

# Lab 1: Computer Simulation of a Light Detector

## 1 Introduction

You will develop a computer simulation of photon detection by any digital detector of light, such as a *photomultiplier tube* (PMT) or a *CCD*. A constant source of light emits photons at unpredictable times. Only the average number of photons emitted per second is constant. During a time,  $\Delta t$ , the number of photons,  $N$ , emitted fluctuates about an average value by  $\sqrt{N}$ . Also electronic noise called *dark counts* add errors to measurements of light.

### 1.1 Schedule

Due in 1 week - Report due Tuesday, September 4.

### 1.2 Recommended Reading

- Tutorials on linux, L<sup>A</sup>T<sub>E</sub>X, and IDL: <http://ugastro.berkeley.edu/aquarius/> (when in rm 705), or outside use: <http://ugastro.berkeley.edu/optical>
- Chapter 11 in “Error Analysis” by Taylor
- LaTeX website: <http://www.giss.nasa.gov/latex/ltx-2.html>

## 2 Detector Simulation

### 2.1 Overview

You will simulate, on the computer, the unpredictable arrival of individual photons at a photon detector. Most light detectors detect photons by the photoelectric effect. An incoming photon hits a piece of metal, ejecting an electron which is sensed by a digital circuit and recorded as one “count”. In the lab next week, we will use an actual PMT light detector.

## 2.2 Poisson Statistics

During a tiny time interval (such as  $1\ \mu\text{sec}$ ), a steady light source will emit an individual photon with some probability, just as rain drops during a steady rain storm fall sporadically into a bucket. If  $N$  photons are expected to arrive, the actual number of photons that arrive may differ by about  $\pm\sqrt{N}$ . Such fluctuations in the number of photons are called “Poisson” statistics (or sometimes “photon statistics”).

According to Poisson statistics, if  $N$  events are expected to occur in a certain time interval, the standard deviation is  $\sigma = \sqrt{N}$ . Poisson statistics apply to many every-day processes such as the number of radioactive decays by an unstable isotope, the number of Earthquakes that hit Berkeley each year, and the number of wins by a bad baseball team in a season. Outside lab, read Chapter 11 in “Error Analysis” on the Poisson Distribution.

Each detected photon yields one count. But a light detector has a certain “time resolution”,  $t_{\text{res}}$ , a short interval of time that is typically  $10^{-6}$  sec during which it can sense a photon. During each  $t_{\text{res}}$ , there is some small probability  $p$  that a photon arrived. After a long *integration*, say 1 sec, the total number of photons that arrived is not a definite number, due to the probabilistic arrival a photon within each  $t_{\text{res}}$ . According to Poisson statistics, if  $N$  photons were expected to arrive on average, then the actual number of photons that arrive will differ with a standard deviation,  $\sigma = \sqrt{N}$ . You will simulate these fluctuations in the arrival of photons in your computer code, by letting the probability be  $p$  for arrival of a photon within each  $t_{\text{res}}$ .

## 2.3 Standard Deviation of the Mean

The more samples (integrations) you take the more accurately you can measure the average number of counts per integration. To illustrate this effect take data for 1 sec, and take 12 such *exposures*. Choose a fixed integration time, say 1 sec.

You will calculate the mean number of counts per sec from this set of 16 integrations, and also calculate the standard deviation,  $\sigma$ , of the 16 measurements, and also compute the uncertainty in the mean,  $\sigma_{\text{Mean}}$ .

Because of statistical fluctuations in the arrival of photons, the true number of counts per sec differs from the mean number of counts per sec *that you measure in your 16 trials*. That is, the true rate of photons is uncertain, and

the mean is only an approximate measure. Its uncertainty is the “standard deviation of the mean,” given by:  $\sigma_{\text{Mean}} = \frac{\sigma}{\sqrt{N_{\text{trials}}}}$ . This is a measure of the uncertainty in our knowledge of the average counts per sample.

### 3 Simulation of Photon-Counting using IDL

Your IDL computer program will simulate the detection of photons within each  $t_{\text{res}}$ , that occur within a total *integration time*,  $t_{\text{integ}}$ , typically many seconds long in astronomical observations. Your IDL program must determine how many photons were detected during that integration time.

The following ideas will guide you in writing the simulation in IDL.

- On the computer, use the *emacs* to open your new file that will contain the computer program:

```
%emacs simpmt.pro & ;Edit the file simpmt.pro (& gives the window back)
```

- In that file, the first line of IDL code should be “pro” followed by the name:

```
pro simpmt ;specifies an IDL program named simpmt
```

Make sure this file ends with the word END. The computer code itself goes between these lines.

- Set the variables for a time resolution of 1  $\mu\text{s}$  and the total integration time, 1 s, with these lines of code:

```
t_res = 1.e-6 ;time resolution of 10-6sec (put a comment after ‘;’)  
t_integ = 1.0 ;integration time in seconds
```

- Compute the number of time resolution elements  $N_{\text{el}}$  during the integration time.

```
Nel = t_integ/t_res
```

- Set the *average* number of photons per second, *photpersec*, that the PMT detects. You might try *photpersec* = 100. or any number!

- Determine the probability  $p$  of photon arrival during each time resolution element, based on  $t_{\text{res}}$  and  $\text{photpersec}$ .

```
p = ...[put your equation here]
```

- Put in your code a random number generator to simulate probabilistic photon arrival during each of the  $N$  time resolution elements. To generate an *array* of  $N_{\text{el}}$  random numbers uniformly distributed between 0–1, use this IDL command<sup>1</sup>:

```
rand_num_arr = randomu(seed,Nel) ;
```

- Construct a “FOR loop” that uses one element of the random-number array in each time resolution element, doing this  $N_{\text{el}}$  times. In each loop, your code should determine if a photon arrived in each time resolution element. One strategy you can adopt is that if the random number lies between 0 and  $p$  (probability of photon arrival), a photon arrived. You must tell the FOR loop how many time resolution elements to “loop” through.

```
for j=0L,Nel-1 do begin ;begin FOR loop (put the ‘L’ in for now)
```

Your code can ‘count’ the accumulated number of photons by creating a variable *nphot* that is increased *if* a photon occurred. Use an *if statement*:

```
if rand_num_arr(j) gt p then begin
nphot = nphot + 1
end
```

Finally, you must end the FOR loop:

```
endfor ;end FOR loop
```

- Have your program print the total number of detected photons during  $t_{\text{integ}}$ :

---

<sup>1</sup>In IDL, “seed” is itself a random number, made different by IDL each usage, to ensure that the  $N_{\text{el}}$  random numbers are different each execution of your code.

```
print, 'Number of detected photons:', nphot
```

To run this program in IDL, it must be *compiled* first. To do this, first get into IDL, by going to a new window and typing IDL. In that window, the prompt IDL> should appear. Now compile your program:

```
IDL> .com simpmt.pro
```

Now run your program by typing the program name at the IDL command line:

```
IDL>simpmt
```

The count results should appear on the screen.

### 3.1 Counting Photons Over Many Observations

For your report, you should simulate 20 observations of the light source. Let each integration be 1 second in duration. List the counts in each of the 20 integration times. Once you have the counts in an IDL array you can plot all of the counts using the IDL command:

```
IDL> plot, count_array,psym=2
```

Make sure that your plot has labels on the axes. A plot title is also helpful, i.e., “Counts in Integrations of 1 sec: 20 trials”.

You can get help on any topic in IDL help by typing a question mark “?” at the IDL command line. Look up “PLOT” this way, and figure out how to add a title and axes to your plot, and how to invoke different plot symbols (psym =).

Your report could address the following interesting issues.

1. List the counts in each of the 20 integration times.
2. Compute the mean, standard deviation, and uncertainty in the mean, of the 20 “observations”.
3. If only one integration had been taken, your uncertainty in the count rate would have to be based on Poisson statistics:  $\sigma = \sqrt{N_{\text{cts}}}$ . What is that uncertainty in the count rate?

4. After 20 observations, you can compare the mean number of counts per sec you collected to the *known, imposed* rate, namely *photpersec*. Do they agree within uncertainty — which uncertainty should you use here and why? Explain carefully in a few sentences what observed quantity is uncertain and by how much.
5. Is the uncertainty above different from the uncertainty you noted two questions above (for one integration). Explain which uncertainty is smaller and explain why.
6. As if explaining to a friend in a physics class, describe in a few sentences why your simulation does not simply yield the same number of counts as the input photon rate, *photpersec*, over and over again.

### 3.2 Dark Counts

Most light detectors produce *spurious* counts due to various electronic effects. Suppose your detection device has a dark rate of *darkpersec*. Detection of light is easiest if your adopted dark rate is *less* than the photon rate, say, 10 cts/s versus 100 cts/s.

Modify your code to incorporate (i.e. add) this dark count effect. Think carefully about how dark counts will be accumulated (within each  $t_{\text{res}}$ ).

Use your code (perhaps by occasionally turning off the light source) to measure the following quantities (as done at the telescope):

- Total counts from combined light and dark counts in 1 sec
- Just Dark counts in 1 sec (turn off the light)
- Net counts from light, i.e., total counts - dark counts . Think about reality here. You must take a separate measurement of dark counts, and use that average. (You can never know the dark count in an actual exposure of a star.)
- The expected uncertainty in the net counts from light in one second, due to propagation of uncertainty caused by Poisson fluctuations in the light and dark counts (see Appendix I).

Make sure your writeup addresses the questions above and the issues in Section 3.1 as they apply to a simulation that includes dark counts. Write

your report using L<sup>A</sup>T<sub>E</sub>X. It should have a section for a short introduction, a section describing the construction of the code you wrote, a section on the results (including a table), and a section with responses to the questions raised. Use a format of your choice within these constraints.