CCD Reduction part 2 - Count Statistics

This notebook will guide you through calculations needed for the CCD characterization lab.

Look for items labeled "student exercise" where you're supposed to fill in the blanks.

Noise from counts

How much noise would we expect the dark electrons to add?

First, recall how propagation of error works: the dark frames have two contributions to the noise-per-pixel, read noise and counting (Poisson) noise:

$$\sigma^2_{darkframe} = \sigma^2_{darkelectrons} + \sigma^2_{readnoise}$$

Then recall that dark current is a Poisson process, so $\sigma_{darkelectrons} = \sqrt{\lambda_{darkelectrons}}$.

For our 10s frames, we estimate that $\lambda_{darkelectrons} = \mu_{darkframe} - \mu_{biasframe} =$ *student exercise [this number should be your estimate of the dark current from the CCD Reduction Exercise Part 1]*,

which suggests that the dark noise should be $\sigma_{darkelectrons} = \sqrt{\lambda_{darkelectrons}} =$ *student exercise [answer this by filling out the cell below]*

In [10]:
```python
# student exercise
import numpy as np

mean_dark_current = 13.815112984837924 # copy this number from the previous noteb
theoretical_dark_current_sigma = np.sqrt(mean_dark_current)
theoretical_dark_current_sigma
```

Out[10]: 3.7168687069679933

We have direct measurements of the noise, though. We have measured $\sigma_{dark} =$ `mean_dark_noise` = *student exercise*

and $\sigma_{readnoise} =$ `mean_readnoise` = *student exercise*. We can then solve for the dark current noise:

$$\sigma_{darkelectrons} = \left(\sigma^2_{darkframe} - \sigma^2_{readnoise}\right)^{1/2}$$

In [11]:
```python
#STUDENT EXERCISE
mean_dark_noise =  10.870688076255952
mean_read_noise =  10.557842457849686
dark_current_sigma = np.sqrt(mean_dark_noise**2 - mean_read_noise**2)
dark_current_sigma # should be 2.5891739776383953
```

Out[11]: 2.5891739776383953

We can now see that the theoretical error from the observed dark current is higher than the measured value by:

In [12]:
```
theoretical_dark_current_sigma / dark_current_sigma
```

Out[12]:
```
1.4355422768299937
```

This difference occurs because of the CCD's *gain*.

Gain

The CCD's reading device (analog-to-digital converter, ADC) has some gain, which gives the number of ADUs per electron (analog-digital unit, or "arbitrary data unit").

$G = \frac{n_{ADU}}{n_{e^-}}$

We can do some math to figure out what the gain is, then.

For our detector, we have that the measured signal S_{ADU} is going to be higher than the number of electrons by G, i.e.:

$S_{ADU} = S_{electrons} G$

Since the number of electrons follows counting (Poisson) statistics, we know that the standard deviation in electrons is:

$\sigma_{electrons} = \sqrt{S_{electrons}}$

which means that we expect the noise in ADUs to be

$\sigma_{ADU} = \sigma_{electrons} G$

Proof

You can prove this by using the propagation of error formula, $\sigma_z^2 = (dz/dx)^2 \sigma_x^2$, for $z(x) = cx$.

Since $S_{ADU} = GS_{e^-}$, that means $dS_{ADU}/d_{e^-} = G$.

Therefore, the error in ADU is:

$$\sigma_{ADU}^2 = \left(\frac{dS_{ADU}}{dS_{e^-}}\right)^2 \sigma_{e^-}^2 = G^2 \sigma_{e^-}^2$$

or

$$\sigma_{ADU} = G\sigma_{e^-}$$

Find the Gain in terms of quantities we measured

We can replace $\sigma_{electrons}$ in the above equation with the square root of the number of electrons.

We then obtain a relation between the signal in ADUs S_{ADU}, the noise σ_{ADU}, and the gain G:

$$\sigma_{ADU} = \sigma_{electrons} G = S^{1/2}_{electrons} G = G\sqrt{\frac{S_{ADU}}{G}} = \sqrt{S_{ADU} G}$$

We can solve for the gain, since we can measure both S_{ADU} and σ_{ADU} directly:

$$G = \frac{\sigma^2_{ADU}}{S_{ADU}}$$

We calculated above the dark current in ADUs and the standard deviation of the dark current in ADUs. Put these together to compute the gain:

In [13]:
```
# student exercise
gain = dark_current_sigma**2 / 13.815112984837924
gain
```

Out[13]:
```
0.485252773092574
```

You should get a gain value of 0.485252773092574

That gain means that it takes approximately *two electrons* to produce one ADU.

The gain is also reported in the FITS header, so we can check our work:

In [21]:
```
from astropy.io import fits
header = fits.getheader("C:\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 1\\I
#header = fits.getheader(bias_001)
header['EGAIN']
```

Out[21]:
```
1.55
```

Our recovered gain value does not agree with the header value. We'll investigate this further later - maybe you will obtain a better estimate of the gain in the lab!