MSC Coverpage Assignment

Name of the Tutor:	Chris Pawley

Course code:	**PRA3500**
Course Title:	
Course Coordinator:	**Chris Pawley**
Tutorial Group Number:	
Student Name:	**Shania Mitra**
Student ID nr.:	**i6176442**
Academic Year:	**2019-2020**
Number of Words:	
Title Assignment:	**Lab Report 4 - A Comparative Analysis of Expected Particle Trajectories with Random Observed Particle Trajectories**
Pigeon Hole nr.:	

Lab Report 4 - A Comparative Analysis of Expected Particle Trajectories with Random Observed Particle Trajectories

Lab Report submitted by Shania Mitra

I. Introduction

Hypothesis testing is a statistical method whereby an analyst tests an assumption regarding a population parameter. In the case of particle physics, the population of particles are taken into account.

Various techniques can be carried out for the aforementioned test, one of them being the chi-square test. (Sullivan, n.d.).

Chi-square procedures involve minimizing a quantity called S in order to determine best estimates for certain function parameters, such as (for a straight line) a slope and an intercept. S is proportional to (or in some cases equal to) a statistical measure called $\chi 2$, or chi-square, a quantity commonly used to test whether any given data are well described by some hypothesized function. Such a determination is called a chi-square test for goodness of fit. ("Chi-Square: Testing for Goodness of Fit", n.d.).

If ν independent variables, x_i are each normally distributed with mean μ_i and variance σ^2_i, then the quantity known as $\chi 2$ is defined by Equation 1:

$$\chi^2 \equiv \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} + \cdots + \frac{(x_\nu-\mu_\nu)^2}{\sigma_\nu^2} = \sum_{i=1}^{\nu} \frac{(x_i-\mu_i)^2}{\sigma_i^2}$$

Given the random fluctuations of the values of x_i about their mean values μ_i, the terms in the sum will be of order 1. Thus, with the correct selection of the mean and variance, chi-square will be approximately equal to ν. If this is the case, it can be concluded that the data is well described by the values chosen for hypothesised function. ("Chi-Square: Testing for Goodness of Fit", n.d.).

For a linear function (ax +b), the chi-squared/S value has the expression:

$$S = \sum_i \frac{\left(x_i - a\,y_i - b\right)^2}{\sigma_i^2}$$

If the calculated value is much larger than ν, and the estimation for the variance remains true, it can be concluded that the data is not well described by the hypothesised function, ax+b. ("Chi-Square: Testing for Goodness of Fit", n.d.).

The number of degrees of freedom is $\nu = N - r$, where N is the set of independent variables, and r is the number of constraints present and the resulting $\chi 2$ sample will be one having ν (rather than N) degrees of freedom. ("Chi-Square: Testing for Goodness of Fit", n.d.). Thus, in this paper, 5 degrees of freedom were chosen, due to the 8 independent variables subtracted by the constraints of the 3 spatial directions.

The data analysed may come from physical observations, or from a data-generating process, such as a simulation.("How Hypothesis Testing Works", n.d.). In this paper, the randomised set of data was generated using Monte Carlo Techniques. Monte Carlo Analysis, at the simplest level, involves running various scenarios with different randomised input and summarising the overall distribution of results. (Sullivan, n.d.).

Within the LHCb and other detectors, there exist uni-directional planes, where hits of the particles can be recorded. These may be recorded on a finite resolution due to electrical components. As they fly through the detector, there may be statistical fluctuations and may not follow a predicted straight line trajectory. Hypothesis testing can be used to compare how much a particle deviates from an expected straight trajectory, and if the hypothesised straight line fit can be used to describe their trajectories. (Pawley, 2020).

The aim of this paper was to generate random trajectories of particles using a Monte Carlo Simulation Method and to include an approximate 10% statistical error in the code. Using this randomised data, the predicted trajectory of the particle was computed. The random trajectories were compared to these predicted trajectories, and it is hypothesised that there is no significant statistical difference between the linear fit function and the random trajectories of the particles.

II. Materials and Methodology

The software used for this project was Python 3.6. The programming simulation models were established as ".py" files.

Within the program, 2 functions were defined, the first being the setup of the overall simulation, returning arrays containing the x-pixels hit and the y-pixels hit.

The second of the functions, get_chi_2, was used to return the overall sum of the chi-squared values from the individual chi-squared values obtained from each of the 8 screens for a single particle

Finally, these two functions were then utilised and generated results in the form of a graph for a loop of 10 particles.

A more detailed description of the overall simulation is as described below:

2.1. Task 1: Setup

Initially, for the overall setup for the experiment, the number of screens was set to be 8, oriented along the z-axis. The distances of the screens along the z-axis were initialised in an array, with the first screen being at a distance of 200 away from the origin, and the subsequent screens being at a distance of 100 away from the adjacent screens.

The overall screen size was chosen to be a 1000, with the entire screen containing 80 pixels in total. The entire screen was then divided into coordinates ranging from -500 to +500, with the pixels then ranging from -40 to +40.

Subsequently, a maximum angle of the degree at which a photon can be shot was defined to be 15 degrees, and using this value, a random x-angle and y-angle of the particle trajectory was generated, between -15 degrees and +15 degrees, converted to radians.

Using the x-angles and y-angles generated, along with the tangent function, two arrays of the x coordinate hit and y coordinate hit on each screen was determined. Utilising these, two arrays of the pixels hit on the x and y axis were produced.

In order to add a small probability (10% maybe), randomly, that the particle is found in an adjacent pixel, a for loop was defined. It parses through hit_x_pixel and hit_y_pixel indices, and if the random number generated is less than or equal to 0.1 (10%), then the pixel at that index for both x and y pixel is shifted by -1, 0 or +1.

For this overall function, 2 arrays are returned, each 8 long, for x pixels for each screen and y pixels for each screen. However, as hit_x_pixel and hit_y_pixel return values between -40 to +40, the values need to be adjusted as indexing initiates from 0,0 and ends at 80,80. To adjust for this, 40 was added to both arrays, and these were returned. An overview of the process is shown in Figure 1 below:

```
# with 10% chance move the hit to adjacent pixel
   for i in range(len(hit_x_pixel)):
       if np.random.rand() <= 0.1:
           # print('test')
           hit_x_pixel[i]+=np.random.randint(3)-1
           hit_y_pixel[i]+=np.random.randint(3)-1
   return np.clip(hit_x_pixel+int(pixels/2),0,pixels-1),
np.clip(hit_y_pixel+int(pixels/2), 0,pixels-1)
```

Figure 1. Adding small probabilities and arrays for x-pixels and y-pixels hit returned.

2.2. Task 2: Chi-squared test

The particle was assumed to travel with a function of first degree polynomial, and so its trajectory was programmed to be linear. The z detection points were initialised to be the screen distances, with the x and y detection points defined to be the x and y pixels obtained, converted back to coordinates on the screen.

Line graphs of the actual measured locations for x and y coordinates were plotted on the y-axis, with the z-detection points on the x-axis. The line graph for x detection points was in blue, whereas the line graph for y-detection points was in green.

For the predicted x and y coordinates, primarily the polynomial coefficients for the 1st degree polynomial function describing the particle trajectory was obtained using np.polyfit. These values were initialised to be a and b for the function ax+b.

The array for the predicted xy coordinates was then generated, and plotted in orange as the line of best fit, using the code in Figure 2:

```
preds =
(np.array(a,ndmin=2).T*np.array(z_det,ndmin=2)+np.array(b,ndmin=2).T).T

   # plot best fit
   plt.plot(z_det, preds, color='orange')
```

Figure 2. Array for best fit line

Consequently, in order to acquire the chi-squared values, the variance of the dataset was chosen to be the width of one pixel. Using this value, Equation 1 was used to obtain an array of chi-squared values, using a for loop running through each index in the predicted coordinates array, and then adds calculated chi-squared values to retrieve the overall sum.

Using the for loop shown in Figure 3, this was then simulated for 10 particles and the sum of chi2 values for x and y respectively, were plotted on a histogram to observe the distribution.

```
chi_2_arr = []
for i in range(10):
   hit_x_pixel, hit_y_pixel = shoot_particle()
   chi_2_arr.append(get_chi_2(hit_x_pixel, hit_y_pixel))
   screens[:,hit_x_pixel, hit_y_pixel]+=1

x_chi2 = np.array(chi_2_arr).T[0]
y_chi2 = np.array(chi_2_arr).T[1]
plt.hist(x_chi2, alpha=0.5, color='blue', bins=25)
plt.hist(y_chi2, alpha=0.5, color='orange', bins=25)
```

Figure 3.

2.3. Task 3 - Generating New Particles

To generate 3 particles at the origin, the for loop mentioned in Figure 3 was altered to generate data for only 3 particles. Additionally, to ensure the particles were generated at the origin, in the following code in Figure 4, the x and y intercepts were set to 0.

```
#set intercept to 0 to make particle come from origin
   A = np.array([z_det, np.ones(len(z_det))]).T
   A.shape
   x_det.shape
   y_det.shape
   a, b = np.linalg.lstsq(A, x_det.T, rcond =None)[0]
   m, c = np.linalg.lstsq(A, y_det.T, rcond =None)[0]

   plt.plot(z_det, x_det.T)
   plt.plot(z_det, a*z_det)
   plt.plot(z_det, y_det.T)
   plt.plot(z_det,m*z_det)
```

Figure 4.

2.4. Task 4 - Degrees of Freedom

For task 4, the chi-squared values obtained were divided by the number of degrees of freedom chosen. In this experiment, 5 degrees of freedom were chosen, due to the 8 independent variables subtracted by the constraints of the 3 spatial directions.

III. Results and Discussion

3.1. Task 1: Setup

For task 1, the simulation was set up, using the function shoot_particles.

For the first particle, the X and Y detection points were graphed as shown in Figure 5.

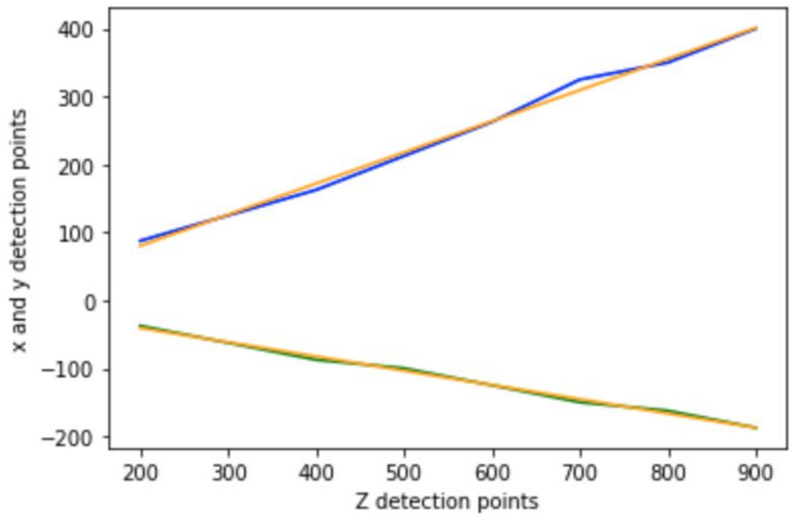

Figure 5. Graph of Z detection points vs the X and Y detection coordinates.

In the above figure, the particle followed an X trajectory as seen by the blue line. Due to a 10% statistical deviation from the mean, the trajectory of the particle along the x-axis is not a straight line. Similarly, the trajectory of the particle along the y axis is as shown by the green line and also contains deviations from the line of best fit, which is shown by the orange line for both spatial directions. The orange lines are the hypothesised linear function along which the particle is expected to travel in the x and y spatial direction.

In Figure 6 below, the detection points for the first particle on each screen was recorded and tabulated.

Z detection points	X detection Point for Particle 1	Y coordinate for Particle 1
200	25	0
300	25	-12.5
400	50	0
500	62.5	0
600	87.5	12.5
700	100	12.5
800	112.5	0
900	125	25

Figure 6. Table of Z, X and Y detection points for the first simulation of a particle.

As can be seen from Figure 5 and Figure 6, at further Z detection points, the absolute X and Y values increase, indicating a divergence in the trajectory for both X and Y values, moving away from the origin. This is as expected, as, if the detector shot out particles at (0,0,0), then as the particle moved away from the origin at a specific angle, then the absolute values of the X and Y coordinates would increase.

Furthermore, the pixel values for the first particle are recorded in Figure 7 below

Z detection point	X pixel	Y pixel
200	42	40
300	42	39
400	44	40
500	45	40
600	47	41
700	48	41
800	49	40
900	50	42

Figure 7. Pixel values for Particle 1 at each Z detection point

3.2. Task 2: Chi-squared test

For Task 2, the sum of chi squares for the X and Y coordinates were calculated and tabulated in Figure 8, for each particle simulation.

Particle Number	Sum of chi Square_ X	Sum of chi square_ Y
1	1.25	2.73
2	0.643	0.535
3	0.726	0.726
4	0.619	0
5	0	0.476

6	0.536	0.571
7	0.536	2.07
8	1.53	0.821
9	0.821	1.9
10	0.476	0.476

Figure 8. Sum of Chi-square for X and Y coordinates, for each particle simulation.

The sum of chi-squared values were then plotted in a histogram and shown below, to illustrate the approximate distribution of the chi-squared values.

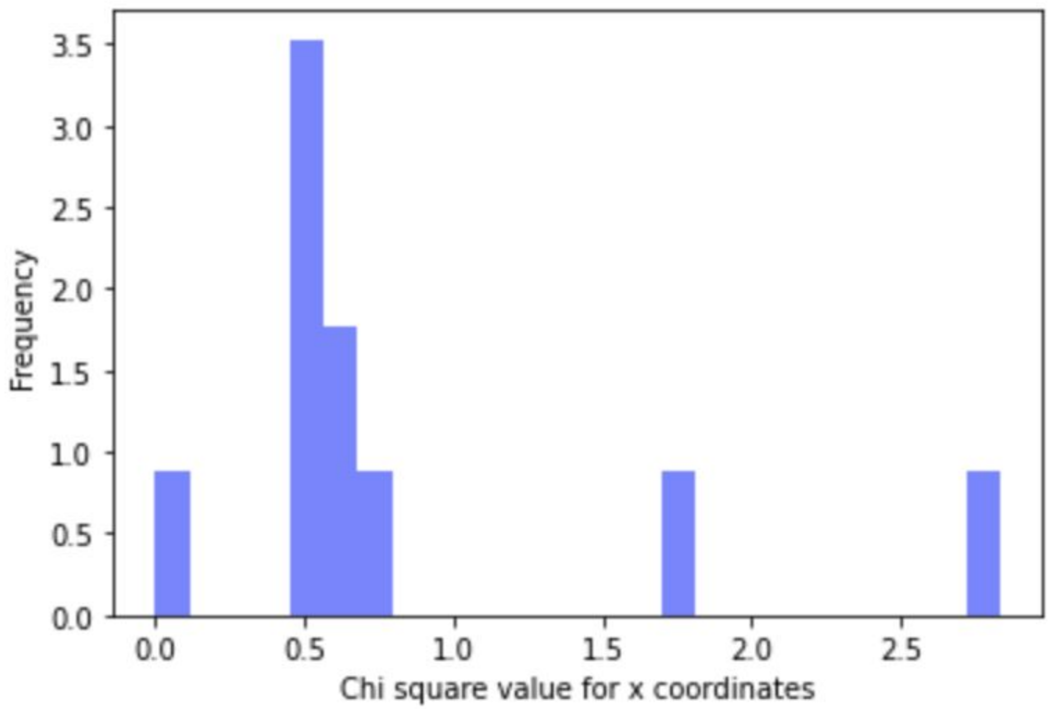

Figure 9. Histogram of distribution of Chi-Squared values for x-coordinates.

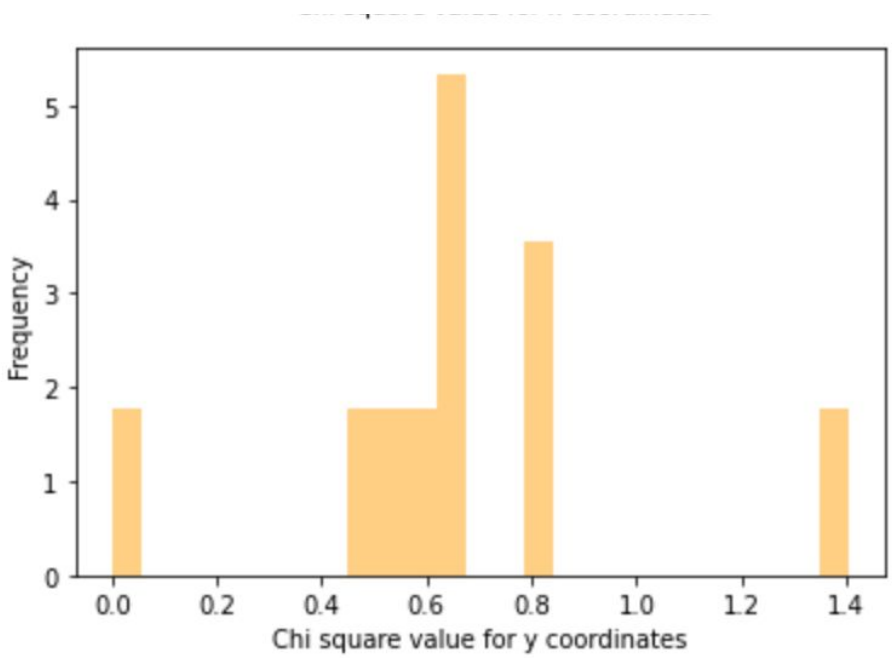

Figure 10. Histogram of distribution of Chi-Squared values for y-coordinates.

The above histograms and the frequencies of chi-squared values show a Gaussian-like distribution.

Using the critical values of a Chi-squared distribution ("1.3.6.7.4. Critical Values of the Chi-Square Distribution", 2020), the null hypothesis is accepted when the Chi-squared value, at 5 degrees freedom and probability less than the critical value of 0.95, is lower than the value present in the table. As the value is 11.07, the computed chi-squared values for each of the particles are lower and thus the null hypothesis is accepted, and there is no significant statistical difference between the expected trajectories and the randomly generated trajectories for each of the particles.

3.3. Task 3 - Generate 3 Particles from the origin and compute Chi-Squared values

For Task 3, particles were generated by setting the intercept of the linear function as 0, and simulating this for the 3 particles. The following graphs for the X and Y detection point were obtained:

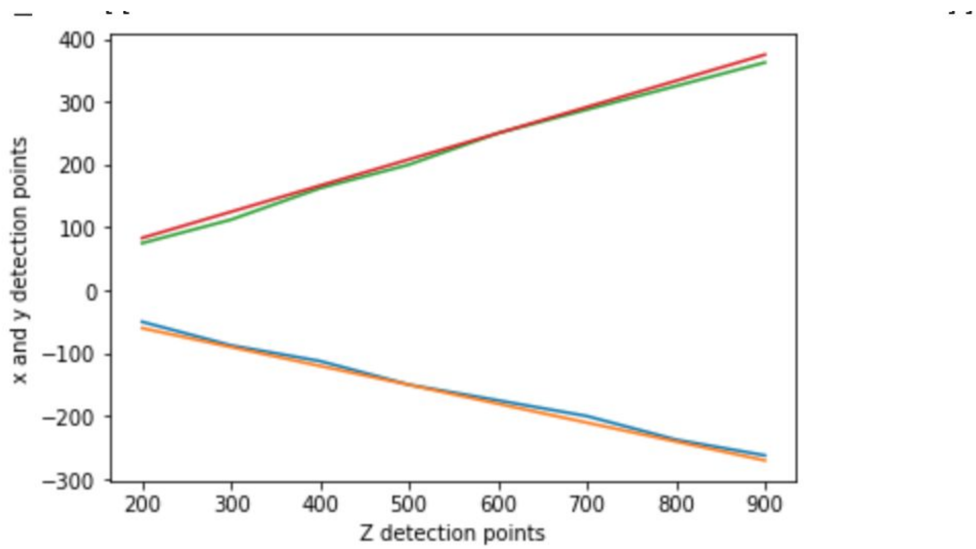

Figure 9. Graph for X and Y detection points for particle 1

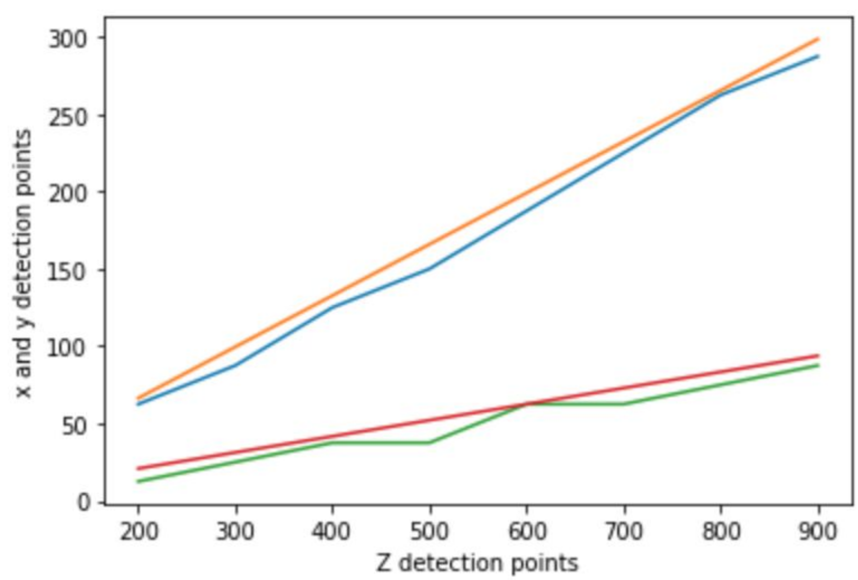

Figure 10. Graph for X and Y detection points for particle 2

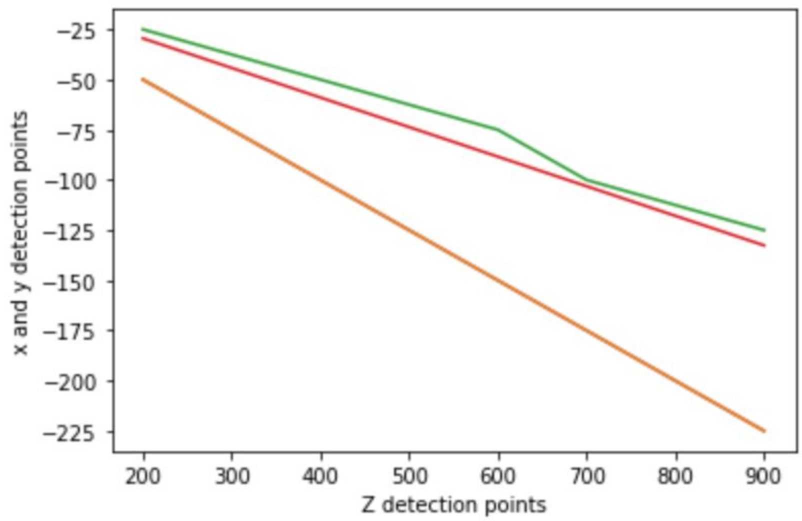

Figure 11. Graph for X and Y detection points for Particle 3.

As the x-axis of the graph does not initiate from 0, the particles do not appear to originate from the origin, however, an extrapolation of the line graphs would show the intercept values to be 0.

The Chi-Squared values for each particle in the X and Y spatial directions are as shown in Figure 12 below:

Particle Number	Sum of chi Square_ X	Sum of Chi Square_Y
1	5.09	1039
2	2.433	9059
3	0	239

Figure 12: Chi-squared values for each particle in X and Y spatial directions.

In this case, using the table of critical values for a Chi-squared distribution, it can be seen that the sum of chi-squared values of X are lower than 11.07, thus, the null hypothesis is accepted for the X coordinates, however, the Sum of chi-squared values of Y are much higher than the critical value. For Y coordinates, therefore, the null hypothesis is rejected, and the expected Y trajectory has a significant statistical difference to the actual Y-trajectory.

The sum of chi-squared values for Y, however, are extremely high, and indicate an error in its possible calculation. This could possibly be due to the low number of simulations leading to significant error, or due to an error in the simulation itself.

Plotting the histograms for the chi-squared values vs. the frequencies at which they appear, it can be duly noted that there is a clear absence of a Gaussian distribution, due to the aforementioned low number of simulations carried out.

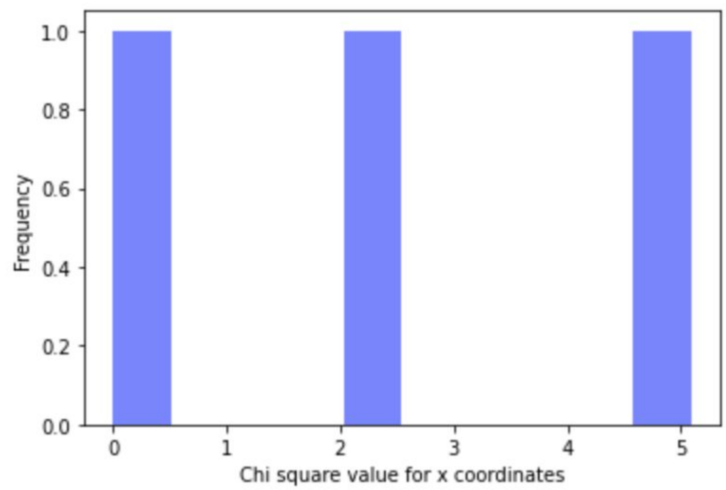

Figure 13. Histogram of distribution of Chi-Squared values for x-coordinates.

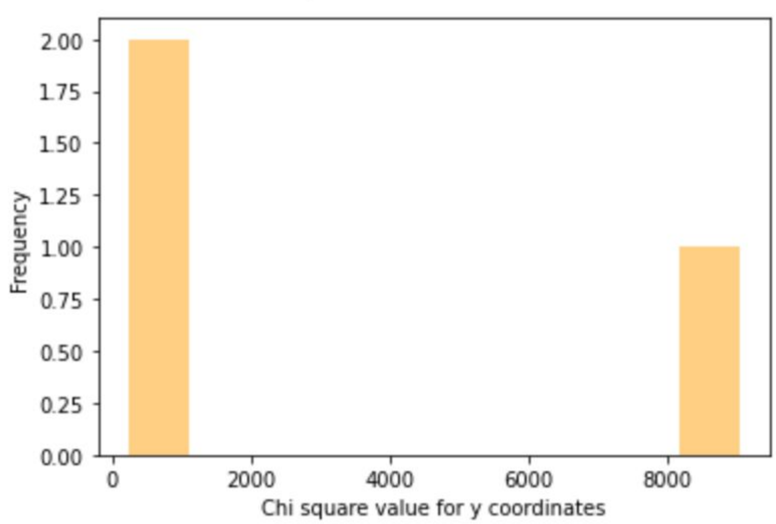

Figure 14. Histogram of distribution of Chi-Squared values for y-coordinates.

A further task could be to generate a fourth particle originating from a different source point. A possible approach to this problem would be to set the intercept to a differing value, instead of 0, such as 12, and to carry out the same procedure as above.

3.4. Task 4 - Taking into account the degrees of freedom in a Chi-Squared test

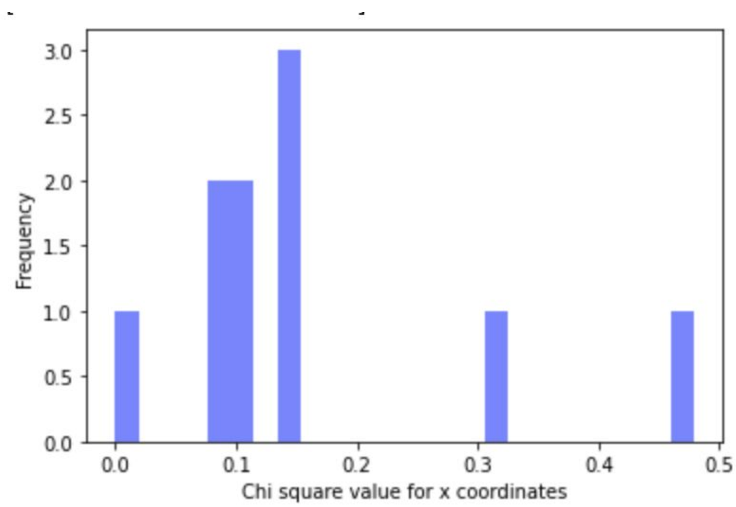

Figure 15. Histogram of distribution of Chi-Squared values for x-coordinates with Chi-Square values divided by the degrees of freedom

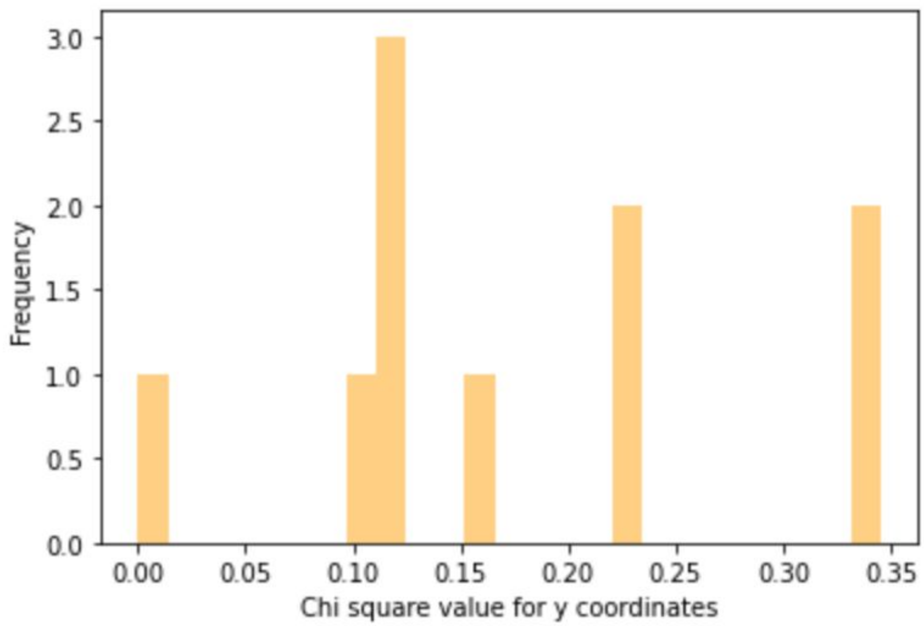

Figure 16. Histogram of distribution of Chi-Squared values for y-coordinates with Chi-Square values divided by the degrees of freedom

By dividing the Chi- Squared values by the degrees of freedom, the Chi-Squared distribution function itself does not change shape, it was found to merely rescale the values. However, it can be noted that the plotted histograms for the Chi-Squared values vs. frequency were found to have a slightly more Gaussian-like shape. The choice of degrees of freedom was 5, which by convention is the number of independent variables subtracted by the constraints in the spatial directions.

A possible alternative choice for degrees of freedom could have been 3N, where N is the number of mass points, where each mass point has a translational degree of freedom in the three spatial dimensions.

IV. Conclusion

In this experiment, the aim was to compare predicted trajectories of particles with the randomised trajectories generated by the Monte Carlo Simulation and to observe the presence of any significant statistical difference between the two.

Overall, this simulation was able to generate particles with various tracks and project them onto 8 screens present on the z-axis. Then, by carrying out Chi-Square tests, it was observed that for the simulation of 10 particles, there was no significant statistical deviation between the expected and the observed trajectories. Thus, the hypothesis was accepted.

For the simulation of three particles emitted from the origin, the hypothesis presented was accepted for the x coordinates due to the Chi-Squared values obtained being lower than the theoretical critical value. Conversely, for the y coordinates, the chi-squared values were found to exceed this critical value greatly, and thus, the hypothesis was rejected for the y coordinates.

Possible errors in this experiment could include the fact that the variance value was arbitrarily chosen to be the width of a single pixel. More specification about the particular type of detector may have been useful.

Further research could be carried out on a higher number of particles, varying choices of degrees or freedom or by using alternative methods of hypothesis testing.

V. References

Sullivan, L. (n.d.). Hypothesis Testing - Chi Squared Test. Retrieved May 17, 2020, from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_HypothesisTesting-ChiSquare/BS704_HypothesisTesting-ChiSquare_print.html

CHI-SQUARE: TESTING FOR GOODNESS OF FIT. Retrieved 17 May 2020, from http://maxwell.ucsc.edu/~drip/133/ch4.pdf

How Hypothesis Testing Works. Retrieved 17 May 2020, from https://www.investopedia.com/terms/h/hypothesistesting.asp

Pawley, C. (2020). *Data Analysis and Statistics for Physicists Week 4 – Hypothesis Testing*. Lecture, Maastricht Science Programme.

1.3.6.7.4. Critical Values of the Chi-Square Distribution. (2020). Retrieved 17 May 2020, from https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm