#K SKYVALUE   = 0.                      counts     %-23.7g
#K KHIST      = 3.                      sigma      %-23.7g
#K BINSIZE    = 0.1                     sigma      %-23.7g
#K SMOOTH     = no                      switch     %-23b
#K SMAXITER   = 10                      number     %-23d
#K SLOCLIP    = 0.                      percent    %-23.7g
#K SHICLIP    = 0.                      percent    %-23.7g
#K SNREJECT   = 50                      number     %-23d
#K SLOREJECT  = 3.                      sigma      %-23.7g
#K SHIREJECT  = 3.                      sigma      %-23.7g
#K RGROW      = 0.                      scaleunit  %-23.7g
#
#K WEIGHTING  = constant                model      %-23s
#K APERTURES  = 5,7,9,11                scaleunit  %-23s
#K ZMAG       = 25.                     zeropoint  %-23.7g
#
#N IMAGE               XINIT     YINIT     ID    COORDS                 LID    \
#U imagename           pixels    pixels    ##    filename               ##     \
#F %-23s               %-10.3f   %-10.3f   %-6d  %-23s                  %-6d
#
#N XCENTER    YCENTER    XSHIFT  YSHIFT  XERR    YERR            CIER CERROR   \
#U pixels     pixels     pixels  pixels  pixels  pixels          ##   cerrors  \
#F %-14.3f    %-11.3f    %-8.3f  %-8.3f  %-8.3f  %-15.3f         %-5d %-9s
#
#N MSKY           STDEV          SSKEW          NSKY   NSREJ     SIER SERROR   \
#U counts         counts         counts         npix   npix      ##   serrors  \
#F %-18.7g        %-15.7g        %-15.7g        %-7d   %-9d      %-5d %-9s
#
```

```
#N ITIME          XAIRMASS       IFILTER                 OTIME                   \
#U timeunit       number         name                    timeunit                \
#F %-18.7g        %-15.7g        %-23s                   %-23s
#
#N RAPERT   SUM           AREA        FLUX          MAG     MERR   PIER PERROOR   \
#U scale    counts        pixels      counts        mag     mag    ##   perrors  \
#F %-12.2f  %-14.7g       %-11.7g     %-14.7g       %-7.3f %-6.3f %-5d %-9s
#
fdb_feb07-043.fit      459.000   624.000   1     nullfile            0      \
   462.001    624.538    3.001   0.538   0.008   0.006         0    NoError   \
   332.4668       10.74996       -3.45535       1247   9        0    NoError   \
   1.             INDEF          INDEF                   INDEF                   \
   10.00     110663.       78.67984   84504.59      12.683 0.004 0    NoError  *\
...
```

The header with the “#” signs preceding the lines helps you to read the mag files, and all the calculated numbers are at the end in the lines with no “#” signs. In this case, the bold parts of the file above are the key and the information, respectively.

First line of information (without the # signs) is the Image, the X and Y values of where you had placed the cursor, and some other stuff we don't care about on that line.

Second line is the X and Y values for the center of the star (or your target), the X and Y shift of where the center was relative to where you placed the cursor, and the uncertainties in the location of the center of the brightness.

Third line is the sky brightness (per pixel) and standard deviation, the skewness of the sky brightness distribution, the number of pixels in the “sky” annulus and some other stuff.

Fourth line is the value of various keywords from the header of the image, if they were found: exposure time, airmass, filter, time of observation.

Fifth line:
Aperture radius (in pixels), total sum of counts within the aperture, number of pixels in the aperture, total counts minus the sky value in the aperture, and the **magnitude** and **uncertainty** of the source (where mag = -2.5*log(flux)+25, and 25 is a random zeropoint that we will leave alone for right now). If you ever see one of these lines ending with “Error” instead of “NoError”, you need to go back and redo the measurements for that image. An error is usually caused by not having the image cursor placed close enough to the actual center of the source you tried to measure, so the algorithm is having a hard time finding the center of the star (or whatever the source might be).

It is useful to check the .mag files for any problems before moving on to the next images, so that you can quickly correct any problems that might have resulted from not choosing the center of one of your targets. Also, it is helpful for the rest of the analysis to delete any “.mag” file with errors in it so that you will only have one “.mag” file for each image, and each one is known to be good.

When you have finished measuring the magnitudes of your target and reference stars in all the images, you will then organize your measurements.

3. fields

Fields is a very useful IRAF task for pulling information out of text files without having to use copy and paste. Suppose I have a file with information that looks like this:

```
#Blah Blah Blah
#
#N RAPERT   SUM           AREA       FLUX           MAG    MERR   PIER PERROR   \
#U scale    counts        pixels     counts         mag    mag    ##   perrors  \
#F %-12.2f  %-14.7g       %-11.7g    %-14.7g        %-7.3f %-6.3f %-5d %-9s
#
rccd100820.0082.fits   550.735   620.808   1     nullfile              0      \
   548.817    622.270    -1.918  1.462   0.006   0.006         0    NoError   \
   477.7075       40.66297       30.80108       806    137      0    NoError   \
   1.             INDEF          INDEF                  INDEF                   \
   7.00     425485.6      154.1158   351863.4      11.134 0.002 0    NoError
rccd100820.0082.fits   462.576   622.503   2     nullfile              0      \
   462.066    624.541    -0.510  2.038   0.008   0.006         0    NoError   \
   336.0357       12.36176       7.067024       934    6        0    NoError   \
   1.             INDEF          INDEF                  INDEF                   \
   7.00     150549.9      153.7792   98874.6       12.512 0.004 0    NoError
rccd100820.0082.fits   681.278   581.815   3     nullfile              0      \
   682.257    583.507    0.979   1.692   0.019   0.015         0    NoError   \
   319.5727       10.02048       3.972704       940    3        0    NoError   \
   1.             INDEF          INDEF                  INDEF                   \
   7.00     65568.21      154.1938   16292.09      14.470 0.012 0    NoError
```

For example, I can pull out just the aperture ("7" in this case), the total flux, the sky flux, and the total-sky with the following parameters in **fields**:

```
        files = "rccd100820.0082.fits.mag.1" Files from which to extract fields
       fields = "1,2,3,4"       Fields to extract
       (lines = "80-200x5")     Lines from which to extract fields
(quit_if_miss = no)             Quit on missing field?
(print_file_n = no)             Print file names if multiple files?
        (mode = "ql")
```

In the parameters, the "fields" are the entries across a single line (the first, second, third, and fourth entries, in this case) and the "lines" are the specific lines in the file (every fifth line from line 80 to line 200 in this case). I use "emacs" as my text editor when I want to find out what line a particular entry is sitting on. The emacs window has a very useful line counter at the bottom center so you can easily figure out what line your useful entries start on.

If I run **fields** on the above file with the parameters I listed, I'll get the following printed out in the IRAF terminal:

```
Files from which to extract fields (rccd100820.0082.fits.mag.1):
Fields to extract (1,2,3,4):
7.00   425485.6   154.1158   351863.4
7.00   150549.9   153.7792   98874.6
7.00   65568.21   154.1938   16292.09
```

What you actually want to pull out is the mag and magerr for each object. In this case, we want fields "5,6" for all the objects.

You can save the output from **fields** by redirecting it with ">" to a new file to save. It is always best to check the output by letting it print to the screen first before running **fields** again and redirecting the output to a file. When you are ready to save the output, then give the following command:

fields > file.dat

The information that would normally print on the terminal will instead be saved to the file "file.dat".

If you have several files, and each file has reference star 1 with the 7 pixel aperture on line 80, you could edit the parameters so that you get one file with all the measurements for just reference star 1:

```
       files = "*.mag.1" Files from which to extract fields
       fields = "5,6"       Fields to extract
       (lines = "80")     Lines from which to extract fields
(quit_if_miss = no)             Quit on missing field?
(print_file_n = no)             Print file names if multiple files?
        (mode = "ql")
```

You can then edit the lines field to choose the next star or your target and pull out all of its MAG and MERR measurements at once and save them to a file. The new files you have created with all the measurements for a single object in one file can then easily be copied and pasted into a spreadsheet (or whatever you like) for further calculations.

An important piece of information that you will need is the heliocentric Julian date at the midpoint of each observation. If you are going to use a spreadsheet, you would then need to include a column for HJD. To quickly get the HJD values, you can use **hselect** like:

hselect *.fits $I,jd-helio yes

"*.fits" will select the HJD from every file that ends with "*.fits", which is not necessarily what you want. The "$I" command (dollar sign and capital i) tells hselect to echo the filename along with listing the HJD value from the header. Perhaps the best way is to build a list of all the images for which you made photometry measurements (images.lst) and then run hselect with the input list of images:

hselect @images.lst $I,jd-helio yes

Remember, the "@" symbol tells IRAF that the input is a *list* of image names (not an image name itself). When you're sure you've got everything right, you can redirect the output to a file to save:

hselect @images.lst $I,jd-helio yes > hjd.dat

This will print a list of file names and HJD values to the file hjd.dat. You can then take these values and put them in the spreadsheet with your MAG and MERR values for each star. Remember that all the reference stars and your target in a single image will have the same HJD value.

4. Magnitude calculations

We will be doing what is called "differential photometry", so we want to take differences between the measured values (magnitudes) for the reference stars and your target. The reference stars should not be changing brightness over time, but they will appear to because of clouds and atmospheric transparency, seeing conditions, moon phase, and a whole host of other things that are very difficult to overcome. The basic idea is to measure the brightness of the reference stars in each image, determine the brightness difference for each image compared to a reference, and then use the measured brightness difference for these non-varying sources to correct our measurements of the brightness of your target for the effects of the atmosphere and clouds and seeing, etc. What's left should just be any real changes in the brightness of your target.

To start, we're going to just determine the difference between each reference star's measured magnitude and its magnitude IN THE SAME FILTER in an image near HA=0 or airmass=1. So you'll likely have a spreadsheet with the following columns:

HJD TARGmag TARGmerr S1mag S1merr S2mag S2merr S3mag S3merr ...

and the difference between the measured mag of a star and its reference magnitude in an image near the meridian (minimum airmass, HA=0).

... S1magdiff S1mdifferr S2magdiff S2mdifferr S3magdiff S3mdifferr

Remember that when you find the uncertainty in a combined quantity (like $m_1 - m_2$), you need to treat the uncertainties properly using error propagation (refer back to lecture notes and homework).

Now we need to find the average mag difference for a single image according to the information we have from all 3 reference stars. To do this, we actually want to calculate the weighted average of the magnitude differences for the 3 stars for a single image. The weighted average will properly take into account our measurement uncertainties and how “trustworthy” each measurement is. The weighted average is defined as the following:

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i},$$

Basically, it's the sum of the weights times the values, divided by the sum of the weights. The weights in this case are the reciprocals of the square of the uncertainties:

$$w_i = \frac{1}{\sigma_i^2}.$$

so the weighted mean is

$$\bar{x} = \frac{\sum_{i=1}^{n}(x_i/{\sigma_i}^2)}{\sum_{i=1}^{n}(1/{\sigma_i}^2)},$$

The uncertainty on the weighted mean is similar to the calculation for the weighted mean in the equation just above. Just replace the entire numerator with “1” and take the square root of the whole thing to get the uncertainty on the weighted mean.

Once you have calculated the weighted mean of the magnitude difference in that image, you can correct the reference stars and your target for the magnitude difference. This will give you the reference star “true magnitudes” (which here we are taking as their magnitudes measured through the lowest airmass that night), and it will give you the corrected magnitude measurement for your target from that image.

If you have done your math correctly, the final magnitudes of the reference stars should always be about the same from one image to another and should always be close to the “true magnitudes” of the stars that you chose earlier. A small amount of scatter around these values is normal.

Repeat the above steps as necessary for each filter and for each target until you have calculated all your corrected magnitudes.

5. Plotting the light curves

We want to plot the corrected magnitudes for the reference stars and for your target against the HJD values. Use whatever plotting package you prefer (**6100 students**: excel is not acceptable, these should be publication quality plots).

Be sure to include appropriate axis labels and error bars showing the uncertainties on your corrected magnitudes. Plot your magnitude axis so that brighter values are near the top and fainter values are near the bottom. Label each plot so that it is clear what object is represented.

In the end, you should have a plot of brightness versus time for your target and each of your 3 reference stars in each filter you were assigned to study. The reference star light curves should be basically flat lines with basically the same magnitude value at all measurement points, while the target will hopefully show some interesting variations as a function of time.

It is good practice to plot your target and your reference stars on the same overall sheet of paper for easy comparison. It is also good practice to use the same y-axis range for your target and for your reference stars (+/- 1 mag? +/- 2 mag?) so that it is easy to see how much your target varied compared to the scatter in the corrected reference star magnitudes.

Final Thoughts:

For the purposes of this in-class exercise, we are only interested in relative flux variations, so it doesn’t matter if we know the “real” magnitudes of our reference stars or target. This is not always the case, however. The standard star observations that you obtained throughout the night could be used to solve the photometric transformation equations that would allow you to determine the “real” magnitude of each reference star and your target. A good guide for this already exists online --- *A User’s Guide to Stellar CCD Photometry with IRAF*, by Philip Massey and Lindsey E. Davis (the relevant text begins in Section 3.6).