Jeffrey Gertler

Advisor: Prof. David Hogg

I. INTRODUCTION

A. Objective

Create a set of data that approximates the times of photon emissions caused by a given gamma $\frac{photons}{time}$ value over a given Δt. Also allow for data sets where the gamma changes with time.

II. CURRENT SOLUTION

A. Implementing a Changing Gamma

For a gamma that changes over time the main concept of the data generation is the same but broken into smaller sections. To find the ideal number of photons in a given Δt the program simply multiplied the gamma by the Δt which is the same as the integral of that linear function over Δt. Similarly, for any function of gamma over time the ideal photon count over that total time will be the integral of that function over the given Δt. A good approximation of the integral of a function is the Riemann sum of the function which breaks the integral into a number of small rectangles (this approximation is exact for a linear function).

With this in mind the problem of a changing gamma can easily be solved by breaking the Δt into many smaller time steps (currently the program uses time steps of 1 second). Over each smaller time step we find an experimental number of photons with the random.poisson() function and generate that number of photon times like before.

A significant advantage to this method is the ability to quickly adapt the program to non-linear functions. The only line of code that would need to be changed for a different function would be

```
idealPNum = int(gamma + (i+.5)*dGamma)
```

Instead, any function could be plugged into the integer typecast, with the correct command line arguments, and the program would correctly generate photon times.

III. GRAPHICAL OUTPUT

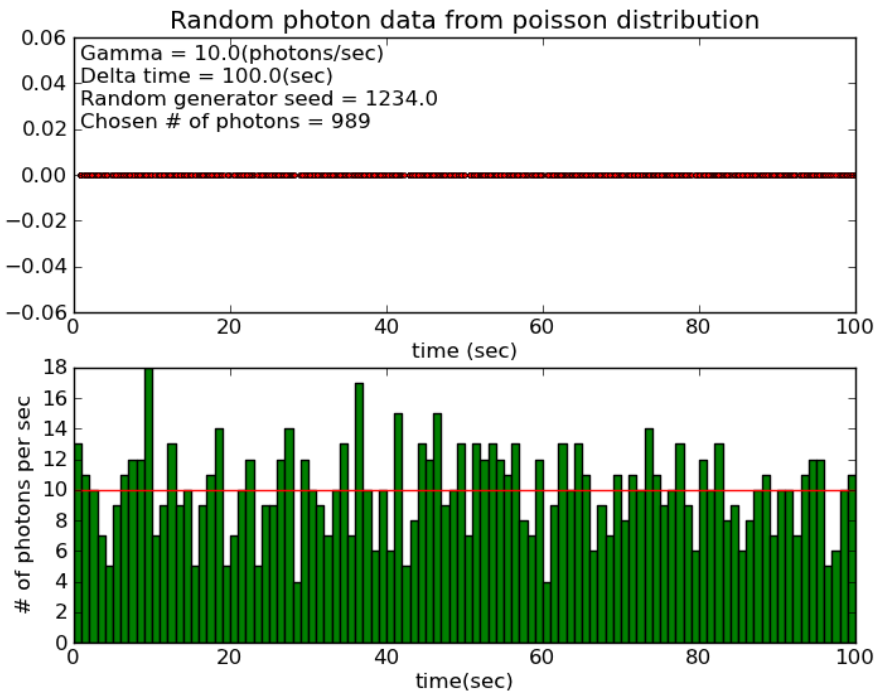

FIG. 1: Data output with a slope of 0

IV. CODE

```
for i in range(0, int(dt)):
    idealPNum = int(gamma + (i+.5)*dGamma)
    expPNum = np.random.poisson(idealPNum, 1)
    for j in range(0, expPNum):
        random.seed(seed*j*i)
        tData = np.append(tData, random.random() + i)
```

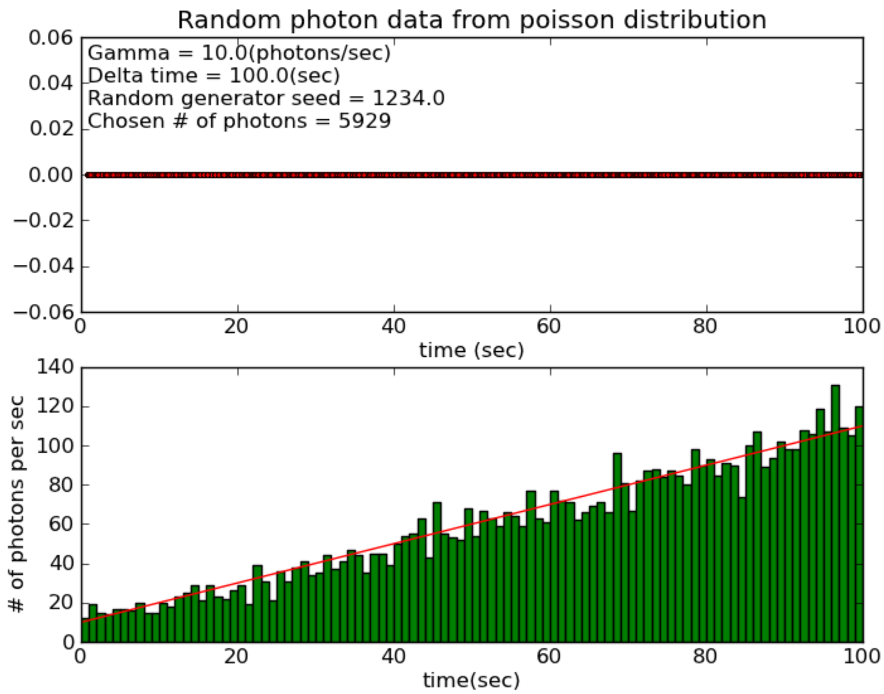

FIG. 2: Data output with a slope of 1

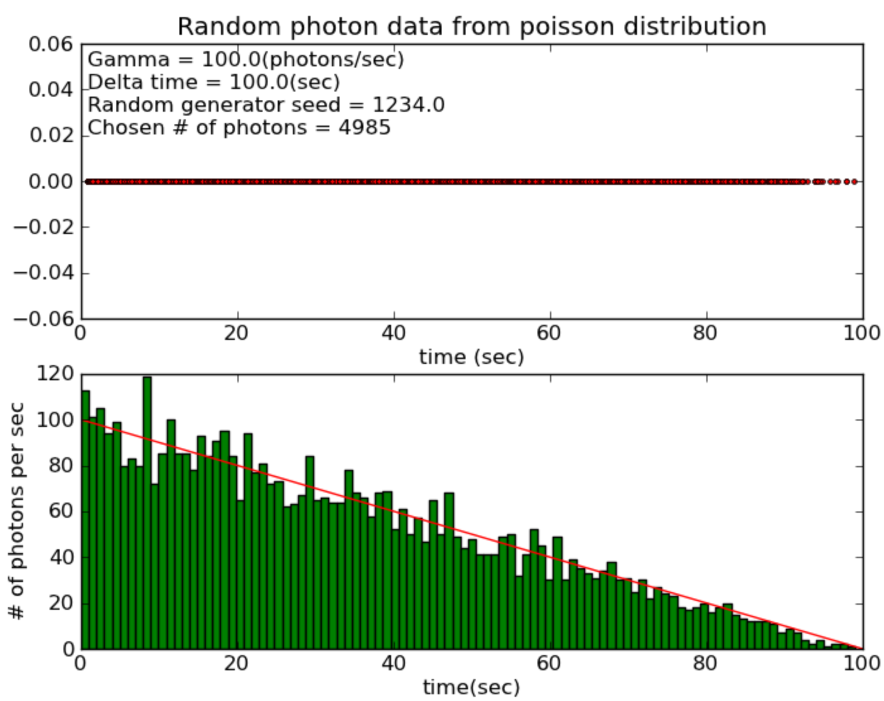

FIG. 3: Data output with a slope of -1