Realistic synthetic images

Due date: Feb 24, 2021

In the past two labs we generated a somewhat realistic synthetic lightcurve from a single object. This swept under the rug the whole aspect of imaging, but worry not, we're about to make that right! For all steps above, don't just write a script to do everything, write *functions* so that you can experiment with different parameters. Discuss carefully each of the effects and the impact they have on the image.

1. Generate a blank square image, say 2000×2000. Then add some read noise to it. Read noise has a gaussian distribution with standard deviation given by read noise in electrons (parameter) divided by gain in electrons/count (parameter).

2. Next, add bias. Parametrize uniformity across the image; also add hot and/or cold pixels, and bad rows and/or columns. Parametrize their numbers.

3. Now add dark current and sky background. Don't forget that dark current is Poissonian, and a function of exposure time, so the units are electrons per pixel per second. Also don't forget to account for gain (i.e. electron to count conversion). Sky background is similar, but the origin is external rather than due to the CCD.

4. Finally, add stars. We will be using 2-D gaussians for their shapes. Parametrize their numbers and brightness distributions. As before, don't forget about the gain.

5. So far all these have been additive quantities. But flat field is a *multiplicative* quantity, so we can apply it only after we've accounted for other sources of noise. Synthesize a flat field where you account for: (a) random per-pixel sensitivity, (b) variations due to dust on a CCD, (c) variations due to dust on the optical system, and (d) variations due to illumination. Then apply the flat field to your image.

6. *Extra credit:* We assumed above that stars have a 2-D Gaussian shape. In reality, the shapes are those of Airy disks. Explain what Airy disks are, what determines that response for stars, and quantify how well the 2-D Gaussians approximate Airy disks.