Timepix4 dead-hit percentage for AGOR cyclotron proton fluxes

Bachelor Thesis Physics

Author:
B.C.Honkoop

Examiners:
K.A.M. De Bruyn
J. Even

December 8, 2022

Abstract

The Timepix4 is a newly developed detector featuring improved performance compared to its predecessor Timepix3. This detector can be used to measure Time of Flight of protons, which are used in proton therapy of tumors and cancers. However, each pixel of the detector exhibits a dead-time immediately after a detection has been made, during which it cannot detect a passing proton. It is thereby of interest to find how many protons go undetected. This percentage of dead-hits depends on the beam intensity hitting the detector. As this research is being done considering the local AGOR cyclotron, only proton fluxes which this cyclotron can produce were considered. Using Python, it was found that for fluxes increasing from 10^4 protons/cm^{2}s to 10^8 protons/cm^{2}s, the percentage of dead-hits increases from 0% up to 1.82%.

Acknowledgements

When I was looking for a BSc project to work on, I wanted to work with Python to develop my coding skill. Kristof proposed this project, where the assignment was to build a simulation to answer a certain question. Building up this simulation from scratch gave me the freedom to decide my path towards the desired result. I am very grateful for this opportunity that allowed me to explore the fusion between Physics and Python. Kristof's pointers helped me focus on getting to the results, and his knowledge on the context behind the problem was a significant help to structure the simulation. For all these reasons along with his overall support throughout the project, I would like to thank my supervisor Kristof de Bruyn.

Contents

1 Introduction **4**

2 Experimental Setup **5**
- 2.1 Agor specifications . . . 5
- 2.2 Timepix4 deadtime . . . 6

3 Theoretical Expectations **8**
- 3.1 Misses between hits on target pixel . . . 8
- 3.2 Hits under limit . . . 10
- 3.3 Average hits per pixel . . . 12

4 Methodology **13**
- 4.1 Simulating incoming protons . . . 13
- 4.2 Pixel grid . . . 14
- 4.3 Misses between hits . . . 15
- 4.4 Hits under limit and Hits per cell . . . 15
- 4.5 Visualization & verification . . . 15
- 4.6 Varying proton flux . . . 16

5 Results **17**
- 5.1 Misses between hits . . . 17
- 5.2 Hits under limit . . . 18
- 5.3 Average hits per cell . . . 20

6 Discussion **20**
- 6.1 Average misses between hits for varying fluxes . . . 21
- 6.2 Hits under limit . . . 22
- 6.3 Hits per pixel . . . 22

7 Conclusion **23**

8 Further Research **23**
- 8.1 Challenging assumptions . . . 23
- 8.2 Multiple detection planes . . . 24
- 8.3 Birthday paradox . . . 24

9 Appendices **27**
- 9.1 Appendix A: Tables of results . . . 27
- 9.2 Appendix B: Link to Python code of simulation . . . 28

1 Introduction

The Timepix4 is the latest hybrid pixel detector readout development on behalf of the Medipix4 Collaboration. Hybrid pixel detectors consist of two layers; a sensor pixel layer and a readout chip layer. The Timepix4 acts as the readout layer, and is the state-of-the-art successor to the Timepix3 with faster readout performances and time resolutions [1] .

When a particle hits the sensor layer, it signals its corresponding Timepix readout chip. The Timepix4 translates the signals into two outputs; the time-of-arrival (ToA) and time-over-threshold (ToT). Using the ToA one can measure how long a particle takes to cover a certain distance, known as its time-of-flight (ToF) [2]. Each of these measurements can used for research, for example when studying high-energy physics problems, ToF mass spectrometry, or imaging applications.

The ToF measurements allow one to determine the kinetic energy of the detected particles. This is specifically applicable to proton therapy, which is an emerging high-energy irradiation technique for treating tumors and cancers. It has been shown that compared to photon therapy, proton therapy has significantly lower vital organ radiation exposure while controlling the disease as effectively as photon therapy [3]. This is because the penetration of protons is much more controllable than photons [4]. It is therefore useful to know what the kinetic energy distribution of the beam of protons is, which can be done with ToF measurements using the Timepix4. There is thus ample reason to expect the Timepix4 to be used more commonly in the future.

Before taking measurements of the Timepix4, it is important to understand its limits in terms of maximum beam intensity. This is because the Timepix4 is limited in its measurements due to the ToT readout. Every pixel within the Timepix4 only allow for new measurements once it has unloaded its charge to below its threshold, which is measured in the ToT. There is therefore a deadtime for each pixel; the time frame after a proton has activated the pixel in which a new proton will not be detected. From an incoming beam of protons, a certain percentage of protons will fall within these deadtimes depending on the intensity of the beam. The aim of this experiment is to thus find these dead-hit percentages of protons, specifically for the varying fluxes available from the AGOR (Accélérateur Groningen-ORsay) accelerator at PARTREC (Particle Therapy Research Center).

Though the results of this research can be used for any accelerator with similar proton fluxes, we specifically look at the case for when PARTREC procures a Timepix4 for experimental measurements on beams from the AGOR cyclotron. The beam intensities to be simulated are thereby taken from the AGOR specifications. Python is used for both simulating the incoming proton beam hitting a Timepix4 detector and analyzing the simulated data to get results. The simulations will be used to answer the following questions: How many of the incoming protons fall within the deadtime? To what extent do different fluxes affect the percent of deadhits? Will a single measurement plane be sufficient for reliable measurements?

2 Experimental Setup

In this section we will look at the details surrounding the experimental setup that will be used for the simulation. This experimental setup consists of a variable proton beam from AGOR hitting a typical Timepix4 detector.

2.1 Agor specifications

The sample of protons used in the experiment come from the AGOR cyclotron at PARTREC. Protons produced in the cyclotron are emitted through a foil and passed through a collimator before they go to the experiment, as shown in Figure 1. The field shape of the produced beam at the foil resembles a spreading 2D Gaussian, though the field homogeneity changes as only the center of the beam passes through the collimator. This collimator filters out a 1cm^2 section of the beam and aligns it by catching stray protons. For example, a large cluster of approximately 6×10^9 protons are sent through a collimator, filtering the beam down into a 1cm^2 beam of 10^8 protons. The AGOR accelerator outputs protons with fluxes between $10^4 - 10^8$ protons per cm^2 per second [5]. At a distance, the Gaussian spread is much larger than the size of the collimator hole, such that the distribution of the beam that passes through can be assumed to be uniformly distributed.

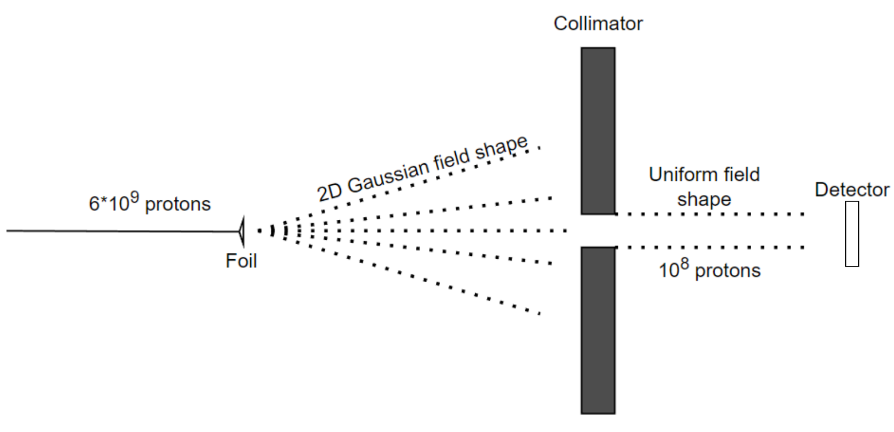

Figure 1: Diagram of AGOR setup. The beam from the foil spreads in 2D Gauss before being filtered and aligned in a collimator. The resultant beam has an area of 1cm^2 and is sent to the detector. The initial output from the foil may be changed to vary the proton output from the collimator.

Rather than sending the protons in a continuous beam, the cyclotron is instead sending particles through in slots that are being accelerated individually. AGOR has a fixed acceleration frequency of 60 MHz, meaning it outputs an acceleration slot every 16.7 ns. Different intensities of flux result in different numbers of protons per slot. For 10^8 protons you get 1.667, i.e. 1 or 2, protons in the same acceleration slot. Statistically however, a timeframe of thousands of nanoseconds will give the same result of 0.1 hits per nanosecond on average. Furthermore, there is no need to consider the problem of identifying protons that hit the detector simultaneously, as there would be at most 2 hits at once. Thus, in the simulations we will assume the protons arrive one-by-one at equal rates of 1 acceleration slot every 10ns.

2.2 Timepix4 deadtime

The Timepix4 was designed to connect to a sensor composed o 448x512 pixels, covering an area of 30.0x24.7mm^2 [6] as shown in Figure 2. Each sensor pixel that is connected to a readout covers a square area of 55x55µm^2. These pixels are made to measure the time-of-arrival (ToA) and time-over-threshold (ToT) of a particle that has hit the pixel. The ToA gives the time at which a particle hits a pixel, which is used for time-of-flight measurements. The ToT gives the time for which the pixel is ionized past a threshold of 800e, and is used to determine charges of incoming particles or precisely determine where a particle hits if it triggers multiple pixels.

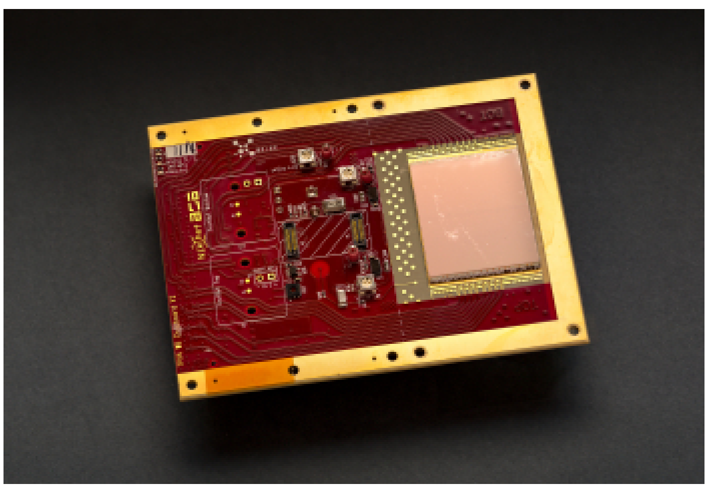

Figure 2: The metallic 3cm×2.4cm plate on the right is the Timepix4 hybrid detector, shown mounted on a Nikhef chip carrier board.

Each pixel from the device used for time-of-flight measurements consists of a sensor bound to a readout chip. The readout chip is the Timepix4, which is an ASIC (Application-specific integrated circuit) that translates the pulses from the sensor into digital ToA and ToT values. The sensor consists of semiconductor chips that act as diodes. Incoming radiation ionizes the detector material, setting charge carriers free between the diodes. These freed electrons and electron holes produce the signal sent to the readout chip depending on the energy of the radiation.

During the period in which the electrons and holes are released, a new incoming proton would have a decreased ionization effect as the sensor pixels still contain charge. During the time in which a pixel is charged over the threshold, new incoming protons are therefore not detected. Only after the silicon diodes have released charge to below the threshold can the pixel detect new particles. Thus, it takes time to recharge the pixel and stabilize its sensitivity towards new detections. The ToT is therefore equal to the deadtime during which the pixel cannot detect a passing proton.

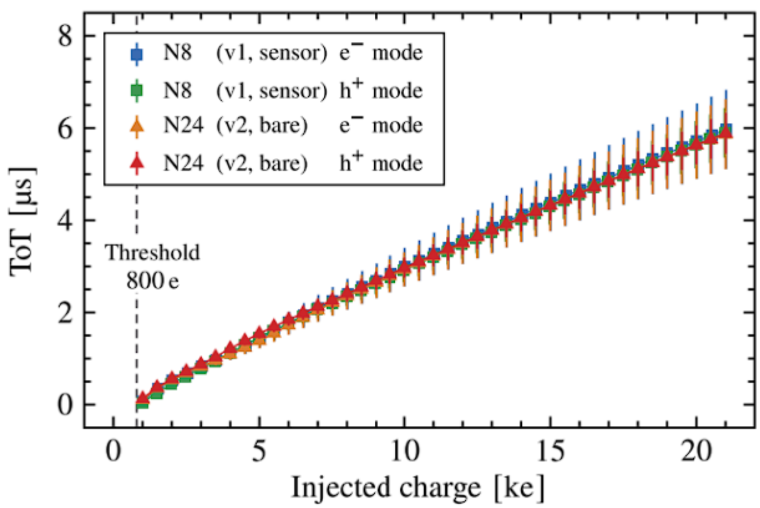

Figure 3: Time over Threshold (ToT) of a pixel as a function of charge injected into pre-amp. Taken from study from Nikhef. [7]

Figure 3, taken from a study from Nikhef, describes the ToT as a function of the charge injected into the preamp [7]. The ToT of a pixel ranges from 0-6µs depending on the injected charge. To fully encompass the possible hits within this time frame, the maximum ToT of 6µs was taken as standard for the deadtime of an individual pixel.

The assumption was made previously that the protons are arriving one-by-one with equal temporal spacing due to the 10ns acceleration slots. Thereby time may be measured based on the number of hits. From the background theory, a deadtime was taken as 6µs. Considering for example 10^8 protons/cm^{2}s, which is equivalent to 10^2 protons/cm^{2}µs, 6µs would encompass 600 protons hitting the detector. Thus, the frequency of hits on a pixel can be found by measuring the number of protons that missed the target pixel before a proton finally hits the target pixel. This will be referred to as the misses between hits, and acts as a direct indicator for the time it takes for a proton to hit the same pixel.

The number of pixels that cover the 100mm^2 square beam coming out of the collimator may be found using a simple calculation. Given that each pixel is a square of 55x55µm^2, we can directly calculate how many pixels fit within. 100mm=10,000µm, so each side fits 10,000/55 = 181.8 pixels. Since we want to be using the entirety of a pixel for detection as opposed to working with fractions, we will be taking the length of the collimator area to be 10.01mm such that 182 pixels fit exactly within. The pixel grid is thus 182x182 pixels fitting within 1cm^2, leading to a pixel number of 33,124.

3 Theoretical Expectations

3.1 Misses between hits on target pixel

To interpret the frequency of coincident proton hits on the target pixel, the number misses between the number of hits on target is measured. To understand what may be expected intuitively, it is useful to look at a simple model using a 6-sided die. To mirror what is happening with the protons, the die is first rolled to determine the target number from 1-6. Consequent rolls are then considered a 'hit' if they land on the target number, or a 'miss' if they land on any of the other numbers.

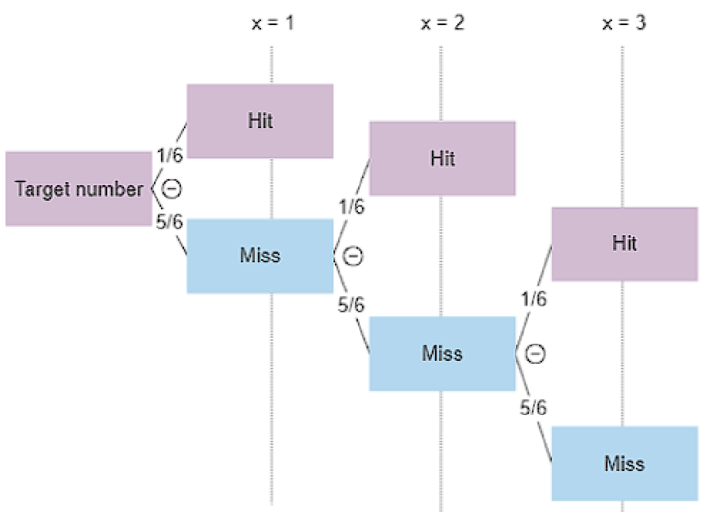

Figure 4: Probability tree for misses between hits. Based on 6-sided die with x amount of rolls.

The important value here is the probability of a hit within x amounts of hits. This probability predicts the number of misses between hits and is displayed in Figure 4. After the first roll of the die, you have a $\frac{1}{6}$ chance of rolling the target number and a $\frac{5}{6}$ chance of rolling any other number. In case of a hit, the counter resets such that the next roll is again the first one at (x = 1). In case of a miss, the miss is added to a counter which counts the number of misses before a hit and the next die is rolled. A hit at (x = 2) can therefore only happen after there has been a miss at (x = 1). The probability of a hit at (x = 2) is thus equal to the product of the miss probability and the hit probability,

$$p_{hit}(x=2) = p_{miss} \times p_{hit}(x=1) = \frac{5}{6} \times \frac{1}{6} = \frac{5}{30} = 0.167 \tag{1}$$

Which is slightly lower probability than a hit after the first roll. Considering (x = 3), a hit can only be made after 2 consecutive misses. The probability is thus

$$p_{hit}(x=3) = \frac{5}{6} \times \frac{5}{6} \times \frac{1}{6} = \frac{25}{216} = 0.116 \tag{2}$$

Where a pattern can be seen of

$$p_{hit}(x=x) = (\frac{5}{6})^{x-1} \times \frac{1}{6} \tag{3}$$

Or generally for any number of faces n;

$$p_{hit}(x) = (\frac{n-1}{n})^{x-1}\frac{1}{n} \tag{4}$$

This equation can be rewritten, considering that $e^x = 1 + x$ for $x << 1$. In this case, $x = -\frac{1}{n} = 3.02 \times 10^{-5} << 1$.

$$p_{hit}(x) = (\frac{n-1}{n})^{x-1} \times \frac{1}{n} = (e^{-\frac{1}{n}})^{x-1}\frac{1}{n} = \frac{1}{n}e^{-\frac{x-1}{n}} \tag{5}$$

This can be related directly to a gamma distribution. The gamma distribution describes a probability density function (pdf) depending on a shape parameter k and a scale parameter θ [8].

$$f(x) = \frac{1}{\Gamma(k)\theta^k}x^{k-1}e^{-\frac{x}{\theta}} \tag{6}$$

For a shape parameter of $k = 1$, the formula becomes

$$f(x) = \frac{1}{\Gamma(1)\theta^1}x^{1-1}e^{-\frac{x}{\theta}} = \frac{1}{\theta}e^{-\frac{x}{\theta}} \tag{7}$$

Thus, our equation can be equated to the gamma distribution pdf if we take the first roll as x = 0;

$$P_{hit}(x) = \frac{n-1}{n}^x\frac{1}{n} = \frac{1}{n}e^{-\frac{x}{n}} \tag{8}$$

In summary, we have a probability density function which can be used to calculate the probability of a hit after any number of misses x. We can thus expect a graph for the measured number of misses between hits to follow this probability density function. The graph for equation $P_{hit}(x)$ is shown in Figure 5.

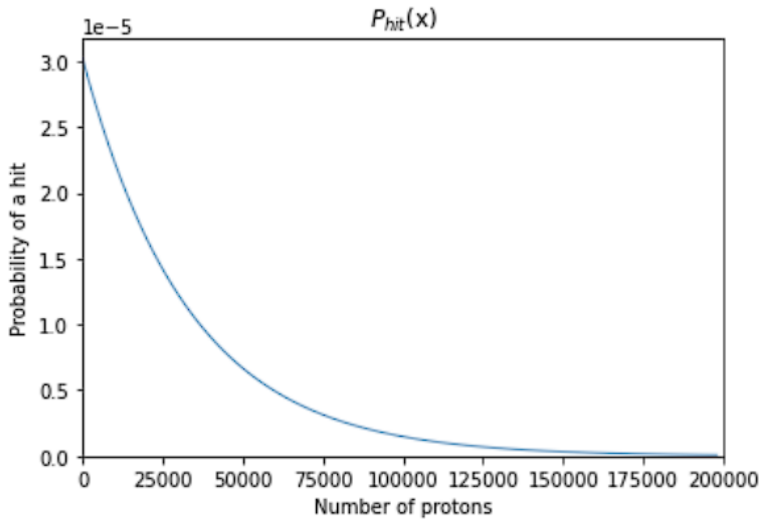

Figure 5: Probability density function from equation 8, giving the probability of a hit after x misses. This follows a gamma distribution of shape parameter k=1 and scale parameter $n = 33,124$.

For a gamma distribution, the mean is given by its shape parameter multiplied by its scale parameter which in our case results in $k = kn = 1 \times 33,124 = 33,124$. The expected value for average number of misses between hits is thereby equivalent to the number of pixels. This makes sense, considering that the protons that passed through the collimator were uniformly distributed such that each pixel has an equal probability of getting hit. On average, each pixel will have 1 hit if 33,124 protons are sent through. Thus the average number of misses one can expect before a hit is equal to 33,124.

This expected number of misses between hits depends purely on the number of pixels, and is independent of the number of protons that hit. It can therefore be expected to remain constant at 33,124 for varying proton fluxes.

3.2 Hits under limit

The hits under limit refer to the protons that landed on a target pixel within the deadtime caused by a previous hit. This is the main objective of this research, to find out how many protons hit a pixel within its deadtime. The number of hits under the limit will be counted, which is equivalent to the number of dead-hits. This can then be displayed in terms of percentage of hits that fall under limit. The percentage of hits under limit will be used interchangeably with the dead-hit percentage. As done previously, we can translate the deadtimes to dead-hit ranges. For a flux of 10^8 protons/cm^{2}s with a dead-hit range of 600 misses, the expected percentage of protons within the limit is calculated to be

Proton flux ($s^{-1}cm^{-2}$)	Deadtime limit (misses before hit)	Hits under limit (%)
$10^{8.5}$	1898	5.57
10^{8}	600	1.79
$10^{7.5}$	189	0.57
10^{7}	60	0.18
$10^{6.5}$	19	0.06
10^{6}	6	0.018
$10^{5.5}$	1.9	0.006
10^{5}	0.6	0.002
$10^{4.5}$	0.2	0.0006
10^{4}	0.1	0.0002

Table 1: Expected number of hits under deadtime limit for varying proton fluxes

1.79% as shown in equation 9. This dead-hit range is thus the limit for which a hit is considered to be within the deadtime.

The probability of a hit at x number of misses is given in equation $P_{hit}(x)$. To find the probability of a hit before x number of misses, one can integrate the probability density function from 0 to x. For 108 protons per second per cm^2 with a limit of $l = 600$ misses;

$$\begin{aligned} Q_{hit}(l) = \int_0^l P_{hit}(x)dx &= [-e^{-\frac{x}{n}}]_0^l \\ &= -e^{-\frac{l}{n}} + e^{-\frac{0}{n}} \\ &= 1 - e^{-\frac{600}{33124}} = 0.0179 \end{aligned} \tag{9}$$

As different fluxes yield different limits for their deadtimes, the expected percentage of hits under limit depends on the value for proton flux. Table 1 below shows the expected percentage of deadhits for the varying fluxes ranging from 10^4 to 10^8. To get additional datapoints at an equal spread on a logarithmic scales, intermediate values of $10^{x.5}$ were used.

A special case presents itself for the fluxes of $10^4 - 10^5$. Though an expected value was calculated using equation 9, an expected value cannot be measured in the simulation. For $10^4 - 10^5$ protons, the number of protons per microsecond is well below one. This means that the 6us deadtime translates to a deadtime limit of below 1. This cannot be simulated, as the first miss would already exceed the deadtime. Due to the assumption of continuous, evenly spread protons, no protons will ever fall within the deadtime. Therefore values for 10^4, $10^{4.5}$, and 10^5 will not be considered in the results for hits under limit. In the table, a value for $10^{8.5}$ protons/cm^{2}s is shown to see how the the percentage of hits under limit increases substantially for larger fluxes. However, as the AGOR accelerator can output a maximum of 10^8 protons/cm^{2}s, this data point will not be used in the results.

3.3 Average hits per pixel

For the hits under limit, the simulation will count how many hits fall under the limit for a certain count of protons. To find the percentage of hits that fell under the limit, we cannot take the hits under limit and divide them by the total number of protons. This is because the simulation is run for a single target pixel, which due to uniformity should show the same result for any pixel. Every pixel measures its own protons and hits under limit, and thus the total sample size for a single pixel is the number of hits on that pixel rather than the total number of hits in the beam. The percentage of hits that fall under a limit for any pixel is thus the number of hits that fell within the deadtime of that pixel divided by the number of hits per pixel. It is therefore necessary to know the average number of hits per pixel.

The expected number of hits per pixel is straightforward due to the uniform spread of protons, and depends on the proton flux and number of pixels. For example,

$$\text{protons per pixel} = \frac{\text{total protons}}{\text{total pixels}} = \frac{10^8}{33,124} = 3018.96 \tag{10}$$

As the number of pixels is fixed at 33,124, the average number of hits per pixel depends only on the flux. For a graph picturing the hits per pixel versus the proton flux, one can expect a straight-line as they are directly proportional. The gradient of this line would be $\frac{1}{33{,}124} = 3.019 \times 10^{-5}$, as for every 33,124 protons the average number of hits per pixel would increase by 1.

4 Methodology

4.1 Simulating incoming protons

Python will be used to simulate a beam of protons hitting the Timepix4 detector. The beam exiting the accelerator is passed through a 1cm^2 collimator before the 10^8 protons hit the detector. As only a fraction of the center of the beam passes through the collimator which consequently works to align angled protons, the field shape of the beam was taken as uniformly distributed. Thus, the sample of protons generated using Python were randomly and uniformly spread within a 100mm^2 square of 10×10mm^2. The exact measurements used were sides of 10.01mm, as explained in Section 2.

The code to generate this sample was straight-forward. As the width of a side is 10.01mm, a random value between [-5.005mm, 5.005mm] is chosen for an x value and for a y value. This is then repeated to get coordinates for every proton. Figure 6 visualizes a generated proton sample. The blue rectangle indicates the 24.7mm x 30.0mm area of the Timepix4, whereas the red square indicates the 1cm^2 shadow of the collimator through which the protons pass. As this figure is only for visualization, a sample of 1000 protons was used since a full sample of 10^7 protons overcrowded the plot.

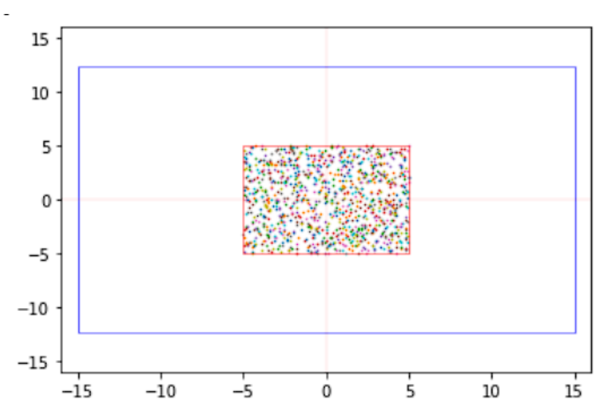

Figure 6: Visualization of 1000 protons hitting Timepix4 detector. The blue rectangle outlines the detector surface, while the red rectangle outlines the beam coming through the collimator.

To verify the uniform distribution, a similar graph is plotted with a larger sample size and different visualization. Figure 7 visualizes a sample of 1000 protons on the left, and a sample of 10^8 protons on the right. In each graph, the large colorful square represents the 1cm^2 beam which has passed through the collimator. The imaged 1cm^2 is divided into 50x50 bins (not representative of the Timepix's 182x182 pixels), where darker colors represent less hits while brighter colors represent denser hits.

The layered bar charts on the top and right indicate the distribution of protons across the respective x and y axes. The small sample simulation shows

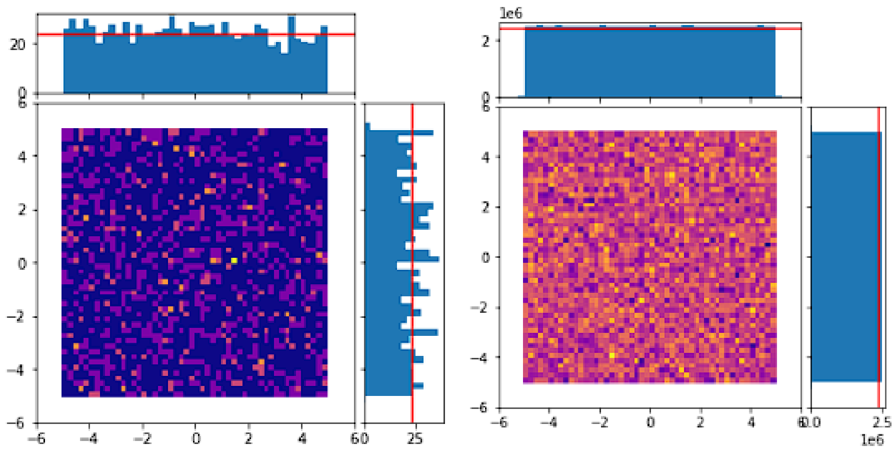

Figure 7: Field shape of incoming protons, for 10^3 protons (left) and 10^8 protons (right)

the randomness of the distribution by the deviation from the red line in the bar charts. This is because of the small sample size over a large set of outcomes. For a massive sample size, the expected outcome dominates divergent outcomes and results in a visibly more uniform distribution as intended. However, the 10^8 distribution still shows differences in density across pixels. Different chosen target pixels thus yield different number of hits, albeit averaging out to the same.

4.2 Pixel grid

After generating a sample of protons, they need to be tested against the grid of pixels to measure how often certain pixels get hit. Thus, it is necessary to first define the pixel grid. As mentioned in Section 2, each pixel in the grid covers an area of 55x55µm. This means that a square area of 10.01 x 10.01 mm^2 will perfectly fit a pixel grid of 182x182. Dividing the area into pixels is similar to dividing an area into bins, where each bin has a lower and upper limit. Each consecutive bin has a lower bound equal to the upper bound of the previous bin, where specifically the lower bound is inclusive and the upper bound is exclusive. The code therefore divides the width of the beam (10.01mm) by 182 to get the length of an individual bin. It then creates the bins starting from -5.005mm and adding the bin length to get the upper bound. In the end, the code outputs a list of 182 [min,max] values which each represent a bin. As the widths and lengths are equal for both the beam and the pixels, the list of bins for x-values is equivalent to the list of bins for y-values. Thus, any specific pixel is defined by an x-value bin with an upper and lower x-bound and a y-value bin with an upper and lower y-bound.

4.3 Misses between hits

Once the pixels have been defined, the generated sample of protons are compared to the bins to see where they fit. However, finding every pixel which has been hit and storing the data for 10^8 protons/cm^{2}s and 3×10^5 pixels takes an excessive amount of processing time. Thus, we instead look at a single specific target pixel and analyze how the 10^8 protons/cm^{2}s landed. A random pixel is selected during each run, giving us a specific x-value bin and y-value bin. Every proton's coordinates are then compared to the bounds of these target bins to check if they fall within, either registering as an 'hit' or a 'miss'. A hit describes a proton with x and y values that fall within the bounds of the target pixel's bins, while a miss has an x or y value beyond the boundaries of the bins. A missed proton is still part of the proton beam that hits the detector, it only does not hit the target pixel.

To investigate the frequency of misses, a 'miss counter' was added to the code. Every time a proton would miss the target pixel this counter would increase by 1, up until a proton hits the target pixel and the counter resets to 0. The value of the counter just before it resets is the number of protons that missed the pixel before there was an hit on the pixel. A low value would mean that a proton hit the target pixel again soon after the previous hit, while a high value means that it took a while before there was a coincident hit. These measured misses between hits are then collected and can be displayed on a graph to show the distribution of how frequently protons hit the target pixel. An average value for misses between hits is extracted, and is one of the three values a typical trial returns. The other two values are measured simultaneously as the misses between hits are counted. One counts the hits under limit, while the other counts the total number of hits.

4.4 Hits under limit and Hits per cell

The hits under limit indicate the number of protons that fell within the dead-hit threshold of a pixel. A proton is considered to fall under the limit if the number of misses before the hit has not exceeded the limit. The number of misses are counted starting from 0 after each hit, and if this counter exceeds the limit then the pixel is discharged and the following proton is measured by the detector. A proton landing before the limit is reached counts as a dead-hit, as the pixel hasn't discharged and won't detect the proton. Besides the number of hits under limit, the simulation is also counting the total number of hits on that pixel. The fraction of hits under limit over the hits per pixel is thus the percentage of hits under limit, which can be compared to the expected percentage of hits under limit.

4.5 Visualization & verification

To visualize the simulation and verify the functionality of the code, a timeline was plotted. The timeline shows us how often hits occur on the same pixel, and whether the code identifies these hits correctly. An example of the timeline for 10^6 protons is shown in Figure 8.

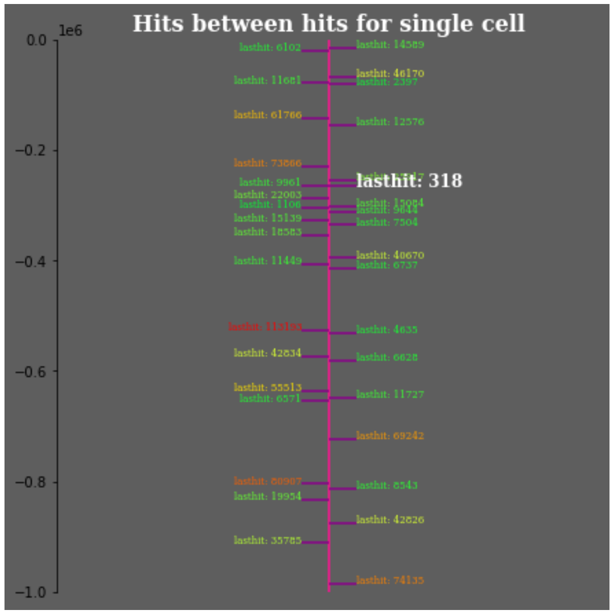

Figure 8: Timeline showing when protons hit the target pixel. Starting at the top from proton 0, increasing downwards to the last proton number 10^6. The value for last hit indicates the number of misses since the last hit, and determines the color of the labels from green (low) to red (high).

In Figure 8, each hit that misses the pixel is hidden as to reduce clutter. For each entry, the value for 'lasthit' indicates how many misses there have been since the last hit on target. This value for the number of misses since the last hit is equivalent to the time between on-hits, and represents the frequency of hits. The values for all the hits cause clutter, and so only a run of 10^6 protons was displayed. The labels are color coded to identify the different types of hits. A white value is a hit landing within the dead-time for a 10^8protons/cm^{2}s beam of 600 hits. The tags are then colored by a gradient, where green hits have fewer misses while red hits have many misses. The red and yellow tags were only used for visualization purposes. It can be seen that events that happen close after a previous one are correctly green, while the events after large gaps in the timeline are correctly colored orange or red. As the expectations for the differently colored hits match the observations, it can be concluded that the function works as intended.

4.6 Varying proton flux

The measurement for these values was then repeated for proton fluxes of 10^4 – 10^8 protons per second per cm^2. For the lower proton fluxes, the values for misses between hits were very skewed if only one second was simulated. The counter for number of misses would not be able to reach values past the proton flux, which undervalued the measured misses between hits. The simulation for these smaller fluxes thus used samples of 10^7 or $10^{7.5}$ protons and were taken to be run for many seconds. The limit for the deadtime would still match the limit of the smaller flux value, and the values for hits under limit and average hits per cell were divided by the number of seconds the simulation supposedly ran to attain the true value.

5 Results

In this section, the measurements from the simulations are shown and compared to the expected values. Results will be stated and briefly explained. In the Appendix, complete tables are shown with measured, expected, discrepancy and uncertainty values for every result. The discussion of the discrepancies and uncertainties will be done in the next section.

5.1 Misses between hits

Figure 9 shows the histogram of a single simulation of 10^7 protons. The red vertical represents the average value for misses between hits. The green vertical at x=60 represents the imposed limit under which protons are considered coincident. The black line is the probability function from equation 8 multiplied by 10^7 to get the expected number of misses between hits. It can be seen that the distribution of the simulated misses between hits measurements matches the expected gamma distribution.

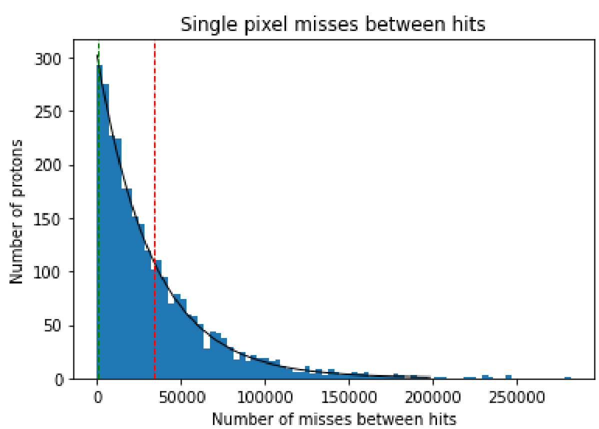

Figure 9: Histogram of misses between hits for sample of 10^7 protons. The black line indicates the expected value from equation 8. The average is indicated by the red line, and the dead-hit threshold is indicated by the green line.

The average number of misses between hits represents how long on average it takes for two protons to hit the same pixel. In the distribution above a proton flux of 10^7 protons/cm^{2}s was used. The average misses between hits for this single simulation is indicated by the red line at an x-value of 33593.47.

It was expected for the average misses between hits to be independent of proton flux, given that the proton fluxes are given time to measure. To confirm this, the average number of misses between hits was found during each simulation. This list was then averaged to give a final value for average number of misses between hits for the range of proton flux rates available from the AGOR cyclotron. The results were taken from 100 simulation runs for each of the fluxes up to 10^7, where $10^{7.5}$ and 10^8 ran for 70 simulations. The found average misses between hits for each proton flux were averaged and are shown in figure 10. It shows the expected misses between on-hits plotted against what was measured for varying proton fluxes.

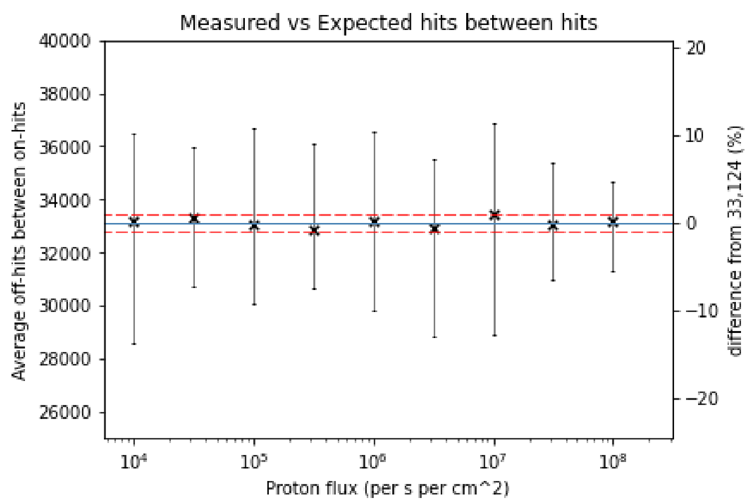

Figure 10: Average number of misses between hits for varying proton fluxes. The blue line indicates the expected value 33,124.

The expected value for hits between hits is equivalent to the pixel number, namely 33,124. This was reasoned to be independent of the proton flux, and is thus represented by a horizontal line. There seems to be no consistent pattern across varying proton fluxes. The discrepancies from the expected value range between 0% to 1%, indicated by the red lines in Figure 10.

5.2 Hits under limit

The hits under limit are the number of protons that land within the dead-time limit of the target pixel. The limit for misses between hits depends on the flux rate. The dead-time of a pixel was 6µs, and the limit in terms of misses/protons was the flux multiplied by 6µs. A flux of 10^6 protons for example has a threshold of 6 misses, and a flux of $10^{5.5}$ has a limit of 2 misses. Fluxes lower than this had a limit lower than 1, and were thus unmeasurable by the simulation as explained in section 3.

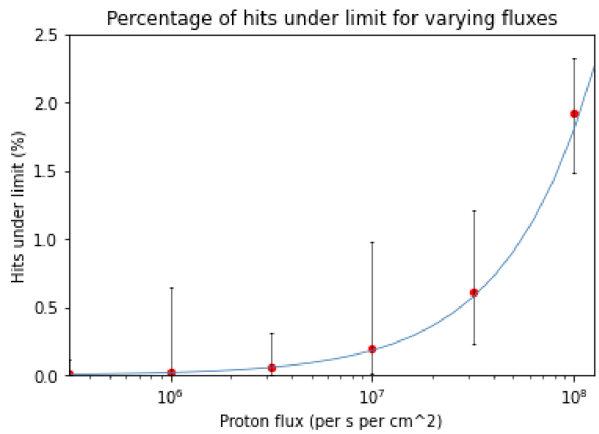

Figure 11: Percentage of hits under limit (dead-hit percentage) plotted for various fluxes between $10^{5.5}$ and 10^8

The expected percentage of protons falling within the deadtime is plotted against the values measured from the simulation in Figure 11. The points match each other closely, especially for fluxes above 10^7 hits per second with small discrepancies ranging from 0-3%. A brief overview of the precise dead-hit percentages for the general flux values are given below in Table 2. The uncertainties are taken from the highest and lowest measured value for deadhit percentage from the simulations.

Proton flux ($s^{-1}cm^{-2}$)	Hits under limit (%)	Discrepancy(%)
10^8	$1.82^{+0.40}_{-0.43}$	1.65
10^7	$0.186^{+0.78}_{-0.186}$	2.69
10^6	$0.017^{+0.62}_{-0.017}$	-5.88
10^5	0	n/a

Table 2: Specific dead-hit percentages and discrepancies from expectation for general proton fluxes.

The uncertainties shown are based on the largest and smallest observed dead-hit percentage for the respective proton flux.

5.3 Average hits per cell

The measured vs expected average hits per cell are plotted in Figure 12. The uniform distribution of the incoming proton sample gave a straightforward calculation for the expected average hits per cell.

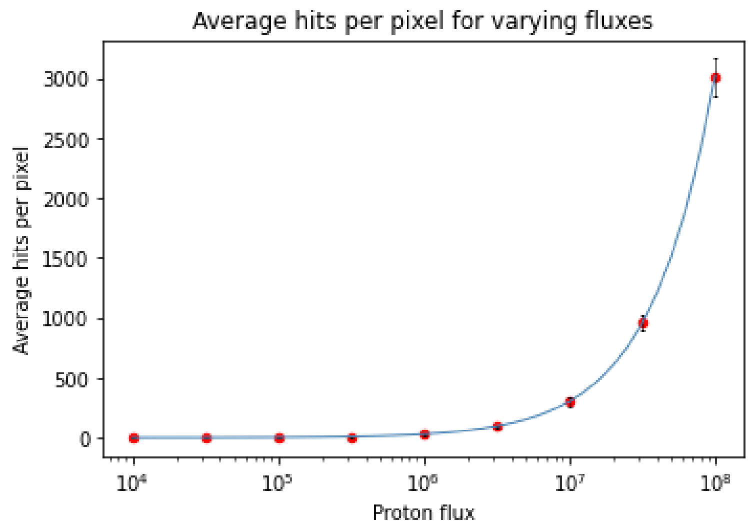

Figure 12: Average number of hits on target pixel for various fluxes ranging from 10^4 to 10^8. The red points are the measured results, while the blue line is the expectation.

Since the varying proton fluxes differ by a factors of 10 the x-axis was made logarithmic. However, on a standard x-axis the graph would show a straight line following the y=mx+c formula. The gradient tells us how many hits it takes to increase the average hits per cell, which relates directly to the pixel number of 33,124. Since the graph plots average hits per cell on the y-axis against the number of hits, the gradient is expected to be $\frac{1}{33{,}124}=3.018\times10^{-5}$. The measured gradient for the graph shown in Figure 12 is 3.015×10^{-5}, which is a difference of only 0.1%.

6 Discussion

In this section, the discrepancies and uncertainties found for the results are discussed. The discrepancies come from the differences between the measured average and the theoretical expectations. The uncertainties were chosen to be based off of the highest and lowest measured value from the measurements.

The uncertainties thus indicate the outliers, and encompass the entire range of measured outcomes. An admittedly better way to approach this would be to plot the distribution of outcomes and base the uncertainties on standard deviations. Unfortunately I could not redo the initial uncertainties as I ran out of time. The initial uncertainties were already considered very late into the project as initially I was only looking for approximate answers to whether the Timepix could reliably handle the largest available proton fluxes. The realization of improving the uncertainties came after, which didn't leave time to recalculate the uncertainties, re-graph the plots and rewrite the discussions.

6.1 Average misses between hits for varying fluxes

The average number of misses between every hit was expected to be equal to the pixel number, as 33124 protons would on average each hit their own pixel. However, this expected number of misses between hits can not be realized over a single second for lower fluxes. The average for misses between hits may not reach the expected probability outcome if it runs out of protons during that one second. For example, in the lowest case of $10,000$ protons per second per cm^2 you can never expect the average number of misses between hits to be 33,124. The lower the proton flux, the more the counter would be restricted and the less accurate the resultant average would be. Therefore, the simulation was run for multiple seconds to allow the counter to accurately count the misses.

Ideally, each simulation for every flux would run forever to maximize its accuracy. With my code however, trying to run 10^9 protons would cause the code to run out of RAM. Larger proton fluxes also obviously take longer to simulate. A balance thus has to be found between enough protons that the counter stays accurate and as little protons such that the simulation does not take too long. A simulation of 10^8 protons takes approximately 10 minutes in real time, while 10^7 protons takes 1 minute. In the interest of time and in order to collect more data points, 10^7 protons was taken as the optimal amount. Thus every simulation for proton fluxes below 10^7 protons/cm^2s simulated a total of 10^7 protons (or $10^{7.5}$ protons for the intermediate values), where for example a flux of 10^4 protons/cm^2s is considered to run for 10^3 seconds.

Discrepancies - The eventual average misses between hits measured for the proton fluxes were found to be very close to what was expected, with each point falling within a 1% discrepancy as seen in Figure 10. There is no evident correlation between the points as flux varies, which means that the discrepancies are likely due to statistical differences. This is to be expected, due to the simulation simulating a similar total of protons for each proton flux. The main difference across proton flux is the number of protons that describe the deadtime, which has no effect on the counted misses between hits.

Uncertainty - They can be seen to jump from larger to relatively smaller as the total number of simulated protons jumps between 10^7 and $10^{7.5}$. The uncertainty for 10^8 protons/cm^2s is the smallest. Aside from that, the extent of the uncertainty is purely due to statistical differences. These uncertainties ranged from 4.6-19.3%. This may because the average misses between hits fluctuated heavily depending on how many tail-end points are measured compared to face-end data points. Reducing this maximum uncertainty would require running every proton flux more than 10^7 protons to improve accuracy.

6.2 Hits under limit

The percentage of hits under limit is equal to the dead-hit percentage, which was the objective of this report. This represents the protons that aren't detected by the Timepix4 due to the pixel dead-time. The hits under limit must be interpreted as a percentage as opposed to the measured count in order to compare it with the expected values. The expected values were calculated from Equation 9, which was the integral of the probability density function and is shown by the blue line in Figure 11.

Discrepancies - The discrepancies in percentage of hits under limit ranged from 0.5% to 8.74%. This discrepancy comes from the errors carried forward from both the count of hits under limit and the count of hits per pixel. As explained in the following sub-section, the hits per pixel carried little error. This means the error mainly comes from the count of hits per pixel. Fluxes of 10^7 protons/cm^{2}s and below only had a couple hits under limit in each run, which meant that the largest possible flux was significant compared to the smallest possible flux.

This explains the specific discrepancies, as the lowest 0.5% came from 10^8 protons/cm^{2}s which had the most values for hits under limit. The largest discrepancy of 8.74% came from the lowest flux. However, this was already about as close as the measured number of hits under limit of 0.006 could get to the expected average number of deadhits of 0.0054. Each simulation ran of 10^6 protons/cm^{2}s ran for 10 seconds, so the measured number of deadhits would be divided by 10. During the 100 simulations, only 6 times did a hit fall within the deadtime, leading to the measured average of 0.006 deadhits. More measurements can be taken to improve the precision.

Uncertainties - The uncertainties for hits under limits was the largest from all, which was also largely due to the number of hits under limit. The minimum measured values in these lower flux cases had 0 hits under the limit, leading to a bottom uncertainty reaching all the way down to 0%. The maximum uncertainties were similarly smaller for higher fluxes and larger in the lower flux ranges. The smallest upper uncertainty was 22% for 10^8 protons/cm^{2}s, due to a measurement of 68 hits under limit compared to the expected 55. The largest upper uncertainty was 3670%, due to a measurement of 2 hits under limit compared to the expected 0.06 hits under limit. Regardless, the measured 0.019% fell close to the expected 0.018%. In the context of the problem, this means that even though certain seconds will contain more deadhits, over an extended period of time one can expect the average deadhit percentage to be reliable.

6.3 Hits per pixel

Discrepancies - The values for average hits per pixel were very accurate, matching the expected values with discrepancies ranging from 0.1-1.1%. The line of best fit for the measured values returned a gradient of 3.0189×10^{-5}, precisely matching what was expected. This accuracy was expected, as the field shape of the proton beam was confirmed to be uniform for large proton fluxes.

Uncertainties - The uncertainties based on largest and lowest measured values were relatively small, ranging between 5-15%. The few discrepancies found can be attributed to the small discrepancies in uniformity displayed in Section 4, Figure 7. The most hit pixel would obviously be busier than the least hit pixel.

7 Conclusion

This investigation was centered around the specific use of the Timepix4 with the AGOR cyclotron. The results will therefore be used when the Timepix4 is brought to the PARTREC facility and tested with protons produced by the AGOR cyclotron. The dead-times of these pixels thus cause protons to pass by undetected, the percentage of which was found in terms of hits under limit. This percentage was found for the varying proton fluxes that the cyclotron can produce. The dead-hit percentage of fluxes under 10^6 were near negligible, while those of higher fluxes ranged between 0.1% to a maximum of 1.8% for a proton flux of 10^8 protons/cm^{2}s.

The proton flux that should be used depends on what the investigated beam is used for. For general measurements where the specific quantity does not matter, 10^8 protons may be used as the percentage is small and it collects data faster. For scientific measurements such as investigating beams used for treating patients, smaller fluxes under 10^6 are advised in order to thoroughly detect the entire sample.

Another question asked in this report was whether multiple detection planes will be needed to make the data more complete. However, the percentage of hits undetected was low, even for the highest possible flux from the cyclotron. It is therefore unnecessary to investigate the use of multiple detection planes, unless samples of 10^8 protons/cm^{2}s need to be measured with under 1% accuracy or proton fluxes larger than 10^8 are investigated.

The performance of the simulation was persistently measured and verified in order to assure the validity of the results. This was done through timeline visualizations, misses between hits distribution analysis, and average hits per pixel accuracy. The results are thereby supported and accurate, under the assumptions made that are discussed in section 8.

8 Further Research

8.1 Challenging assumptions

While designing the experiment and simulation, it was necessary to make many assumptions. These assumptions approximate what happens in reality, but can always be improved upon to make them more accurate.

Uniform field shape - The proton beam that eventually hit the detector was assumed to be uniform. This is because a very small central radius from the 2D Gaussian passes into the collimator, and the collimator further works to align the proton beam. However, the exact field shape of the proton beam coming out of the collimator may not be exactly uniform, in which case the simulation has to be updated.

Continuous proton beam - Typically, cyclotrons produce particles in bursts. The time difference between consecutive protons not necessarily constant. In the simulation however, the protons are assumed to arrive one-by-one in fixed temporal intervals. The AGOR cyclotron emits protons in pockets which arrive at 60MHz. For lower fluxes, the expected protons per pocket would equal $10^7/6*10^7 = 0.167$ which is less than 1, and thus the assumption approximately holds. On the other hand, $10^8/6*10^7 = 1.67$, meaning there may be 1 or 2 protons in a pocket. Two protons can thus arrive simultaneously, which the code does not consider under its assumption. This assumption also means that the protons travel at equal velocities, which does not hold for proton beams with varying kinetic energies.

Dead-time threshold - The period of time during which a pixel is inert after recently detecting a proton was taken to be 6µs. However, as seen in Figure 3 from Section 2, the threshold varies depending on the charge injected into the pre-amp. The maximum it reached was 6µs, which was used in the simulation in order to encapsulate the largest possible dead-hit percentage. However, if the injected charge is know to be lower than the corresponding 20ke, then the measured dead-hit percentage will be naturally also be lower.

8.2 Multiple detection planes

Though the dead-hit percentage is low, it may still be significant in certain cases as mentioned in the conclusion. The optimal proton flux rate to treat a patient may be larger than 10^8 protons/cm^{2}s, such that the dead-hit percentage has to be reduced in order to properly calibrate the beam used for treatment. Or research could be done on larger proton flux rates from a different cyclotron. In both cases, the dead-hit percentage has to be further reduced to make a more complete measurement.

This may be done by using multiple detection planes. By letting the proton beam travel through two detection planes, the detected protons on both planes can be compared to compile a more complete detected proton list. With more planes, separate protons will be easier to identify and the dead-hit percentage would further decrease. Simulating this would require restructuring the proton beam, such that the protons slightly diverge from each other as opposed to them all running in parallel.

8.3 Birthday paradox

Early in the research, I found a connection between the misses between hits problem and the birthday paradox. They mirror a similar problem, where there are a number of people/pixels and the value of interest is how long it takes for there to be a shared birthday/hit. The paradox of the birthday problem entails that the number of people with a 50/50 chance of sharing a birthday is much smaller than expected, being ∼23 people.

Applying this logic to the simulation, it can be expected that the chance of a coincident hit on any of the pixels from the detector is much larger than that of a hit on a specific pixel. This is because the 'misses' of one pixel count as 'hits' to another pixel. Furthermore, each pixel that has been hit will trigger their own dead-time that can slip by protons. Thus it has a significant initial increase as the number of pixels that have been hit increases

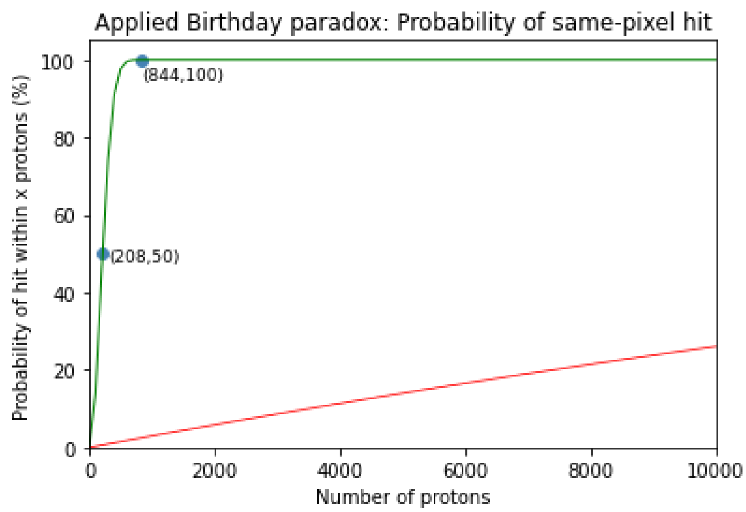

Figure 13: Probability of a hit on the same pixel considering all pixels (green) and a single pixel (red).

Exploring this result however was unfeasible, as the simulation platform couldn't handle it. The standard simulation of running 10^8 protons against a single target pixel takes about 9 minutes. According to Figure 13, the expected number of hits it takes to get a coincident hit on any pixel is 208 hits for a 50% of coincident hit and 845 hits for a $99.99 \approx 100\%$ chance of coincident hit. This means it would take 20 hours for a 50% chance of a single data point, or up to 85 hours for a 100% chance of a data point. To gather a sufficient number of data points, it would take at least two weeks of computing which is outside of the timescale for this project. The simulation for the chance of coincident hits on a single target pixel takes only those 9 minutes, so a meaningful number of data points for multiple variables is attainable within a shorter time span. However, this may be investigated if one either significantly optimizes the code for the simulation, has access to more computing power, or has a long time frame for taking measurements.

References

[1] Kazu Akiba et al. “First charged tracks reconstructed with Timepix4 ASIC”. In: *arXiv preprint arXiv:2210.01442* (2022).

[2] Marıa Plana, KAM De Bruyn, and MC van Veghel. “Feasibility Study: Time-of-Flight Measurements of a Proton Beam with the Timepix4 ASIC”. PhD thesis. 2022.

[3] Yang-Gun Suh et al. “Proton Beam Therapy versus Photon Radiotherapy for Stage I Non-Small Cell Lung Cancer”. In: *Cancers* 14.15 (2022), p. 3627.

[4] Hui Liu and Joe Y Chang. “Proton therapy in clinical practice”. In: *Chinese journal of cancer* 30.5 (2011), p. 315.

[5] UMCG Research. “Particle Therapy Research Center (PARTREC)”. In: *UMCG Reserach* (). URL: https://umcgresearch.org/w/partrec.

[6] X Llopart et al. “Timepix4, a large area pixel detector readout chip which can be tiled on 4 sides providing sub-200 ps timestamp binning”. In: *Journal of Instrumentation* 17.01 (2022), p. C01044.

[7] K Heijhoff et al. “Timing performance of the Timepix4 front-end”. In: *arXiv preprint* (2022).

[8] National Institute of Standards and Technology. “Gamma Distribution NIST”. In: *Engineering Statistics Handbook* (). URL: https://www.itl.nist.gov/div898/handbook/eda/section3/eda366b.htm.

9 Appendices

9.1 Appendix A: Tables of results

Table 3: Misses between hits

Proton flux (s^{-1} cm^{-2})	Measured	Expected	Discrepancy (%)
10^8	$33030^{+5.7\%}_{-4.6\%}$	33124	0.28
$10^{7.5}$	$33089^{+6.3\%}_{-7.1\%}$	33124	0.1
10^7	$33093^{+13.6\%}_{-10.4\%}$	33124	0.09
$10^{6.5}$	$33087^{+12.3\%}_{-7.9\%}$	33124	0.11
10^6	$33105^{+19.3\%}_{-13.1\%}$	33124	0.06
$10^{5.5}$	$32901^{+6.8\%}_{-9.9\%}$	33124	0.68
10^5	$33016^{+18.0\%}_{-11.1\%}$	33124	0.33
$10^{4.5}$	$33016^{+7.9\%}_{-8.1\%}$	33124	0.33
10^4	$32959^{+13.9\%}_{-10.0\%}$	33124	0.5

Table 4: Hits under limit

Proton flux (s^{-1} cm^{-2})	Hits per cell	Hit count under limit	Measured hits under limit (%)	Expected	Discrepancy (%)
10^8	3023	55.1	$1.82^{+22\%}_{-24\%}$	1.79	1.65
$10^{7.5}$	955.2	5.62	$0.59^{+102\%}_{-64\%}$	0.57	3.39
10^7	302.3	0.563	$0.186^{+421\%}_{-100\%}$	0.181	2.69
$10^{6.5}$	95.6	0.055	$0.058^{+386\%}_{-100\%}$	0.057	1.57
10^6	30.23	0.0060	$0.019^{+3670\%}_{-100\%}$	0.018	5.5
$10^{5.5}$	9.610	0	n/a	0.005	n/a
10^5	3.030	0	n/a	0.002	n/a
$10^{4.5}$	0.958	0	n/a	0.001	n/a
10^4	0.303	0	n/a	0.000	n/a

9.2 Appendix B: Link to Python code of simulation

The python code for the simulation was uploaded to a google colab document. This can be accessed via this link .