Physics 134L, Fall 2021
Professor Maxwell Millar-Blanchaer

Lab 2 **Name:** Tim Marquez

Due date: Monday, October 18, by 5:00 pm, submitted through Gradescope

Read through this entire lab before you start. This lab will introduce you to the idea of astronomical coordinate systems, the Messier Catalogue, and astronomical spectra. To complete this lab, you will need to read Chromey chapter 3 (you can also check out the links on the class website to the two celestial coordinate tutorials/presentations). You may also find it useful to read the Wikipedia entries for "Right ascension," "Declination," and "Sidereal time."

Part 1: Celestial Coordinates

Objects in the sky have "equatorial" coordinates just as do objects on the Earth. But instead of "longitude" for the east-west coordinate, astronomers say "right ascension" (or "RA"), and instead of "latitude" for the north-south coordinate, they say "declination" (or "δ"). Right ascension is typically measured in hours, not degrees. Dividing the full circle of the celestial equator into 24 hours, each hour corresponds to 15 degrees.

We'll begin by revisiting the same datafile that we used in Lab 1: `object.fits`. **In ds9, open `object.fits`. Click on "file/display_header" to display the header.** The entries "RA" and "Dec" give the coordinates in the sky where the telescope was pointing while taking the image. Also notice the entry commented "start time of the observation." This is the time (in Universal Time, or UT) when the shutter opened for the CCD image. UT corresponds to the local time on the prime meridian (longitude = 0 degrees), which passes through the observatory at Greenwich, England. This time zone is 8 hours ahead of Pacific Std Time, and 7 hours ahead of Pacific Daylight Time.

On a certain day, the Sun crossed the meridian at 12:00 noon UT in Greenwich, England. Santa Barbara lies at 119.7 degrees W longitude. On that same day, what is the UT time when the Sun crosses the meridian in Santa Barbara?

$\frac{119.7^\circ}{15^\circ}$ 1hr = 7.98 8 pm

Using ds9 to get information from the headers, list the RA, δ, date, and UT for the file `object.fits`.

RA = 14:32:33.877 UT: 7 a.m UT

δ = -44:13:57.80

date = March 14 2013

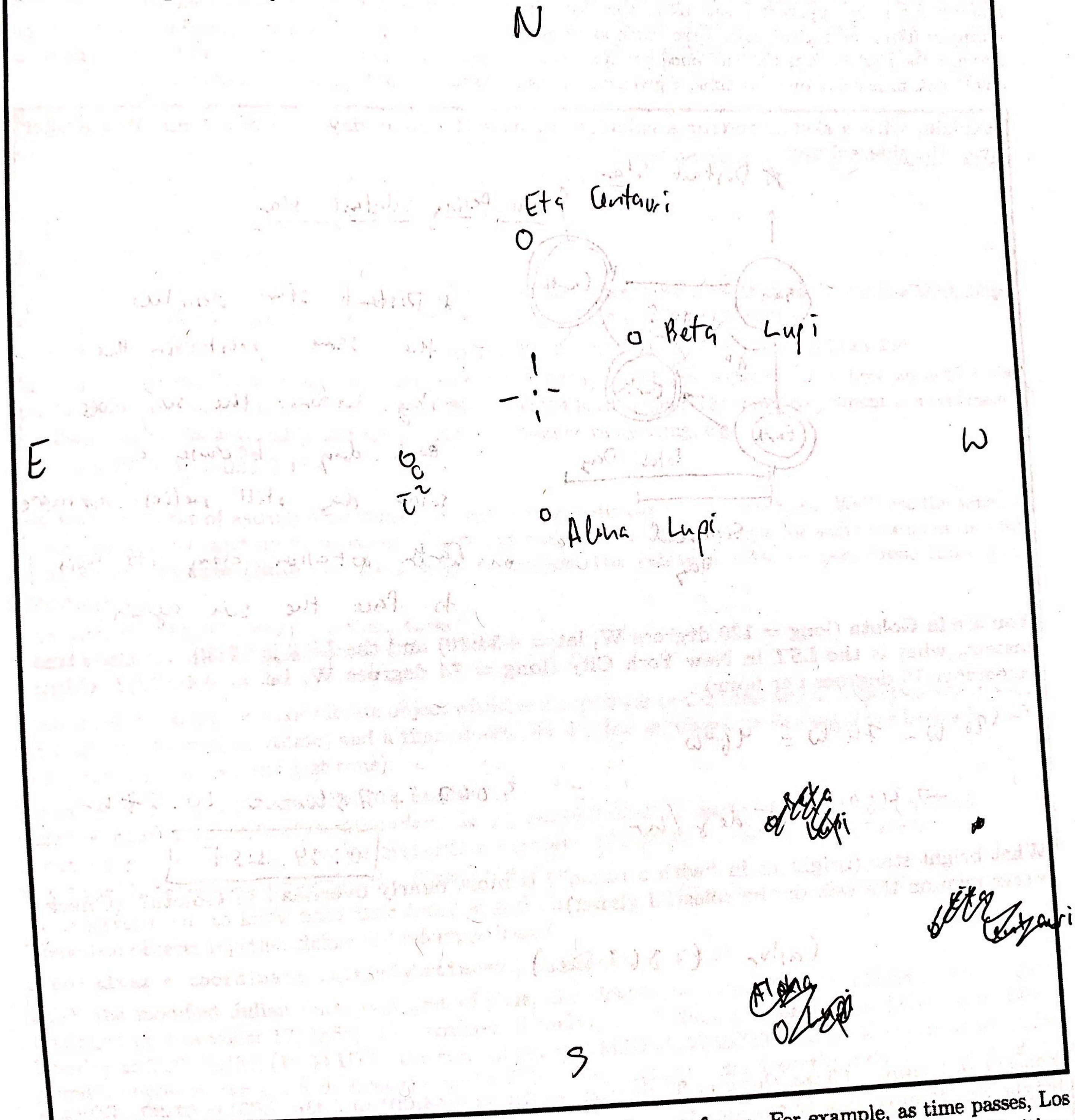
On the next page, sketch a map of the part of the sky where the image was taken large enough to include a couple of bright, named stars (please label them). Check out the Stellarium website if you need some help. Note that RA is defined so that as the Earth turns, the RA of objects on the meridian increases with time. Draw your map with N up and E to the left (as it would appear if you were facing the southern horizon). Does RA increase to the right, or to the left?

RA increases to the right

N

E

Sketch the region of sky here:

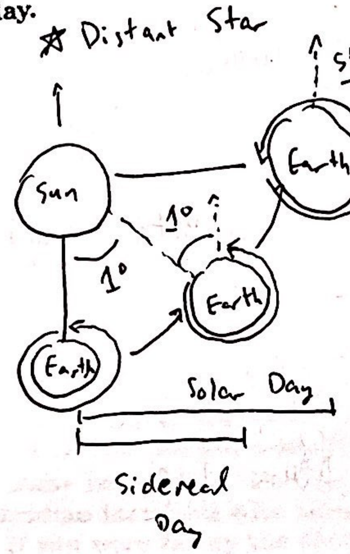


Positions on the Earth stay fixed in the planet's rotating reference frame. For example, as time passes, Los Angeles always remains about 100 miles, or 160 km, east-southeast of us here in Santa Barbara. Positions in the sky are different. The stars remain basically fixed in their current positions in the sky (they do move across the celestial sphere, but very slowly). As the Earth rotates about its axis, the entire sky appears to rotate from the perspective of someone fixed on the Earth's surface.

Part 2: Sidereal Time

The word "sidereal" means "with respect to the stars." The current Local Sidereal Time (LST) is the value of the RA in the equatorial coordinate system that is crossing your meridian at the moment. Since the coordinates of stars are essentially constant over very long times, at a given LST you will always find the stars in the same apparent positions in the sky. Sidereal time is not the same as solar time (which we normally use) – at a given solar time (such as noon), we find the *Sun* in the same position, not the stars. Because the Earth orbits the Sun once per year, the sidereal day is about 4 minutes shorter than the solar day. Thus, measuring by solar time, a given star rises and sets about 4 minutes earlier every day.

Explain, with a sketch and/or a calculation, why the solar day is about 4 minutes longer than the sidereal day.



* Distant star reaches the same point in the sky before the sun does each day, because a solar day still relies ~~on~~ more on Earth rotation along its axis to face the sun again

You are in Goleta (long = 120 degrees W, lat = +34:30) and the LST is 07:30. At the same instant, what is the LST in New York City (long = 74 degrees W, lat = +40:00)? (hint: remember 15 degrees per hour)

120°W − 74°W = 46°W

→ 46°W / 15 deg/hr = 3.067 + 7.5hr = 10.57hr

$= \boxed{10:34 \text{ LST}}$

What bright star (bright as in "with a name") is most nearly overhead in Goleta? (Check a star map on the web or the celestial globe.)

Castor (+36° Dec)

You are in Urumqi, China (long = 87 degrees E, lat = +44:00) and the LST is 07:30. What bright star is most nearly overhead?

Deneb (+45° Dec)

The Messier Catalog

In the late 1700s, Charles Messier and his colleague Pierre Mechain compiled a catalog of astronomical objects that do not look like stars, and that confused them in their main work, which was comet-hunting. Over the years, this catalog has come to be recognized as, in effect, a list of everybody's favorite astronomical objects. A full list (with pictures) may be found on Wikipedia under "List of Messier Objects." Useful lists of the Messier objects sorted in various ways (by type of object, by location in the sky, etc) may be found at `http://www.maa.clell.de/Messier/indexes.html`. The answers to many of the following questions can most easily be found by looking at an appropriate web site. I will provide some suggestions, but it is best if you poke around and find your own favorite sources of information. If you use a web page in answering a question, write down a link to it as part of your answer.

We will be looking mostly at objects identified as "open clusters." How many open clusters are there in the Messier catalog? Get the answer to this in the cleverest way you know (ie, the least work, consistent with doing it yourself), and explain how you did it.

I went to maa.clell.de/Messier/indexes.html and went to "Messier Objects by Type" > Open star Clusters and read that there are 33 open clusters.

I even counted 33 on the same area of the website.

Suppose you live on a planet circling a star in a distant galaxy, but your planet and galaxy are similar in all important respects to our own. Say your astronomer Reissem Selrahc makes a list of the 110 brightest non-stellar objects in your sky. How many of them should you expect to be open clusters? What is the uncertainty in this estimate?

Since Messier catalog contains 110 objects, the uncertainty is 33/110 or 30%.

At the bottom of the Wikipedia entry mentioned above, there is a nice map of the sky coordinates of the Messier objects. The blue sinusoidal band across the map is the Milky Way. What is the Milky Way, anyhow? (In physical terms)? Why does it appear as a (more or less) sinusoidal band on the map?

The Milky Way is a barred spiral galaxy and is the galaxy that contains our solar system

It appears sinusoidal because it's a flattened rectangular display of the way we see the galaxy from Earth. We're also tilted on an axis so the Milky Way isn't exactly aligned with our equator and will show up sinusoidal

Open Clusters and Galaxies have different (from each other) distributions in the sky, with respect to the Milky Way. What is the difference? What is the reason for this difference?

~~The~~ Open Clusters are usually found around the Milky Way, whereas Galaxies are grouped together regardless of the Milky Way's pattern. This is because Open Clusters are held together by stars' gravitational pulls. So, they are found to be around the Milky Way, until they are disrupted by encounters with the galaxy center or other clusters. On the other hand, Galaxies are found around each other due to them being bound gravitationally to each other early on in the time of the Universe.

What is the northernmost Messier object, and what is its declination?

M82 Cigar Galaxy → $\delta = +69° 40' 47''$

What fraction of the area of the celestial sphere lies between this declination and the north celestial pole (ie, at higher declination)? Given that there are 110 Messier objects in the sky, and supposing that they are randomly distributed, how many would you expect to find in this area? What is the uncertainty in this estimate? Is it surprising to find none in this polar cap? Show your work.

North Celestial Pole $\delta = +90°$

$M82 \rightarrow +69° \; 40' \; 47'' = 69° + \left(\frac{40 \text{ min}}{60 \text{ min/deg}}\right) + \left(\frac{47 s}{3600 s/\text{deg}}\right)$

$= 69.6797$ deg

$A = \int_{lat_1}^{lat_2} 2\pi r \, d\theta = 2\pi r \left(-\sin\theta \Big|_{lat_1}^{lat_2}\right) = 2\pi r(\sin(90°) - \sin(69.67979°))$

$= 2\pi r(0.06223) = \boxed{0.1245 \pi r}$

We would expect to find at least $\boxed{6}$ Messier objects

$\left(\frac{0.1245 \pi r}{2\pi r}\right) \times 110$

What is the southernmost Messier object, and what is its declination? Answer the same questions as above, only for the southern polar cap that contains no objects. How do you explain the difference in the sizes of the two empty polar caps?

$M7 \rightarrow \delta = -34° \; 47' \; 34''$

$A: 2\pi r(\sin(lat_2) - \sin(lat_1))$

$= 2\pi r(\sin(-34.7927) - \sin(-90)) = 2\pi r(0.42939)$

$= \boxed{0.859 \pi r}$

$\left(\frac{47'}{60 \text{ min/deg}}\right) + \left(\frac{34}{3600 s/\text{deg}}\right)$

$\frac{0.859}{2} \times 110 \rightarrow$ we should see 47.

It is more surprising to find none in this area

What are the maximum and minimum declination visible from the north pole? What fraction of the sky is visible from the north pole?

Max dec $= +90^\circ$

Min dec $= 0^\circ$

$\frac{1}{2}$ of the sky is visible

Repeat the question above for Santa Barbara (latitude 34°) and for an observer on the equator.

For Santa Barbara: Max dec: $= +90^\circ$

Min dec $= 34 - 90 = -56^\circ$

For Equator: Max dec $= +90^\circ$

Min dec $= -90^\circ$

$\frac{1}{2}$ of the sky is visible

To get quantitative, an angle of 1 arcsec on the sky maps into a distance $d = fl/206264.80$ measured in the focal plane of a telescope having focal length fl. Explain the origin of the magic number 206264.80. Check out Chromey Section 5.3 if you get stuck here.

This number is the number of AU in 1 parsec

There's also 206264.80" per 1 rad.

The header of the image cluster.fits has a keyword FOCALLEN that gives an estimate of the telescope's focal length. The mean angular diameter of the Moon is about $\frac{1}{2}$ degree, or 1800 arcsec. What is the diameter of an image of the Moon made by this telescope, in mm?

$fl = 6491 mm$ ang diameter of the Moon = 1800"

$d = 6491 mm / 206264.8 = 0.03146$ mm/arcsec $= \boxed{56.628 mm}$

Other keywords give the widths and the number of detector pixels, in the x- and y- directions. What is the (x,y) size of the detector as projected on the sky, in arcsec? Will the entire image of the full Moon fit on it?

Pixel height = 18 microns

y = 1024 × 18 microns = 18432 arcsec

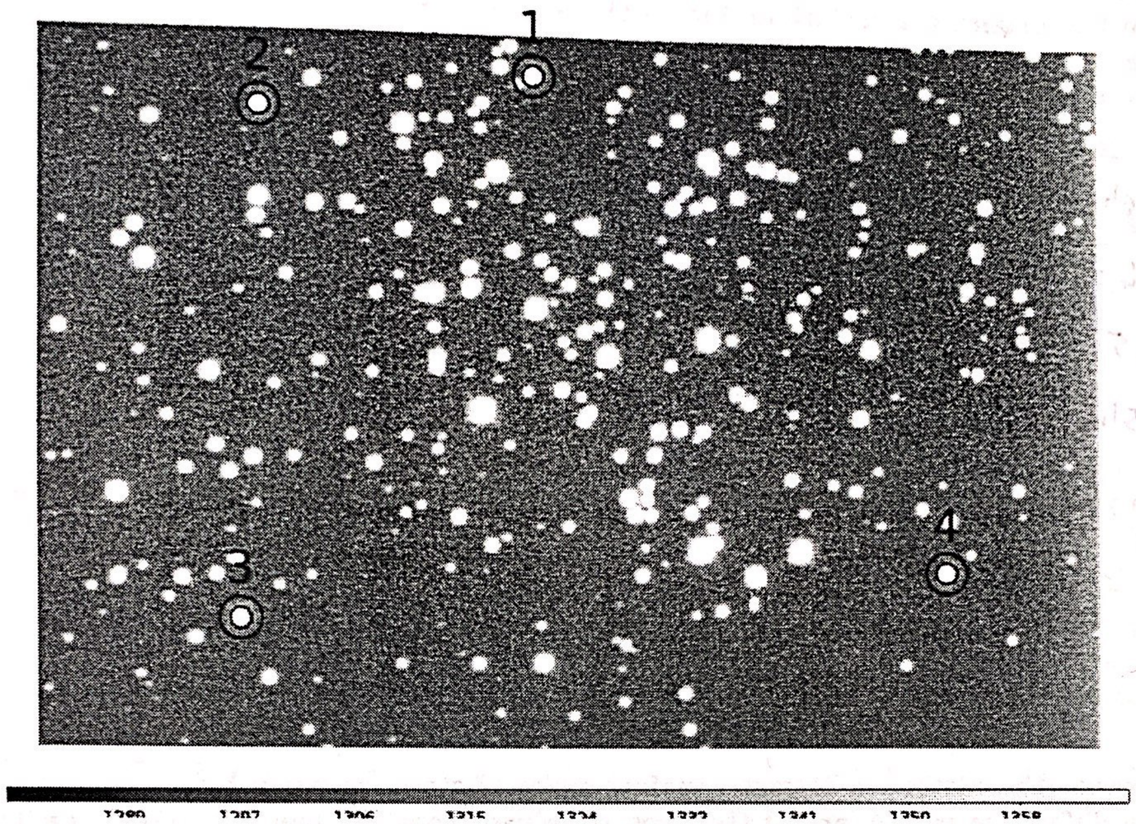
x = 1536 × 18 microns = 27648 arcsec

How do we know if the header value for the image scale of the telescope is correct? We will measure it. Use ds9 to open the image cluster.fits. Click on "scale" and "zscale" so you can see faint stars. Display the header, and write down the RA, δ coordinates to which the telescope is pointing.

RA = 8hr 51min 25s

$\delta = +11° 48' 18''$

After doing a Y-invert (see below), your image should look like this, except for the little circles and numbers, and the color scheme I used to make the numbers show up better.



By no coincidence at all, this is the center of an open cluster in the Messier catalog. By matching coordinates in one of the Messier catalog lists that you have already used, identify which open cluster it is. Write down the name.

RA = 8h 51.25m → M67

δ = 11° 48.18'

In the ds9 "Zoom" menu bar (not the Zoom button), click "Invert Y" to flip the image upside down. This will facilitate comparison with other images that are in a more standard format. (Notice, for future reference, that other flip and rotate operations are possible.) Now open a browser and go to `http://www.sky-map.org`. This is a nice piece of planetarium software (similar to the Stellarium page we used earlier). Go to "Home," and enter the name of the cluster in the "Find Object" window. When the map comes up, point to the small elliptical icon in the upper left called "DSS," and select "DSS2 All Sky Survey." These data come from various releases of the Digital Sky Survey. Zoom in 3 or 4 clicks on the size scale, and you should see a familiar star cluster (a bit off center). Drag it to the center of the window, and zoom it to whatever degree makes you comfortable. Now when you drag the cursor over a star image you will see lots of information about each star, including a long catalog number, and (most importantly) the star's equatorial coordinates.

Identify the 4 numbered stars in the image above on the sky-map.org site, and list their RA, δ values in a table, 1 row per star. Also measure the $\{x, y\}$ coordinates of each star on cluster.fits, using the cursor to pick out the brightest point in each star. Do this carefully, zooming so that setting the cursor is easy, and adjusting the "scale" options so you can easily see the brightness variations inside the star images. Make a subjective guess about the error (in pixel units) with which you can measure the star positions. Put this in the table too, under error_g.

Star	RA	δ	x position	y position	error_g
1	8h 51m 27.01s	+11° 51'52.6"	704.952	58.9879	± 2
2	8h 51m 42.36s	+11° 51'29.1"	311.978	112.009	± 5
3	8h 51m 43.56s	+11° 44' 26.4"	283.097	838.996	± 3
4	8h 51m 3.51s	+11° 45' 02.7"	1308.98	771.988	± 1

What do you think is your largest source of error (the one that dominates your estimate of error_g)?

~~star 1~~ The x position since the values are much larger.

Now compute the distances between various pairs of stars, as given below. Do this first by using the difference in RA and d that you obtained from sky-map.org. Formally this is an exercise in spherical trigonometry, but because all of these stars are very close together on the sky, we may use small-angle approximations. In this case we get sufficient accuracy by taking

$$\Delta r = \sqrt{\left((\Delta\delta)^2 + (\Delta \mathrm{RA} \cos\delta)^2\right)},$$

where Δr is the angular separation between two stars, $\Delta\delta$ is the separation in Declination, and ΔRA is the separation in Right Ascension, with all angles are expressed in units of angle (use arcsec). Remember that RA is normally expressed in units of time, not angle – one second of RA (the difference between RA = 08:30:00 and 08:30:01, for instance) equals 15 arcsec. You should think about where the factor $\cos(\delta)$ comes from. Try computing the RA and Dec values as arcseconds first by hand, and then you can double check using astropy coordinates.

Also compute the separation between these pairs of stars in units of pixels, using your measured values of x position and y position. In this case the normal Pythagorean law may be used, with no $\cos(\delta)$ factor. (Think about why.) Use your estimates of error_g and standard propagation-of-error rules (see the textbook by *Taylor* linked on the Lab 2 tab in Gauchospace for a refresher) to estimate the errors in these separations which we will call error_p. In the space below, show the formula(s) you used for calculating the error_p values. Then put all of the data into the table on the next page.

$\Delta p = \sqrt{(\Delta x)^2 + (\Delta y)^2}$

error_p: uncertainty in x: $\frac{\Delta(\Delta x)^2}{(\Delta x)^2}$ uncertainty in y: $\frac{\Delta(\Delta y)^2}{(\Delta y)^2}$

$\left(\frac{\text{error\_p}}{\Delta p}\right) = \left(\frac{\text{uncertainty } x+y}{\Delta p}\right)\frac{1}{2} \longrightarrow$ since $q = x^n \rightarrow \frac{\delta q}{|q|} = n\frac{\delta x}{|x|}$

Scanned with CamScanner

Star Pair	Δr (arcsec)	Δp (pixel)	Error_p (pixel)	Scale ($\frac{\text{arcsec}}{\text{pixel}}$)	Error_s ($\frac{\text{arcsec}}{\text{pixel}}$)
{1,3}	510.596	886.78		0.5758	
{1,4}	540.6	934.462		0.5785	
{2,3}	417.0885	727.56		0.5733	
{2,4}	695.863	1195.653		0.5820	
{3,4}	601.796	~~[illegible]~~ 1028.069		~~[illegible]~~ 0.5854	

For each star pair, compute the image scale $\Delta r/\Delta p$ in units of arcsec/pixel, and enter this value in the table. Use Taylor's error propagation rules, starting from your estimates of error_p, to estimate the error in your derived value for the image scale (which we will call error_s). Assume that the star separations derived from www.sky-map.org positions have negligible errors. Put your error_s values in the table.

Now you have five not-quite-independent measurements of the image scale. Do they agree with each other, within the plausible errors? If not (and especially if the disagreement is very large), the most likely explanations are (a) there is a mis-identified star, or (b) there is an error in computation. In either of these cases, you should go back and correct the error before proceeding.

Yes, they do agree with each other.

Average your results for image scale (we will do weighted averages later), and estimate the uncertainty of this mean value, again using the rules described in Taylor's book. From the image scale, compute the telescope focal length and its uncertainty. Compare this to the focal length value found in the fits image header. Show your work.

$$\text{Avg} = \frac{\sum_i \text{scale}}{4} = \boxed{0.5774}$$

Part 4: Stellar Spectra

The last part of this lab will be an introduction to spectra. Astronomical data comes in two basic forms: images and spectra. You have already seen images. Spectra are a bit different: we lose at least some of our ability to resolve an object and see its shape, but we spread out its light into its colors, or wavelengths. The course webpage includes a spectrum as it appears on a detector. It is a colored spectrum of the Sun, showing many black parts where there is less light. These are absorption lines and give a whole lot of information about the physics of the Sun. Just like SExtractor takes the images and extracts the positions and brightnesses of stars, we can extract the flux of a star as a function of wavelength.

For the rest of this lab, you will use three spectra contained in the files `spectrum1.dat`, `spectrum2.dat`, and `spectrum3.dat`. You will complete the lab in a jupyter notebook (please create a new one, lab2b). The spectra are of three stars. The data files have two columns: the first is wavelength in Angstroms ($1\,\text{Å} = 10^{-10}$ m), and the second is flux density in units of $\mathrm{erg\,s^{-1}\,cm^{-2}\,Å^{-1}}$. You may read the files using, e.g.,

```
spectrum = np.loadtxt(specfile)
lam = spectrum[:, 0]
flam = spectrum[:, 1]
```

Please be aware that `lambda` has a special meaning within python, so don't try to define it as a variable!

Make a plot of each star's spectrum as a function of wavelength. Please label the figures nicely using, e.g.,

```
plt.xlabel(r'Wavelength ( $ \AA $ )')
```

The `r` before the string tells matplotlib that it should use Latex to render this. Please get the units right for the vertical axis. Hint: if you are unsure of how to write a symbol in Latex, try `http://detexify.kirelabs.org`.

Now let's try to learn something about these stars. We can try to approximate the stars as blackbodies using the Planck function (look it up in Wikipedia). **Write down the Planck function in a markdown cell in your notebook so that it has as close as possible to the same units as the flux density of your spectra (i.e. make sure it's per Angstrom, not per Hertz or per nm).** The units won't be completely identical: the Planck function will still be per steradian. This will be important soon.

By eye, find the temperature and normalization of the Planck function that best fits each stellar spectrum. You should choose the temperature that gives a good fit to the peak wavelength (a blackbody isn't a great fit to `spectrum2.dat` for reasons that we will discuss later), and multiply the Planck function by a number to put it on the same scale as your spectrum. This number will be very small. Make the three stellar spectrum plots again, with your best-fit Planck functions overplotted.

You have fit two parameters by eye to each star's spectrum: the temperature and a normalization factor to the Planck function. What units does your normalization constant need to have in order for the units to work out?

$$\frac{\text{ste} \cdot \text{Å} \cdot \text{cm}^2 \cdot \text{s}}{\text{erg}}$$

In another markdown cell, give your best-fit temperatures and angular diameters, in units of milliarcseconds (3.6×10^6 mas $= 1°$). You'll have to first convert steradians to square milliarcseconds for the last part. **Finally, compute the approximate physical radii of the stars, in units of the Solar radius, using parallaxes of 0.73 mas for the first star, 130.2 mas for the second star, and 11.3 mas for the third star.** Look up the value of the Solar radius if you need it.

In this box put the name of your partner, and describe how you worked together to complete this lab.

Aubrie Payne — we met up a few times to compare answers.

Upload this worksheet and the pdfs of your jupyter notebook(s) to the Gradescope assignment link on the "Lab 2" tab of the Gauchospace page. Each group member should upload their own version of this page, but it is ok if your code submissions are the same if you worked in partners the whole time. Clearly label the different parts of the lab using the Markdown features of Jupyter notebook and using code comments. You want to make it easy for the TA to find your work.

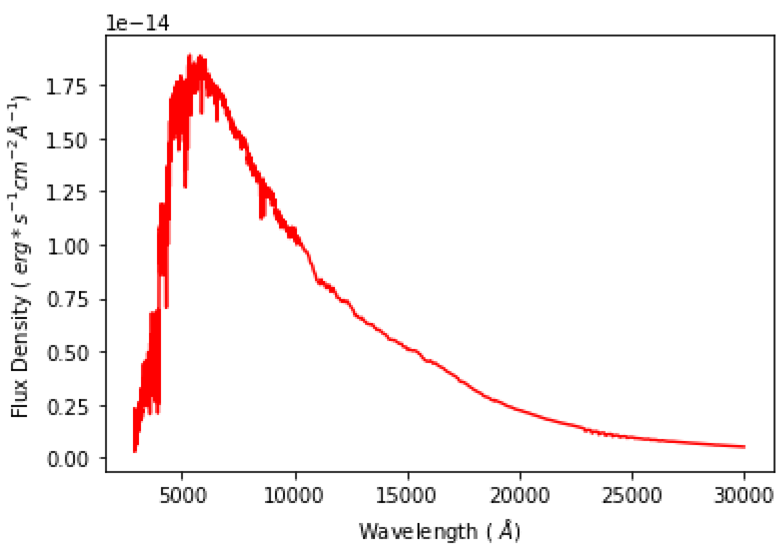
In [71]:
```
#import necesssary libraries
import os
import numpy as np
from astropy.io import fits
import matplotlib.pyplot as plt
import matplotlib as mpl
```

In [72]:
```
#import spectrum text files after uploading them to jupyter
#define and index the two columns in each .dat file
spec1 = np.loadtxt('spectrum1.dat')
lam1 = spec1[:, 0]
flam1 = spec1[:, 1]
```

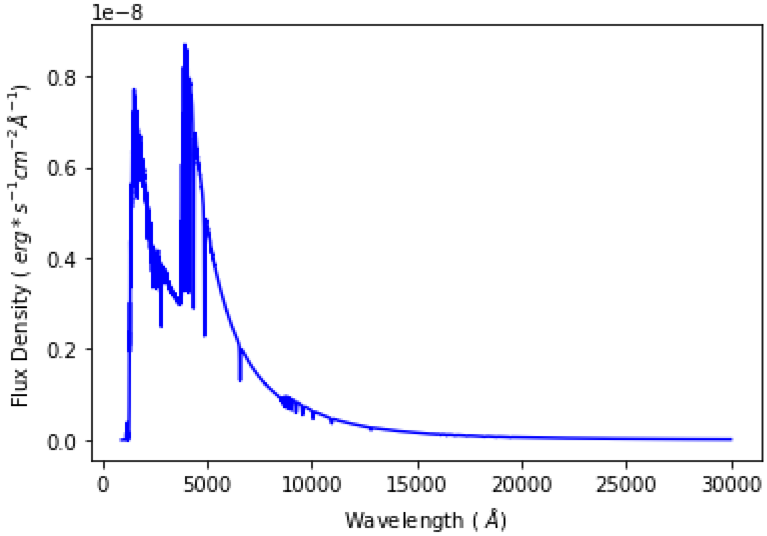
In [73]:
```
spec2 = np.loadtxt('spectrum2.dat')
lam2 = spec2[:, 0]
flam2 = spec2[:, 1]
```

In [74]:
```
spec3 = np.loadtxt('spectrum3.dat')
lam3 = spec3[:, 0]
flam3 = spec3[:, 1]
```

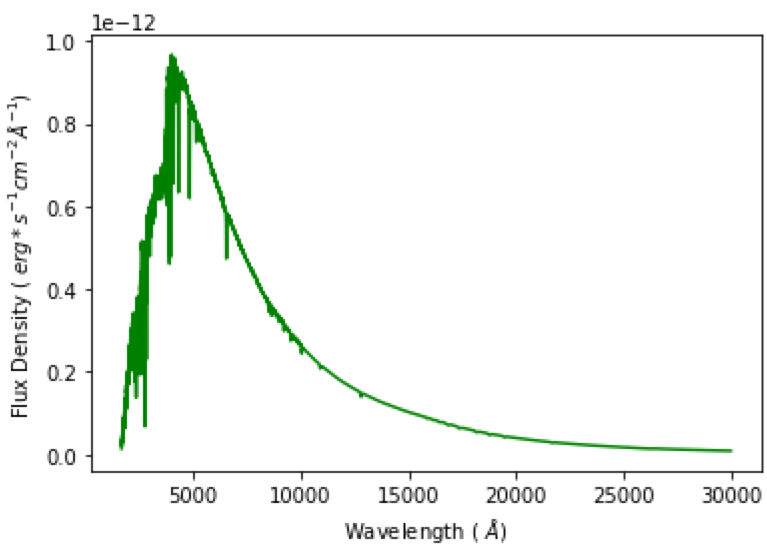
In [99]:
```
#plot each spectrum and label the axes
plt.ylabel(r'Flux Density ( $erg * s^{-1}cm^{-2}\AA ^{-1} $)')
plt.xlabel(r'Wavelength ( $\AA $)')
plt.plot(lam1, flam1, color = 'r')
plt.show()
```



In [100]:
```
plt.ylabel(r'Flux Density ( $erg * s^{-1}cm^{-2}\AA ^{-1} $)')
plt.xlabel(r'Wavelength ( $\AA $)')
plt.plot(lam2, flam2, color = 'b')
plt.show()
```



In [101]:
```
plt.ylabel(r'Flux Density ( $erg * s^{-1}cm^{-2}\AA ^{-1} $)')
plt.xlabel(r'Wavelength ( $\AA $)')
plt.plot(lam3, flam3, color = 'g')
plt.show()
```



Plank Function: B(T) = (2hc^2/lambda^5) / (e^(hc/(lambda)kT) - 1) where h = Planck's constant, c = speed of light, lambda = wavelength, k = Boltzmann's constant, T = temperature.

In [98]:
```
#Define a function with the two arguments of Temperature and Lambda. I defined tl
def planck(T, Lambda):
    c = 3*10**10
    h = 6.6261*10**(-27)
    k = 1.3807*10**(-16)
    B = (2*h*c**2 / Lambda ** 5)/(np.exp(h*c / (Lambda * k*T) - 1))
    return B
```

In [ ]:
```
```

In [1]:
```
import os
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy import coordinates, time
from astropy import units as u
#importing all the necessary libraries
```

In [2]:
```
imfile = os.path.join(os.sep, r'E:\Documents\134L\object.fits')
image = fits.open(imfile)[0].data
header = fits.open(imfile)[0].header
#importing the image from my directory
```

In [3]:
```
mjd = header['MJD-OBS']
#setting "mjd" as the header for MJD keyword
```

In [4]:
```
radec = coordinates.SkyCoord(header['RA'], header['DEC'], unit = (u.hourangle, u.
location = coordinates.EarthLocation(lon = header['LONGITUD'], lat = header['LATI
t = time.Time(header['MJD-OBS'], format = 'mjd')
#setting variables to our object's location on the celestial sphere and time of c
```

In [5]:
```
frame_altaz = coordinates.AltAz(obstime = t, location = location)
#creating a reference frame of when and where the object was observed
```

In [6]:
```
lst = t.sidereal_time('apparent', header['LONGITUD'])
print(lst)
```

```
13h56m46.0333s
```

In [9]:
```
print(header['LST'])
#comparing the header in the fits file to the computed LST at time of observatior
```

```
13:56:45.76
```

As seen in the last 2 cells, the LST header in the fits file is slightly earlier than the computed LST of 13h 56min and 46.033s.

In [8]:
```
coord_altaz = radec.transform_to(frame_altaz)
altitude = coord_altaz.alt
```

In [10]:
```
print(altitude)
```

```
74d07m30.382s
```

This value is equal to 74.12511 deg

In [11]:
```
print(header['ALTITUDE'])
#comparing altitude of the telescope pointing
```

```
74.1248588
```

As seen above, the computed altitude is greater than the value that the header gives.