ASTR 310: Computing in Astronomy
Final Project

Due online Saturday, May 11, 2019 by 11:59 pm

PLEASE READ INSTRUCTIONS CAREFULLY

For this assignment, your solutions should consist of a Jupyter notebook file. Have the notebook embed Matplotlib plots inline with the rest of the content. In the text or notebook file, clearly label the work for each problem using a separate markdown cell, and do each part in a separate cell. Clean up any missteps in the notebook and ensure that it runs from scratch.

Remember:

- **The final project may not be submitted late**, but you may upload draft files and replace them with a final version at any time up until the deadline.
- Final projects must be your own work and **must not be discussed** with your peers prior to submission.
- The instructors will answer clarifying questions as they would during an exam, but they **will not offer assistance** as with homework assignments. This is a summative assignment.
- You are encouraged to consult lecture/preflight slides and homework solutions from the course web site, as well as Python, NumPy/SciPy, and Astropy documentation. The instructors will answer clarifying questions about the slides and solutions. **You are not permitted to request help from online forums**, and if you are caught doing this it will be treated as an academic integrity infraction.

For this project you will write code to read a calibrated FITS image, apply some basic processing to it, fit a model to it, and make some plots. To begin, download the file `u2e64x01t_c0m.fits` from the course web site to the directory in which you will work. This is a Hubble Space Telescope WFPC2 image of the galaxy NGC 6212 with the F606W filter. Write code to perform the following tasks and generate the requested outputs.

1. Read the first image extension in the indicated FITS file. The image data are in photon counts/pixel, but you may use `unit='adu'` since we will not be relying on the Astropy unit interface here (apart from setting the cutout region size).

   Read the PHOTFLAM, PHOTZPT, and EXPTIME values from the image's FITS header. These are, respectively, the inverse sensitivity, the photometric zero point, and the exposure time of the image. Multiply the image data by PHOTFLAM/EXPTIME and divide by 0.0455**2 to convert the image values to erg/s/cm^2/Å/arcsec^2. (The pixel size of the Planetary Camera was 0.0455 arcsec.)

2. Write a function that converts an intensity I from erg/s/cm^2/Å/arcsec^2 to magnitudes/arcsec^2 by returning $-2.5 \log I + \text{PHOTZPT}$. However, do not apply it to the image data. We'll use it later.

3. After scaling the image data, create a cutout of the $0.5' \times 0.5'$ region centered at J2000 right ascension 16h43m23.111s, declination +39d48m22.829s. (This covers most of the Planetary Camera chip.)

4. Using the image's World Coordinate System (WCS), plot the cutout region as a color map with the origin at the bottom left. Use a logarithmic stretch with normalization range $(0, 7 \times 10^{-16})$ erg/s/cm^2/Å/arcsec^2. Use the "Greys" color map. Overlay the coordinate grid using solid yellow lines. Label the coordinate axes appropriately and give the plot an appropriate title. Note that you may find later questions easier if in this part you create a function that takes a 2D array, a WCS object, and a title string as input and produces the plot, then call this function with the cutout data, cutout WCS, and an appropriate title as arguments.

5. Remove foreground stars and cosmic rays from the **unstretched** cutout image in the following way. First write a "de-starring" function that takes as arguments a 2D image array I, a scalar blurring width σ, and a scalar threshold t, and returns a "de-starred" version of the image array (call it D). If we denote the ij-th pixel of the input image array as I_{ij}, the function should perform the following steps:

   $$B = \text{blur}(I, \sigma)$$

   $$M = I - B$$

   $$D_{ij} = \begin{cases} B_{ij} & \text{where } M_{ij} > tI_{ij} \\ I_{ij} & \text{otherwise} \end{cases}$$

   return D

   For the blurring step, use the `scipy.ndimage.gaussian_filter` routine.

   Once you have written your "de-starring" function, apply it to the **unstretched** cutout data with $\sigma = 3$ and $t = 0.25$, then apply it to the result with the same parameters, and then apply it to the result of the second application with the same parameters. Create an image of the "de-starred" cutout using the same stretch as applied in the previous question. Remember to use the cutout WCS to establish the projection, label your axes, and add an appropriate title.

6. Use the cutout region's WCS object to compute the right ascension and declination for each pixel in the image. Store the results in two 2D arrays, `ra` and `dec`, so that `ra[i,j]` and `dec[i,j]` refer to the right ascension and declination, respectively, of the ij pixel, with both measured in decimal degrees.

   Create a Boolean mask array that is `True` where the result of the previous problem is nonzero. Use it to mask the that result and the `ra` and `dec` arrays. Find the pixel with the maximum value and print its right ascension and declination along with the intensity at that pixel in magnitudes/arcsec^2 (using the magnitude function you wrote earlier).

7. In this problem you will fit a Sersic profile to the galaxy. The Sersic model for the surface brightness I as a function of radius r is

$$I(r) = I_0 e^{-(r/r_0)^{1/n}} ,$$

where I_0, r_0, and n are fitting parameters.

To begin, subtract the peak right ascension and declination computed in the previous problem from the masked `ra` and `dec` arrays, then flatten them and the masked pixel array (ie. reduce them to 1D arrays).

Our model must account for the parameters of the Sersic model and the galaxy's unknown position angle and inclination with respect to the line of sight, as well as the unknown background level. Define a fitting function that takes as independent variable the array index i and fitting parameters I_0, r_0, n, f, θ, and b, where the first three parameters are as defined above, f describes the apparent ellipticity caused by the tilt of the galaxy's disk with respect to the line of sight, θ is its position angle (angle of the apparent major axis of the ellipse with respect to the right ascension axis), and b is the background. The fitting function should first compute rotated sky coordinates given `ra[i]` and `dec[i]`, then compute a deprojected radius, then return the model intensity:

$$x = \mathtt{ra}[\mathtt{i}] \cos\theta - \mathtt{dec}[\mathtt{i}] \sin\theta$$

$$y = \mathtt{ra}[\mathtt{i}] \sin\theta + \mathtt{dec}[\mathtt{i}] \cos\theta$$

$$r = [x^2 + (y/f)^2]^{1/2}$$

$$I = I_0 e^{-(r/r_0)^{1/n}} + b$$

The data to be fit are the array indices for the independent variable and the masked pixel values for the dependent variable. Use the SciPy `curve_fit` routine to fit these data with your fitting function, choosing sensible initial guesses for the fitting parameters (inspect the shape and size of the galaxy in the image and look at the maximum pixel value).

For the uncertainties (`sigma` argument to `curve_fit`), use the square root of the number of counts in each pixel multiplied by the count-intensity conversion factor PHOTFLAM/EXPTIME/0.0455**2. Remember that the masked pixel values have already been converted from counts to intensity.

Once you have performed your fit, print the the best-fit values of the fitting parameters (with units as appropriate) and the reduced χ^2 value.

8. Create arrays containing the best-fit deprojected radius r and Sersic model intensity (converted to magnitudes/arcsec^2) at your masked right ascensions and declinations

and convert your masked pixel values to magnitudes/arcsec^2. Plot the pixel magnitudes and best-fit model magnitudes against r on a log-linear scatter plot using different colors, including appropriate labels and a legend. Reverse the direction of the y-axis to place smaller values on top, as appropriate for magnitudes. Give your plot appropriate axis labels, including units.

NGC 6212 is a barred Seyfert galaxy, and a better fit would be obtained by considering a Sersic profile for the bulge plus an exponential disk (e.g. Kim et al. 2017, ApJS, 232, 21). So don't publish your result!