Introduction

The purpose of this lab and its associated mathematics is to gain some insight into the nature of random events and into the use of fast nuclear electronics instrumentation. You will also gain familiarity with MATLAB. This write-up assumes that you have completed the Introductory Poisson Statistics lab which should be in the same folder as this write-up. If you haven’t done this then start with it. Regardless of whether or not you are doing the experiment in that write-up, at this time read the Introduction in that write-up.

We expand a bit on the last paragraph from the Introduction of the Introductory experiment. As mentioned there, there are three main distribution functions of interest: the Binomial Distribution, the Poisson Distribution, and the Gaussian Distribution. All three distributions can be derived independently from assumptions made about the nature of the random process. The Poisson and Gaussian distributions can also be derived from the Binomial Distribution. This is very instructive because the Binomial Distribution is quite simple and intuitive to derive whereas the other two are not. As an aside, we note that the Gaussian and Poisson Distributions cannot be derived from each other. However, finding empirical relationships between them in certain limiting cases is an interesting project which can be undertaken later in this lab.

References

You should begin this experiment by getting started with the experiment. Later, perhaps when data is being collected, you can investigate the theory. The depth to which you do so will depend on the depth to which you are going to do this experiment. This, in turn, will depend on how long you are spending on this experiment, whether or not you are doing a formal report, and so on.

The Binomial Distribution, the Poisson Distribution, and the Gaussian Distribution are presented at a very simplistic level in Chapter Two of Rohlf. However, this is still a good first step and allows one to get going. The formulae are presented and the concepts of mean and standard deviation are discussed.

Boas (Mathematical Methods in the Physical Sciences) discusses all three distributions in the last chapter of her book. There is a thorough derivation of the Binomial Distribution and the Gaussian Distribution is derived from the large number limit of the Binomial Distribution. She then derives the Poisson Distribution independently. This discussion forms the basis of the presentation given in the Introductory Experiment.

Reif (Statistical and Thermal Physics) has a very thorough presentation of the Binomial Distribution and the Gaussian Distribution using the same general approach as Boas. He then leads you through a derivation of the Poisson Distribution from the Binomial Distribution in problem 1.9.

Heller’s very old two-volume set on probability theory (An Introduction to Probability Theory and its Applications) is a masterpiece. He introduces the Binomial and Poisson Distributions in Chapter VI and the Gaussian Distribution (called only the Normal Distribution when this book was published) in Chapter VII, obtaining the latter two from the former. In chapter VII he shows how a Poisson Distribution becomes a
(special case of a) Gaussian Distribution. This is an important derivation and related directly to this lab.

Using Radioactive Isotopes

The sole purpose of a radioactive source is that it produces random events for us. Read the sheet that outlines the precautions about working with radioisotopes. We are using a one microcurie cobalt-60 source. This level of radioactivity is well below that which requires licensing or the wearing of a film badge. The source is encased in plastic and is stored in a small heavy lead canister. These sources come through the regular mail. When the time comes, ask an instructor to show you how to handle the sample. A side product of this lab is to become familiar with radioactive samples and to put the level of radioactivity encountered in this lab into context, given the natural background levels of radioactivity and those commonly used in medicine. There is a portable, hand-held Geiger counter in the lab. Turn it on (and the sound) so you can measure the background radiation levels. This radioactivity comes from the Earth, from the stone used in the building, from the sun, and from the galactic center. Using this Geiger counter, you can investigate, at various distances, the activity of the sample being used. In particular, from where you will be sitting most of the time, how does its activity compare with background? There may be other experiments in the lab using radioactive samples. You can measure the level of radioactivity in the surrounding area as well.

Introductory Experiment, Definitions, and Theory

At this point you should read the entire Introductory Experiment write-up whether or not you are doing it. Make notes on the Poisson distribution and clearly note the definitions of all the parameters. You should be able to do all the calculations suggested in that write-up.

Advanced Experiments

We will now use more sophisticated nuclear physics apparatus to acquire and analyze much larger data sets.

If you are unfamiliar with the basic nuclear physics spectroscopy set-up (detector, preamp, DLA, scope, and MCA), obtain a Gamma Spectroscopy write-up and read to the point that you have a basic understanding what all this equipment is doing. In this experiment, the preamplifier, amplifier and high-voltage supply are all housed in the white box labeled “Ortec ACEMate” (also labeled “925-SCINT”). The number of decays per counting interval will be recorded by the Ortec EASY-MCS which is a small beige box that is connected to the laptop via USB.

An MCS is a Multi-Channel Scaler. “Scaler” simply means “counter.” So a MCS has many channels, each of which can accumulate counts for some fixed duration of time (called the “dwell time”). Each channel is like an independent “experiment.” So if you choose to accumulate counts for 100 ms and you have 2048 channels, the MCS will count decays for 100 ms and then put the total in the first channel. It then moves to the second channel and counts for 100 ms and stores the total in the second channel. It does this 2048 times. You can of course, choose a different dwell time (from 100 ns to 1,300 seconds per channel), or even a different number of channels (from 4 to 65,536, i.e. from $2^2$ to $2^{16}$). So you can accumulate a huge dataset. Using an MCS is conceptually equivalent, but much, much easier and more reliable than using a stopwatch and pencil and paper to count and
log events by hand. The details of how you configure the EASY-MCS to count will come later. For now, let’s ensure that we can see the signal output from the PMT.

ATTENTION: Please read through this entire list before doing any of it. Then follow these directions exactly to avoid damaging the PMT.

1. Put a 1 microCurie Co-60 source next to the detector, so that the source disk is concentric with the detector. If you are not familiar with what “the detector” is, by all means ask for help!
2. Ensure that the ACEMate is not powered (the red LED on the right-hand-side of the front panel should be off)
3. Ensure that the “BIAS 0-2kV” knob is at zero (turned fully counter-clockwise)
4. Ensure that the HV ON/OFF toggle switch is set to “OFF” (switch is down)
5. Ensure that the COARSE GAIN is set to 4 and the FINE GAIN is set to 5.0 (that’s a 5 on the outer dial and a 0 on the inner dial).
6. Next, ensure that the hardware interconnections are correct. You should find two cables coming out of the “S” and “HV” ports on the back (the base) of the PMT. The “HV” cable carries the high voltage bias from the ACEMate’s “HV OUT” to the PMT. The “S” cable carries the output current pulse signal (hence the “S”) from the PMT to the ACEMate’s “LINEAR IN” port. Also, ensure that there is a BNC Tee on Channel 1 of the oscilloscope. On one end of the Tee, a BNC cable should connect the “AMP OUT” on the back of the ACEMate to the scope. On the other end of the Tee a BNC cable passes the ACEMate output to the “IN” connector on the front panel of the EASY-MCS.
7. Turn on the ACEMate by flipping the black plastic toggle switch to the right.
8. The red LED just above the power switch should turn on, and you should see numbers on the LCD display.
9. Ensure that, to the right of the LCD display, only the red LED next to HV POS is on (i.e. that the HV is currently OFF and that the polarity is positive).
10. The “LL” LED under the LCD screen should be lit. Push the black rectangular button under the LCD screen. The “UL” LED is now lit. Repeat until the “KV” LED is on. This means that the numbers on the LCD screen indicate the output voltage of the high-voltage supply (“KV” stands for kilo-volts”).
11. Now we’ll turn on the HV supply for the PMT. On the ACEMate, flip up the HV ON/OFF toggle switch. You should see the “HV ON” LED next to the LCD screen come on. The LCD screen should read something like 0.06. Remember, this is 0.06 kV (a.k.a. 60 Volts).
12. Now, configure the oscilloscope to trigger on PMT pulses. Channel 1 should be set to 1V/div, DC coupling. The timebase should be at 2.5 us/div, and opt to trigger on Channel 1, rising edge. Set the trigger level to 240 mV. At this point, you should not expect to see any triggers (the PMT bias voltage is too low).
13. Now, slowly increase the PMT voltage to 0.2 kV (that’s only 1/5th of a turn of the BIAS knob). Do you see any events in the scope yet? Slowly increase to 0.3kV, then 0.4kV, then 0.5kV. Any events? By 0.6kV you should certainly see some event triggers on the scope. If not, please get an instructor right away.
14. If you do see triggers, slowly increase the voltage to the operating point of 0.75kV. Notice what happens to the amplitude of the signal on the oscilloscope.
as you increase the PMT bias voltage. Why does this happen? What information is contained in the amplitude of the signal?

Now, let’s take a short sample run, to get you started. Set the EASY-MCS dwell time to 1 milliseconds, and the number of channels to 10,000 (see the separate instruction sheet for details). How long do you expect it will take to acquire this dataset? Now collect a scan. To do so, click the “Clear” button on the right-side of the screen in the “Acquisition” area. Then click on the “Start” radio-button. You should see data filling up the channels. Move the marker around in the MCS-32 program and make a guess-timate of the mean number of counts. For example, how many counts are there in channel 125? How about in channel 9043?

Exporting the data for off-line analysis: To compare your data with various probability distributions (Gaussian, Poisson and Binomial), we need to export our data from the MCS-32 program in a format that can be read by other programs like MATLAB. To do so, from the MCS-32 toolbar, select “File à Export ASCII” and under “Save as Type” choose “ASCII Files”. Then give the output file a name (e.g. call it “1ms.dat” and put it on the Desktop).

You can then open the newly-created .dat file in (e.g.) Notepad (click on the Start Menu, then type “notepad” and hit enter). From within Notepad open the 1ms.dat file. The contents should look something like this:

```
0.0000 1
0.0010 3
0.0020 0
...
```

So the first column is the time since the start (i.e. the channel number times the dwell time, starting with channel number 0), and the second column is the number of counts measured in each channel.

You can now analyze the data in MATLAB (see the accompanying sheet “Tips for analyzing data in MATLAB”).

**The Poisson Distribution**

You can vary the dwell time and investigate how the distribution changes. Ideally, you want to investigate data sets where the mean is in the range of 0.01, 0.3, 2, 100, 1000, and 20,000. The latter will probably require an overnight run. Note that the standard deviations will be the square roots of the means. Can you visualize what these vastly different Poissons are going to look like?

For means between about 0.3 and about 5 or so you can compare the experimental data to a theoretical expression using the procedure outlined in the Introductory Experiment. For means very much less than one, you will just have a huge N for n = 0, and a tiny N for n = 1. (See the Introductory Experiment for definitions.) It doesn't make any sense to try to plot this but you can determine a theoretical prediction. For large means, the Poisson Distribution approached infinity divided by infinity for most n values and computers and calculators simple can't handle the calculations. One can try to work...
with logarithms but it is better to think about the Gaussian approximation of the Poisson Distribution.

**The Gaussian Distribution**

When the mean for a Poisson Distribution is large, the distribution becomes a Gaussian Distribution, albeit a special one. A general Gaussian Distribution is given by

\[
P(n) = \frac{1}{\sqrt{2\pi \sigma}} \exp\left[ -\frac{(n - <n>)^2}{2\sigma^2} \right]
\]

(14)

For a general Gaussian, the mean \(<n>\) and the standard deviation \(\sigma\) are independent constants. This, by itself, shows that a general Gaussian Distribution cannot be derived from a Poisson Distribution since \(\sigma = <n>^{1/2}\). Your data with high means can be fitted to a Gaussian Distribution with this constraint. In MATLAB do this Gaussian fit (i.e., with \(\sigma\) and \(<n>\) independent) and see if indeed the best fit first of all works reasonably well and secondly shows that \(\sigma = <n>^{1/2}\). Your data with means of about 1000 and 20,000 should both work but there is an interesting difference between the two. The former will show more scatter. Indeed, note that the values of \(N\) (not \(n\)) fluctuate by \(N^{1/2}\). (See the Introductory Experiment for definitions.) That is to say, we have "Poisson noise" on a Gaussian Distribution.

**The Binomial Distribution**

Unless he who composed this write-up is missing something, we're going to have to do about ten years of experiments to make sense out of this part of this proposed experiment. (That is, it's under development.)

The Poisson data with a mean of very much less than one (say, about 0.01) should be a string of zeroes with the occasional one (99 zeros for every one for a mean of 0.01). The Poisson plot should just be a huge \(N\) at \(n = 0\) and a tiny \(N\) at \(n = 1\).

We ought to be able to analyze this using the Binomial Distribution given by (purposely using none of the symbols used to date!)

\[
P(m) = \frac{L!}{m!(L-m)!} q^m (1-q)^{L-m}.
\]

(15)

where \(q\) is the probability of getting "an event" in a single trial (single time period?). \(P(m)\) is the probability of getting \(m\) events in \(L\) trials. For example, if you are flipping a coin, \(q = 1/2\) and \(P(m)\) is the probability of getting \(m\) heads (or tails) in \(L\) flips. But we digress.

Reif gives a very clear picture of how to get the Poisson Distribution from the Binomial Distribution in problem 1.9 at the end of chapter 1. Heller does the same thing in chapter VII. First, we need \(q <<1\) and \(m << L\). This means that in most trials, nothing happens. (Flipping a coin is not a Poisson process.) For our data with a very small mean, we have \(L\) trials (channels) with zeros in most trials and, occasionally, a one. There cannot be any two's or this won't work. If you have some two's, do it again either with a
smaller dwell time or move the source farther away. So, \( q = S/L \) where \( S \) is the number of
time we got a one. But isn't \( q = u \), also? So what? Who cares? What's the point?
Setting the Dwell Time and Number of Channels on the Ortec EASY-MCS

The Ortec EASY-MCS has no physical knobs to turn. Control of this device is done entirely in software, via the Ortec MCS-32 program. On the laptop, you will find the MCS-32 icon on the taskbar.

When you have started the MCS-32 program, you can configure the EASY-MCS hardware with two menus: “Pass Control” and “Input Control.” We’ll start with the latter.

From the top toolbar in MCS-32, select “Acquire → Input Control” (or press “Alt+i”). Ensure that the “Use Disc Input” button is selected. Then click “OK”.

Next, let’s enter the “Pass Control” menu to choose our dwell time and number of channels. From the top toolbar in MCS-32, select “Acquire → Pass Control” (or press “Alt+p”). At the bottom of this window, under the “Dwell” section, ensure that “Internal” is selected. Then choose a dwell time (e.g. to have a 500 microsecond dwell time, type the number 500 in the “Bin Width” field, and then select “us” from the dropdown list).

Ensure that “Pass Count Preset” is set to “1” (this means just count one time per channel).

Finally, select the number of channels you want. You should choose 2048 or larger. The more you choose, the better your data will match the limiting distribution. But there is a trade-off. For example, if you are aiming for a mean of 20,000 counts per channel, that requires a ~10 second dwell time. If you max out the number of channels (65,536) then your run would take a total of 655,360 seconds, or 7.5 days! Clearly that’s too long. So for the short dwell times, you can use the maximum number of channels. For longer dwell times, choose the number of channels so that the run can be completed overnight, or within 24 hours at most.
Page left blank intentionally
**Tips for analyzing data in MATLAB**

For now, you can just type the following commands directly into MATLAB. But once you get going, it’s much better/efficient to create a MATLAB .m script (e.g. myscript.m) to enter these commands into and then you can run and edit and re-run your code very easily.

Read in data from the ASCII file 1ms.dat into a single 2D array in memory

```matlab
>> allData = dlmread('1ms.dat');
```

Get the counts (all rows, second column of the 2D allData array).

Array indexing is (row, column) and “:” means “all entries.”

```matlab
>> counts = allData(:,2);
```

Plot the counts (this should be similar to what you see in the MCS-32 program). The only difference is that here, the x-axis is in channel number and in the MCS-32 program it is in unit of time.

```matlab
>> stairs(counts)
```

Now, we want to make a histogram of the counts data. To do so, we want to define the bin edges of the histogram and then let MATLAB do the counting (i.e. “how many times did we get zero counts? etc.).

```matlab
>> min(counts), max(counts)
ans =
    0
ans =
    9
```

So let’s make an array of numbers from min to max, in steps of one:

```matlab
>> binEdges = 0:1:9
binEdges =
    0  1  2  3  4  5  6  7  8  9
```
To generalize this (e.g. for other datasets), you can use:

```matlab
>> binSize = 2;
>> binEdges = min(counts):binSize:max(counts);
```
to go from min to max but have each bin 2 units wide.

OK, now we can ask MATLAB to tell us how many times we had zero counts, one counts, etc. To do this, we make use of a built-in function called `histc()` for “histogram count,” which returns the frequency of each count. For example:

```matlab
>> data = [21 23 23 20 22 22 22 20];
>> bins = 20:23;
>> histc(data, bins)
ans =
    2     1     3     2
```

In other words, in our input vector (`data`), the number 20 appeared twice, the number 21 appeared once, the number 22 appeared thrice and the number 23 appeared twice. We can now apply this to our dataset:

```matlab
>> freq = histc(counts, binEdges);
>> plot(binEdges, freq, 'k*')
>> xlabel('Number of Co-60 decays')
>> ylabel('Frequency')
```

To fit a Poisson probability distribution function to this data, use the built-in MATLAB function `fitdist()`. Note, `fitdist()` is in the the statistics toolbox of MATLAB.

```matlab
>> pd = fitdist(transpose(binEdges), 'Poisson', 'frequency', freq)
pd =
    poisson distribution
       lambda = 2.0137
```

So the best-fit mean of your data is 2.0137.
And then generate a visual representation of this best-fit by constructing the Poisson curve for a mean of 2.0137 (to be plotted below).

```matlab
>> mean = pd.Params(1);
>> yfit = sum(freq)*poisspdf(binEdges, mean);
```

Note, the term `sum(freq)` term in front of the `poisspdf()` function calculates the total number of events in your dataset (in this example, it is 10,000). This number is used to normalize (scale-up) the best-fit Poisson distribution so that the integral of the data and theory curve are the same (otherwise the `poisspdf()` function will return a Poisson distribution function which, by definition, has an integral of unity. Note, than in general, your histogram bins will not have a width of 1 and so the scale factor would need to be modified to be `sum(freq)*binWidth`. Also, if you instead want to fit a Gaussian function, you can repeat the above, but just change ‘Poisson’ to ‘Normal’ in the `fitdist()` call, and then use `normpdf()` instead of `poisspdf()` to calculate the Gaussian curve. Note that a normalized Gaussian function has two free parameters (the mean and standard deviation), whereas the normalized Poisson distribution has only one. So in `normpdf()` you will have to specify two parameters in addition to `binEdges`. The other caveat is that the `fitdist()` function will now return two parameters since the Gaussian has an independent mean and sigma. The mean is in `pd.Params(1)` and sigma is in `pd.Params(2)`.

To learn more about `normpdf()`, at the MATLAB prompt, you can type:

```matlab
>> help normpdf
```

OK, now do the actual plotting:

```matlab
>> errorbar(binEdges+0.5, freq, sqrt(freq),'k*')
>> hold on
>> stairs(binEdges, yfit,'k')
>> xlabel('Number of Co-60 decays')
>> ylabel('Frequency')
```

We have shifted the data points to be plotted in the middle of the bin `binEdges+0.5` in the `errorbar()` call. We have also introduced errorbars on our histogrammed data (the square-root of the number of occurrences of each count).