

```
In [1]: import matplotlib.pyplot as plt
        %matplotlib inline
```

Plate scale and optimal pixel size

1. The TMO 24" telescope is a f/5 system. The current detector has a pixel size of 3.76 microns. What angle in the sky would one pixel subtend?
2. Do you think this is a good pixel scale and why or why not?

```
In [2]: #scale is 1 / focal length = 1 / (focal ratio * diameter)
        #multiply by 206265 to convert from radians to arcsec
        scale=206265/(5*0.6e6)*3.76
        print('scale: {:.2f} arcsec/pixel'.format(scale))
```

```
scale: 0.26 arcsec/pixel
```

Typical seeing at a ground based site give images with a full width half max (FWHM) of about an arcsecond, maybe a little better. 0.26 arcsec/pixel means that light from a point source is spread over pretty many pixels even with the best seeing, which means that readout noise will contribute more than necessary for faint, low-background observations. It would be better from that perspective to have a larger plate scale, something like half of the best seeing.

Two mirror telescopes

TMO is a Newtonian f/5 telescope with a parabolic primary mirror. What do you expect images to look like on and off axis? The ARC 3.5m is a Ritchey Chretien f/10 telescope. What do you expect images to look like on and off axis? Sketch the beams (a cone) coming into the focal plane for both TMO and the ARC 3.5m. What are the relative plate scales of the two telescopes, without calculating the plate scale of either?

1. With a parabolic primary, there will be no aberrations on-axis, but there will be both coma and astigmatism off-axis; coma will dominate until you get to large field angles.
2. For a Ritchey-Chretien, the primary and secondary shapes are matched to eliminate both spherical astigmatism and coma. So there will be no aberrations on-axis, and only astigmatism off-axis.

```
In [3]: # plate scale is inversely proportional to both focal ratio and diameter
        print('ratio of TMO/ARC plate scales: ', (10*3.5)/(5*0.6))
```

```
ratio of TMO/ARC plate scales: 11.666666666666666
```

Diffraction limited imaging

Betelgeuse has a diameter of about 4 AU and is located at a distance of about 200 pc. How big a telescope would be required to marginally resolve the surface of Betelgeuse using the Rayleigh criterion ($1.22 \lambda/D$), assuming you could achieve diffraction limited images (e.g., with adaptive optics), both in the optical and in the near-IR?

```
In [4]: #angular diameter of Betelgeuse
angdiam=4*1.496e13/(200*3.086e18)
print('angular diameter in arcsec: ',angdiam*206265)
# D = 1.22 lambda / angular diameter
print('optical: {:.2f} meters'.format(1.22*5500e-10/angdiam))
print('near-IR (H band): {:.2f} meters'.format(1.22*1.6e-6/angdiam))
```

```
angular diameter in arcsec: 0.019998213869086196
optical: 6.92 meters
near-IR (H band): 20.13 meters
```

Focal reducer

Imagine you have a detector that has pixels that are 24 microns on a side. You want to design a camera to match this detector to the ARC 3.5m telescope, which produces an f/10 beam in the telescope focal plane.

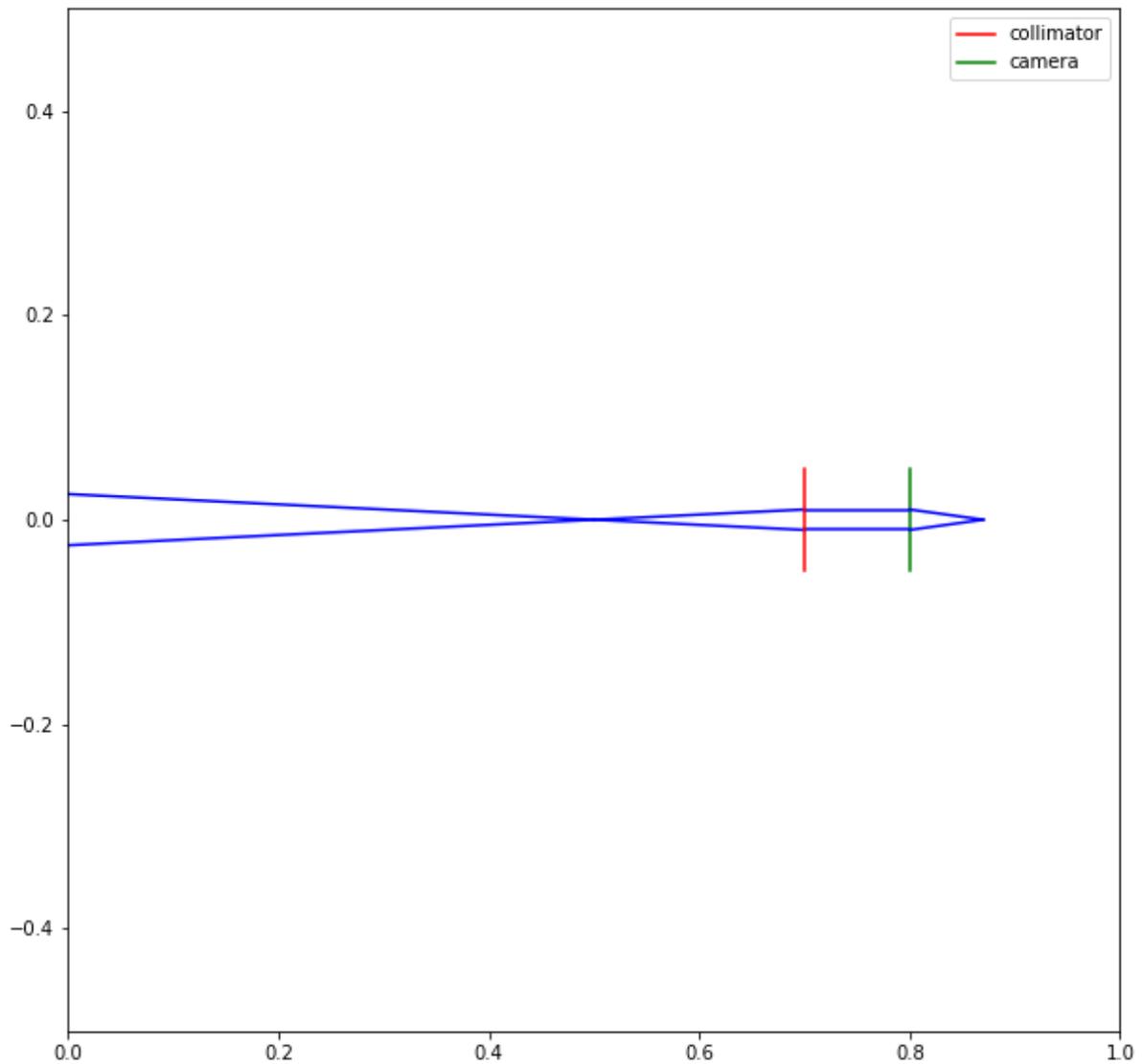
What is the pixel scale in the telescope focal plane?

What would be a desired pixel scale to match your detector, under the assumption that APO is a seeing limited site, with best seeing about 0.8 arcsec?

Sketch a design for a reimaging camera that would achieve your pixel scale, showing the rays coming into the focal plane, the reimaging optics, and the rays coming to the reimaged focal plane, with approximately correct beam sizes.

```
In [5]: scale=206265/(10*3.5e3)
print('scale : {:.2f} arcsec/mm, {:.2f} arcsec/pixel'.format(scale, scale*.
# good scale would be something like 0.4 arcsec/pixel (2 pixels per FWHM at
reducer_ratio=scale*.024 / 0.4
print('focal reducer ratio: {:.2f}'.format(reducer_ratio))
# we want a final focal ratio
reducer_fratio = 10*reducer_ratio
print('reducer focal ratio: {:.2f}'.format(reducer_fratio))
# we want a f/10 coming in, a collimator to collimate it, then a camera lens
plt.figure(figsize=(10,10))
plt.ylim(-0.5,0.5)
plt.xlim(0,1)
plt.plot([0,0.5],[0.025,0],color='b')
plt.plot([0,0.5],[-0.025,0],color='b')
plt.plot([0.5,0.7],[0,0.01],color='b')
plt.plot([0.5,0.7],[0,-0.01],color='b')
plt.plot([0.7,0.8],[0.01,0.01],color='b')
plt.plot([0.7,0.8],[-0.01,-0.01],color='b')
plt.plot([0.7,0.7],[-0.05,0.05],color='r',label='collimator')
plt.plot([0.8,0.8],[-0.05,0.05],color='g',label='camera')
plt.plot([0.8,0.8+.02*3.5],[0.01,0],color='b')
plt.plot([0.8,0.8+.02*3.5],[-0.01,0],color='b')
plt.legend()
plt.draw()
```

```
scale : 5.89 arcsec/mm, 0.14 arcsec/pixel
focal reducer ratio: 0.353597
reducer focal ratio: 3.54
```



Spectrograph resolution

Imagine you want to make a measurement of the He and Ca H lines in a spectrum of a star. He has a rest wavelength of 3970 Å, Ca H of 3968.5 Å.

- What resolving power (R) is required to resolve these two lines?
- Imagine you are observing this star with DIS with the grating which has 400 lines/mm and you find you have a resolution of about 4 angstroms with a 1 arcsecond slit. What slit width would you need to go to to resolve these two lines? Would the use of this slit width cause any significant trouble, and if so, what?
- What slit width would you need if you used the 1200 lines/mm grating?

```
In [6]: R=3970/1.5
print('need R={:.2f}'.format(R))
print('required slit width with 400 lines/mm grating: {:.2f} arcsec'.format(R))
# this would be a problem in typical seeing, as only a small fraction of light
# make it down the slit
print('required slit width with 1200 lines/mm grating: {:.2f} arcsec'.format(R))
```

```
need R=2646.67
required slit width with 400 lines/mm grating: 0.38 arcsec
required slit width with 1200 lines/mm grating: 1.12 arcsec
```

KOSMOS resolution

Last weekend, I obtained some calibration lamp (line lamps) exposures with KOSMOS with both the 1.18 and 2.1 arcsec slit.

How do you expect that these will differ in appearance, if at all? Load two images from /home/holtz/raw/apo/oct21/ (via web at astronomy.nmsu.edu/holtz/Q4NM01 : Q4NM01/UT211031/kosmos/Ne.0010.fits Q4NM01/UT211031/kosmos/Ne.0014.fits Choose a line and measure the FWHM in each of the two frames, and discuss your results.

Width of line is broader with broader slit, roughly by the ratio of the slit widths

Near-IR data and observing

What are some differences between working in the optical and working in the near-IR?

In the near-IR H band, the sky brightness might be something around 13.5 magnitudes / square arcsec

If you observe a star with H=13.5 and make a 1% error in determining the background level, by how much will you make an error in your measurement of the star brightness?

If you observe a star with H=18.5 and make a 1% error in determining the background level, by how much will you make an error in your measurement of the star brightness?

How do we typically measure the background level?

In the near-IR, the sky is much brighter than in the optical. We also use different types of detectors, and need to do multiple non-destructive readouts to avoid kTC noise. Readout noise is typically larger than in optical detectors.

*If background is equal to star flux, then $\text{flux} - 0.99(\text{background}) = 0.01 * \text{flux}$, i.e. 1% error in sky leads to 1% error in measured flux*

*If background is 5 magnitudes (=100x) brighter, then $\text{flux} - 0.99(\text{background}) \sim 2 * \text{flux}$, i.e. 1% error in sky leads to factor of two error in measured flux!!*

Background is typically measured in the optical by getting the sky from an area nearby the object. This does depend on flat fielding being good. In the IR, since we need the sky to much better accuracy, we typically dither exposures so as to be able to subtract the bulk of the sky from the same pixels as the object was observed on.

Spectroscopic data reduction

Working with He.0015.fits, Ne.0016.fits, and Ar.0017/fits lamp exposures with KOSMOS from UT211030,

What is the approximate dispersion of KOSMOS? What is the approximate resolution, both $\delta\lambda$ and R?

```
In [7]: #measure pixel positions of two lines of known wavelength, dispersion is de
#reolution is width (FWHM) of lines, times the dispersion to get delta(lambda)
# R==lambda/delta(lambda)
```

```
In [ ]:
```