Homework Assignment
Data Analysis II: High Energy
Due: 3:30 PM, Wednesday, November 10

1. In Lab 9, you used the "fimgstat" ftool to extract some information about various files. Record these values in the table below.

File name	Maximum value	Minimum value	Total	Mean
ngc4406_suzaku_8bin_img.fits	66	0 or 1	127712	8.64
exposure_map_img.fits	98198	0 or 0.0417	********	79481
flat_field_ds9smo15_img.fits	197	0 or 0.000442	********	89.0
flat_field_smo_trim_img.fits	197	0 or 21.2	********	132
flat_field_expmap_img.fits	98200	0 or 10550	********	66008
ngc4406_suzaku_8bin_vigcor_rate_trimg.fits	0.000680	0 or 1.15e-5	1.79	0.000121

2. In **ds9**, open the following images that you have constructed into six separate frames, and tile these. Print a hardcopy of the six-images-in-one, and hand this in with the assignment:

   ngc4406_suzaku_8bin_img.fits
   exposure_map_img.fits
   flat_field_img.fits
   flat_field_ds9smo15_img.fits
   flat_field_expmap_img.fits
   ngc4406_suzaku_8bin_vigcor_rate_trimg.fits

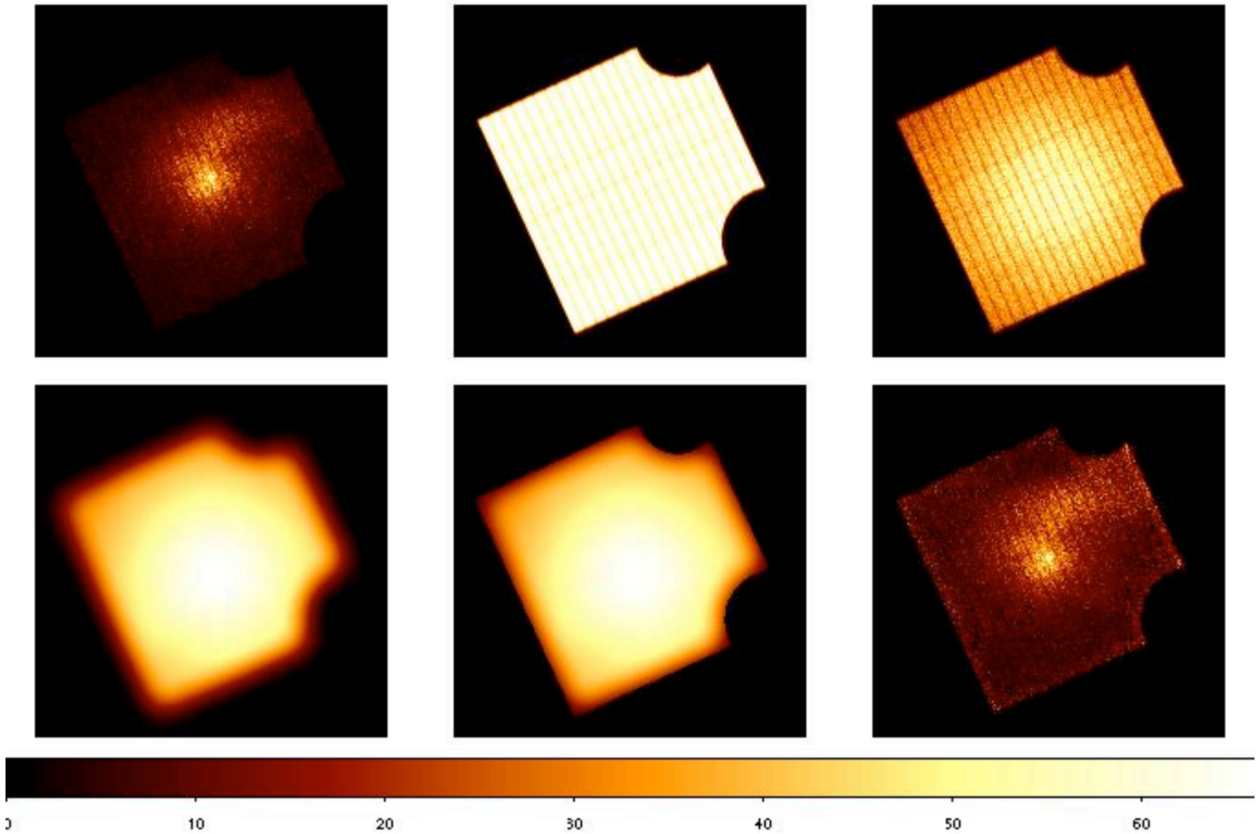
There will likely be artifacts at the edges of the last image, which would generally be excluded from any additional analysis.

3. Many of the manipulations done in the lab and homework can be done as easily with **IDL** (though, generally not mission specific tasks). Write an **IDL** procedure that reads the fits files *ngc4406_suzaku_8bin_img.fits*, *exposure_map_img.fits*, and *flat_field_ds9smo15_img.fits*, and produces the final vignetting-corrected count rate image. This can actually be done in just a few steps, as follows (though you are free to construct the program as you wish – as long as it works!). The Lecture 9 notes should be helpful here.

   a) Read the fits images and headers using readfits, assigning these arrays and strings to different variables, i.e. *my_image=readfits("something_img.fits",my_header).*
   b) Use the **IDL** *where* function to identify the indices where the exposure map image has values less than 1000. These are to be trimmed from the smoothed flat field image. Let's call that one-dimensional array "index_trim."

c) Create a trimmed flat field array ("flat-field_array_trim") that originally is a duplicate of the flat field, and set the values of entries corresponding to "index_trim" equal to 0.
d) Divide "flat-field_array_trim" by its maximum value to normalize it.
e) Get the exposure time (one way is to use the **IDL** command *sxpar* on *my_header*, with EXPOSURE as the argument), and multiply the normalized, trimmed, flat field image by this value.
f) Create an array, initialized at 0, to hold the final vignetting-corrected count rate image. Let's call this array "vigcor_cr_image." To avoid dividing by 0, find *where* "flat-field_array_trim" (or any of its rescaled descendants) is nonzero. Let's call this 1d array "nonzeros."
g) Fill the "nonzero" entries of "vigcor_cr_image" with the corresponding entries of the original binned image divided by these same "nonzero" entries in the normalized, trimmed, flat field image. The rest of the entries should remain at 0.
h) Use writefits to create the new file.
i) It will be helpful to "print" out max's, min's, and mean's of arrays as you go along to crosscheck with the entries in the table above.
j) Attach a copy of your procedure to this assignment.

```
PRO VigCor

eventimg = readfits( "ngc4406_suzaku_8bin_img.fits", evthdr)
exposeimg = readfits("exposure_map_img.fits",exphdr)
flatimg = readfits("flat_field_ds9smo15_img.fits", flathdr)

index_trim = where(exposeimg LE 1000.0)

flat_field_array_trim = flatimg
flat_field_array_trim(index_trim) = 0.0

flat_field_array_norm_trim = flat_field_array_trim / max(flat_field_array_trim)

exposetime = SXPAR(exphdr, 'EXPOSURE')

flat_field_array_exposuremap = flat_field_array_norm_trim * exposetime

vigcor_cr_image = dblarr(192,192)

nonzeros = where(flat_field_array_norm_trim GT 0.0)

vigcor_cr_image(nonzeros) = double(eventimg(nonzeros))/flat_field_array_exposuremap(nonzeros)

writefits, 'ngc4406_suzaku_8bin_vigcor_IDL.fits', vigcor_cr_image

end
```


0
10
20
30
40
50
60