Simulation of the Passage of Milli-Charged Particles in a Scintillator for Application to a Proposed Auxiliary Detector for ATLAS (ADAM).

Natalia Litvak
University of Alberta
Supervisor: James Pinfold

I. Introduction

Millicharged particles (mCPs) are a proposed type of particle that have charge much smaller than that of an electron. The Standard Model features fractionally charged particles, quarks and antiquarks, which have charge ±1/3 or ±2/3 that of the electron, however, no Standard Model particle has charge less than 1/3 that of the electron. The search for mCPs could offer insight on questions in physics such as electric charge quantization and dark matter. MCPs are predicted in several beyond the Standard Model physics theories. For instance, extensions of the Standard Model with extra U(1) gauge symmetries feature these particles (1). This is common in string theories (1), as well as in certain dark sector models, which proposes a new sector of particles as an explanation for dark matter (2).

One experiment searching for mCPS is the MoEDAL-MAPP detector at CERN, which searches for evidence of exotic particles produced at the LHC, and the MAPP upgrade to the MoEDAL detector adds a scintillator setup designed specifically to search for mCPs. In addition, experiments like the milliQan detector, a CMS sub detector that analyzes data from proton-proton collisions at the LHC (3), and the planned ArgoNeuT detector at the Main Injector beam at Fermilab (4) search for evidence of mCPs. From these and other experiments, as well as through theoretical means, constraints have been placed on the mass and charge of mCPs, however, there is still a large range of mass and charge that these particles might have (5).

I worked on simulating the production of mCPs in proton-proton collisions at the LHC and their subsequent passage through a detector. My project focused on learning how to simulate the mCPs and eventually apply that knowledge to the design of a new sub-detector for ATLAS called ADAM (Auxiliary Detector for Atlas Muons), which will use a scintillator structure to search for mCPs. Through this simulation, we can find out the expected light output and energy deposition of mCPs of a range of masses and charges according to our model.

II. Tools Used

Geant4

GEANT4 is a simulation toolkit designed to simulate the passage of particles through matter, and the core tool in our project. GEANT4 simulates particle interactions in matter using the Monte Carlo method. To create a simulation in GEANT4, the user needs to define the set-up of the experiment, including the geometry, material, volume, electromagnetic fields, and all other necessary information. However, the initial particles need to be fully defined with their kinetic information. For us, this means we need to define the mCPs with their initial kinetic information already known to simulate their passage and energy loss.

Madgraph

MadGraph is a Monte Carlo event generator, which is widely used to simulate events at the LHC. MadGraph allows the user to import a model, and generate events accordingly. We use

MadGraph to simulate the creation of mCPs from proton-proton collisions, simulating the conditions at the LHC, and obtain the kinetic information of produced mCPs according to our model. This allows us to obtain mCPs with their initial kinetic information to use in our GEANT4 simulation.

ROOT

ROOT is a framework developed at CERN, and used for data processing. It is able to efficiently store, access, and visualize large amounts of data. ROOT is written in C++, but PyROOT allows full interaction with the system through Python. We use ROOT to store the simulation data from the GEANT4 simulations, and I used PyROOT in my project to access this data, using it to create graphs for visualization of the data and access and store the values of radiation produced in our simulations.

III. <u>Project Details</u>

Collision Event Generation

The end goal of my project is to build a full simulation of mCPs passing through the ADAM detector. The results can be used when analyzing data to search for mCPs in this detector, and will help determine the optimal design for ADAM for this search. At first, we are working on simulating the passage of mCPs through a single plastic scintillator, as this allows us to test our simulation in a less computationally heavy environment before implementing the full geometry for ADAM.

We used MadGraph to generate the proton-proton collision events that produce mCPs. We used a model for the production of mCPs, in which mCPs and their corresponding antiparticles are represented as ψ and ψ˜, respectively, and defined in accordance with the Monte Carlo particle numbering scheme using the PDG number 9000005 (typically a 7-digit identifier) (6). Then, I ran the MadGraph program for a variety of masses (0.1, 0.10566, 0.5, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 GeV), 100 000 events for each mass, and a collision energy of 14 TeV, the energy that the LHC collisions will have in the high-luminosity upgrade (7). As its output, MadGraph generates Les Houches Event (LHE) files. The LHE files contain the specific information of the mCPs produced in the collisions. For each mass I specified, one LHE file was produced, containing information of one opposite charged mPCs pair for each generated event.

Parsing

To extract the information from the LHE files, we use a parser code written in python. I made a code that looped through all the LHE files produced, and made histograms of the beta value and kinetic energy distributions of the mCPs produced in each run. In addition, I produced a graph of the beta value distribution across all runs.

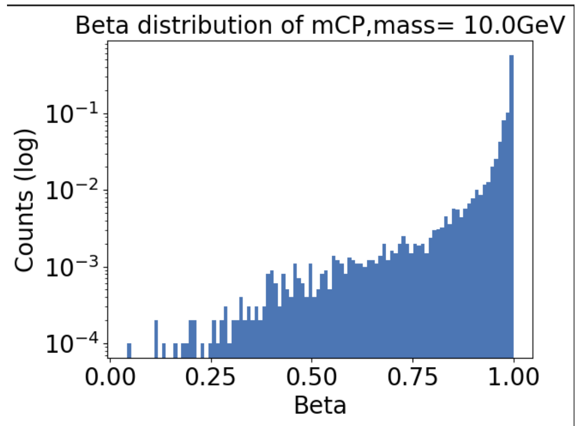

Figure 1. Beta distribution for mCPs with mass 10 GeV

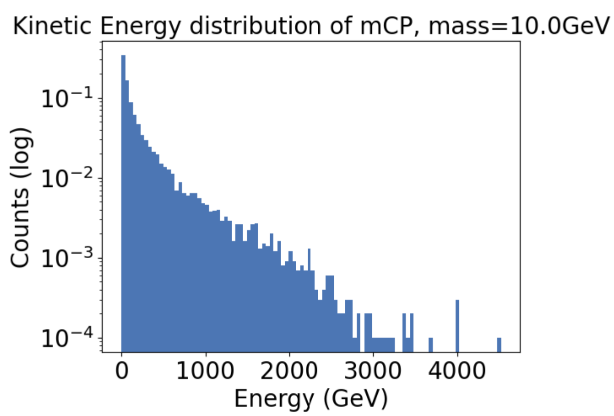

Figure 2. Kinetic energy distribution for mCPs with mass 10 GeV

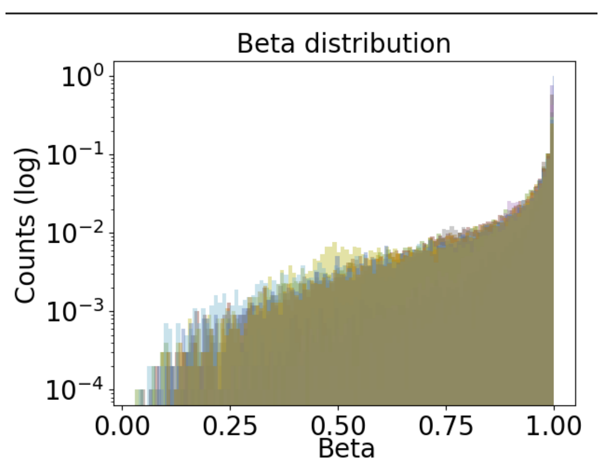

Figure 3. Beta distribution for all generated mPCs

The main purpose of the parser was to create ASCII text files containing the information from the LHE files. The parser locates the mPCs' PDG numbers within the <event> tag, and from this extracts the components of momentum (x, y, and z), energy, and the particle’s mass. This information is then stored in the output file. The ASCII files that are obtained through this process are subsequently employed in the GEANT4 simulation, used to define the particles for the simulation.

GEANT4 Simulation and Verification

We use GEANT4 for the main simulation of the project. The latest version of GEANT4 is GEANT4.11, and one of my tasks was to update the code from GEANT4.7, a previous version in which the code was written in, to the most recent version, GEANT4.11.1.1. To do this, I used the Geant4 Cross Reference online, which has the information of all previous GEANT4 versions and the changes implemented in the updates (8).. I learned a lot about GEANT4 and working with it while doing these updates, as I had to investigate all parts of the code and learn the uses of the outdated lines and find their newly implemented versions.

In GEANT4, we defined the geometry of the detector under consideration. This encompassed the detailed specifications and material properties of the scintillator, as well as defining the volume within which the particle exists in the simulated system. In our model, our particles pass through a wall of concrete before reaching the scintillator.

In addition, we defined the physics lists that dictate how the mCPs interact with the materials they pass through. Charged particles going through the scintillator are detected due to the energy loss that happens as they interact with the material (9). The three main ways of energy loss for mCPs are ionization, bremsstrahlung and pair production (10). Ionization happens when the charged particle collides with an atom. During this interaction, energy is transferred from the charged particle to an electron in the atom, which escapes from its bound energy level, and the atom becomes ionized (11). Bremsstrahlung occurs when a charged particle is deflected by a nucleus, and loses energy as it decelerates, releasing photons (11), and pair production occurs when a virtual photon is produced by mCPs interacting with an atomic nucleus, which decays into an electron-positron pair (10). We defined the physics of these three interactions for mCPs as a part of our model.

In order to check that our models are realistic, we also simulate the passage of muons, whose interactions and properties we understand well. Muon interactions are found in the standard GEANT4 physics list. If our models are realistic, an mCP with the same mass and charge as a muon simulated with our model should behave identically to a muon simulated with the standard GEANT4 physics. Muons also interact in the scintillator through bremsstrahlung radiation, ionization, and pair production. One difference we have to account for is that mCPs do not decay, due to charge conservation, so when doing this comparison, we have to disable muon decay from the list of allowed interactions.

I ran a GEANT4 simulation of muons created in proton-proton collisions, and passing through the geometry defined for our simulation, that is, a concrete wall and then a plastic scintillator. I ran the GEANT4 simulation for muons of 9 different initial kinetic energies, (100, 1000, 5000, 10000, 50000, 100000, 1000000, 5000000, and 10000000 MeV). I modified the code to allow one energy loss mechanism at a time, so that I had three runs per energy level. For the seven lowest energy levels, the simulation was ran with 10 000 events, but for the last two, it was ran with 1 000 events, because at higher energy much more photons were produced and the simulation took much longer, so this was decided on to allow us to have the results in a reasonable time. In addition, we found that the ionization photon production was much higher than the other methods, so we modified the code to make the frequency of events 1/10 the original, again to make the simulation easier and faster to run, and multiplied our results for the ionization photons by 10 to receive the expected production.

The GEANT4 simulation produced a root file every run, containing data about the run including the total energy of the muons, and the mean amount of scintillation photons detected. I created a code using PyROOT to loop through the root files, read them, and collect the total energy and the mean amount of scintillation photons produced through each energy loss mechanism. Then, I created a plot representing the photons produced at each energy.

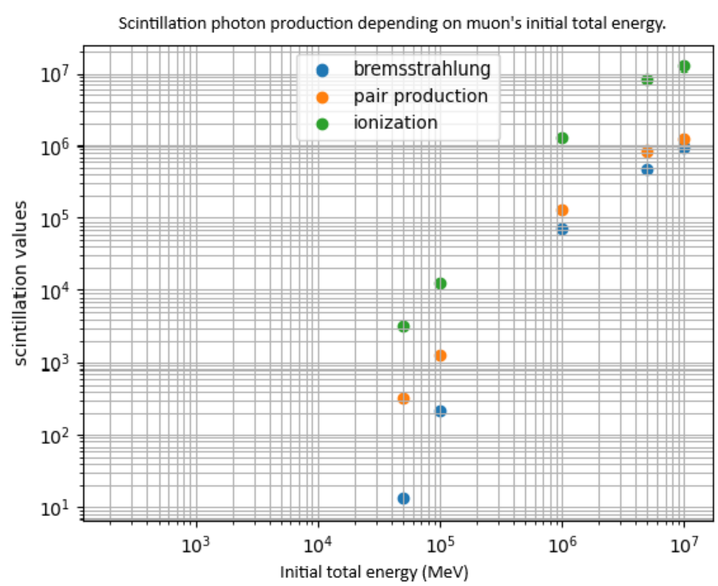

Figure 3. The amount of scintillation photons produced at varying initial energies for bremsstrahlung radiation, pair production, and ionization.

We noticed that photons were only produced at energy levels equal to or above 50000 MeV. This meant that before this energy, the muons were unable to penetrate through the concrete wall and reach the detector. Because I used a logarithmic y-axis to best represent the data, the values below these energies are not visible, as they would be uniformly zero. In addition, I calculated the uncertainty in our values using the standard error formula, $dx = \frac{\sigma_x}{\sqrt{N}}$, where ±dx is the uncertainty, σ_x is the standard deviation of the mean (which we obtain from the root file), and N is the number of events done in that simulation.

IV. **Technical knowledge and challenges**

During my summer work I learned a lot about the principles and architecture of the computer simulation of physical experiments, and the technical tools involved. To install and run all the needed software on my computer, I installed and got familiar with Windows Service for Linux and Linux itself. Windows Service for Linux is a feature of Windows OS, which helped me as GEANT4 and ROOT are fully supported for Linux, but not Windows, which I had been using. I was already familiar with Python programming, but I also learned the ROOT Python interface (PyROOT) and familiarized myself with the ROOT framework, and with python programming in ROOT. Additionally, during my work I obtained basic knowledge of C++, as part of the work I did with upgrading the GEANT4 code, as well as basic ROOT commands, involved C++ code. The core of the simulation I worked on was done using the GEANT4 toolkit, so to work with it I had to learn many parts of it, including its class structure, the difference in GEANT4 versions, and the simulation code structure. Another software I was learning and using was the MadGraph5 event generator. I installed MadGraph5 and ran several practices with it to get familiar with how it works. I was provided with a model to run in MadGraph5, so I familiarized myself with the model so that I understood it, and I modified parts of it with the runs to allow different masses of the mCPs to be produced.

V. **Conclusion**

MCPs are a proposed type of particle which, if found, could provide answers in areas such as BSM physics and dark matter. In our project, we worked on simulating the production of mCPs in proton-proton collisions at the LHC, and their subsequent passage through a concrete wall and plastic scintillator. In this project, we simulate the proton-proton collisions using MadGraph5, obtaining the mCPs initial kinetic conditions, then use a python parser to convert MadGraph's output LHE files to ASCII text files that we can then use in a GEANT4 simulation of the mCPs passage through the concrete wall and scintillator, obtaining the values of their photon production in the scintillator. In this project, I worked on upgrading the GEANT4 code of the simulation, running MadGraph, and writing part of the parser code. In addition, we use simulations of muons to check that our models are reasonable, and I helped run these simulations and analyze their results. Through these simulations, we can provide a framework for completing the design of ADAM that will be used in the search for mCPs at the LHC.

VI. Acknowledgments

I want to thank my supervisor, Professor James Pinfold, and also Abhinab Mukhopadhyay for help and guidance with my project. I also thank the MoEDAL and ATLAS teams at the University of Alberta, who I worked with and learned from as our projects shared similar software and principles, for being welcoming and helpful to me during my summer. I also thank CERN, the Natural Sciences and Engineering Research Council of Canada (NSERC), the Institute of Particle Physics (IPP) and Physics department of the University of Alberta for the possibility to participate in the CERN Summer Student Program.

References

1. Agnese R. *et al.* (CDMS Collaboration): First Direct Limits on Lightly Ionizing Particles with Electric Charge Less than e/6. *Phys. Rev. Lett. 114,* 111302 (2015)
2. Cline, J. M., Liu, Z., & Xue, W.: Millicharged atomic dark matter. *Phys. Rev. D, 85(10), 101302* (2012)
3. Ball, A., Brooke, J., Campagnari, C., Carrigan, M., Citron, M., Roeck, A., … & Zaraket, H.: Sensitivity to millicharged particles in future proton-proton collisions at the LHC with the milliQan detector. *Phys. Rev. D, 104(3). 032002* (2021)
4. Acciarri R., Adams C., Asaadi J. et al.: Improved Limits on Millicharged Particles Using the ArgoNeuT Experiment at Fermilab. Phys. Rev. Lett. **124**, 131801 (2020)
5. De Montigny, M., Ouimet, P. P. A., Pinfold, J., Shaa, A., & Staelens, M.: Minicharged Particles at Accelerators: Progress and Prospects. *arXiv preprint arXiv:2307.07855* (2023)
6. Tanabashi, Masaharu, et al.: Review of Particle Physics: particle data groups. *Physical Review D* 98.3 p.560 (2018)
7. High Luminosity LHC, CERN, https://home.cern/science/accelerators/high-luminosity-lhc
8. Geant4 Cross Reference, https://geant4.kek.jp/LXR/index.html
9. Pinfold, J. L.: The MoEDAL Experiment at the LHC–a New Light on the Terascale Frontier. *Journal of Physics: Conference Series*. Vol. 631. No. 1.(2015)
10. Arefyeva, N., et al.: Passage of millicharged particles in the electron beam-dump: Refining constraints from SLACmQ and estimating sensitivity of NA 64 e. *Physical Review D* 106.3: 035029. (2022)
11. OMA School on Monte Carlo Simulations, CERN, Ionization and Transport,, (2017)