Creating a Web-Based Toolkit for Modeling Reactions in the ISOLDE Solenoidal Spectrometer

Project available on GitHub Repository

Student: Pejanovic Rajka
Supervisor: Patrick MacGregor

Erasmus Mundus Joint Master Degree in Nuclear Physics
CERN - Summer School Program 2023

August 24, 2023

Abstract

The development of a web-based tool to assist in the selection of viable reactions for measurement with the ISS would be a significant advancement in nuclear physics research. By enabling researchers to explore and optimize their experimental designs in real-time, the tool would enhance the efficiency and impact of experiments conducted at the ISS. It would empower scientists to make informed decisions, leading to a deeper understanding of the nuclear structure and further advancing our knowledge of the fundamental properties of atomic nuclei.

This paper presents the design, implementation, and functionality of the developed toolkit. The toolkit integrates crucial components, including the calculation of optimal angles, mass excess values, and velocity, that play a pivotal role in nuclear physics experiments. The calculations are based on well-established physics principles and are executed through a user-friendly interface, making the toolkit accessible to both novice and expert researchers.

The toolkit's architecture encompasses several interconnected modules, such as data retrieval, calculation of nuclear masses, velocity determination, and optimal angle calculations. These modules are linked via JavaScript scripts and server-side code, enabling seamless communication between user inputs, calculations, and results visualization. Notably, the toolkit accommodates variations in detector characteristics, allowing users to account for the finite size of the detector when calculating optimal angles.

The user interface of the toolkit provides an intuitive and efficient platform for researchers to input relevant parameters and obtain desired results. Through interactive forms, users can input variables such as particle masses, energies, and magnetic field strengths. The toolkit then performs complex calculations behind the scenes, offering outputs such as optimal angles and velocities. The results are presented in a clear and concise manner, enhancing the accessibility of complex nuclear physics concepts.

Key Words: ISOLDE, ISS, Development, ToolKit, Nuclear Reaction, Magnetic Filed, HTML, CSS, JS, Python

Contents

1 Introduction

CERN, situated near Geneva, Switzerland, is a monumental center for particle physics research. It houses an array of particle accelerators and detectors designed to explore the deepest corners of the physical universe. Among its renowned facilities, ISOLDE stands as an exemplary laboratory for nuclear physics studies. ISOLDE's pioneering concept involves producing isotopes through nuclear reactions and subsequently isolating and studying them. This approach allows for the investigation of exotic and rare isotopes that offer unique insights into the nuclear landscape.

ISOLDE's capabilities have been instrumental in advancing nuclear physics research. By providing a wide spectrum of isotopes, ISOLDE empowers researchers to probe a diverse range of nuclear properties, from nuclear masses and shapes to decay modes and lifetimes. These investigations contribute not only to fundamental science but also to interdisciplinary fields like astrophysics and material science.

The ISOLDE Solenoidal Spectrometer (ISS) is undergoing development to facilitate meticulous studies involving inelastic scattering and transfer reactions, induced by radioactive ion beams from HIE-ISOLDE. Envisioned as part of an extensive scientific program, ISS aims to explore critical facets of nuclear structure and nuclear astrophysics. The spectrometer employs an advanced design founded on the successful HELIOS concept. This approach efficiently conveys light-charged particles emitted during nuclear reactions through a potent solenoidal magnetic field, directing them to an array of position-sensitive silicon detectors positioned along its axis. The silicon array's capability to gauge particle energies and interaction positions eliminates the kinematic compression issues encountered in conventional methods, enabling precise determination of the reaction Q-value. The ongoing development entails a configuration featuring 24 double-sided silicon strip detectors equipped with ASIC readout technology. This configuration is engineered to achieve Q-value resolutions of nearly 20 keV. The full ISS array's integration is scheduled for the second extended shutdown (LS2) and subsequent availability for physics investigations.

Simplifying, the ISOLDE Solenoidal Spectrometer (ISS) is a powerful device utilized in nuclear physics research to measure the nuclear-structure properties of nuclei formed in transfer reactions. Transfer reactions involve the exchange of individual nucleons between a beam and a target nucleus in a single step. By studying these reactions within a strong solenoidal magnetic field, the ISS enables the precise investigation of charged particles' helical trajectories. Detectors strategically positioned around the beam axis capture the energies and positions of these charged particles, facilitating the extraction of key properties of the excited states in the residual nucleus.

When selecting reactions to be measured using the ISS, it is imperative to simulate various aspects of the reaction to ensure the feasibility and success of the proposed experiment. These simulations encompass a range of parameters, including the optimal positioning of detectors, the strength of the magnetic field, the expected reaction rate, the angular distribution of the ejected particles, and factors that broaden the widths of states in an excitation-energy spectrum. By considering these factors and their impact on the experimental outcome, physicists can develop proposals that are more likely to yield significant insights into the nature of nuclear structure.

One crucial element in the design of experiments involving the ISS is the strategic placement of detectors. These detectors are carefully positioned around the beam axis within the solenoidal magnetic field to ensure efficient and accurate measurements. The positioning is determined based on factors such as the desired angular coverage, the specific particles of interest, and the optimization of the signal-to-noise ratio.

By simulating the trajectories of particles within the magnetic field and their interactions with the detectors, researchers can assess the effectiveness of different detector configurations and make informed decisions on their placements.

Thus, to facilitate the process of designing viable experiments with the ISS, the aim of this project is to develop a web-based tool that enables researchers

to tune the experimental parameters and receive real-time feedback on their choices. Such a tool would serve as an invaluable resource for future proposals at the ISS and provide significant benefits to the current users of the device. Researchers would be able to input their desired experimental parameters, such as detector positions, magnetic field strength, reaction rates, and angular distributions, into the web-based tool.

The tool would then utilize sophisticated simulations and algorithms to evaluate the impact of these choices on the experiment. It would provide real-time feedback, presenting researchers with information on the expected outcomes, including the number of detected events, energy resolution, angular coverage, and the impact of broadening effects. This feedback would allow researchers to make informed decisions and optimize their experimental designs to enhance the chances of success and improve the understanding of nuclear structure.

1.1 Background and Theory

The study of nuclear reactions traces its origins back to the years 1909 to 1911 when Rutherford's research team made a significant discovery [1], [2]. They observed the nuclei of gold (Au) and platinum (Pt) atoms through the analysis of the scattering of alpha particles at large angles by thin foils made of these metals. However, it was not until 1932 to 1933 that the first systematic investigations into nuclear reaction mechanisms took place, led by Fermi and his colleagues [3]. During their experiments, they exposed 38 different nuclides, ranging from hydrogen to uranium, to newly discovered neutrons and observed the resulting induced radioactivity. This research revealed two notable findings: (a) the cross sections for the (neutron, gamma) reactions were exceptionally large, and (b) the cross sections at very low energies varied significantly between different nuclides. The first observation suggested that the system remained in a highly excited state for an extended period, emitting photons over a timescale estimated to be at least 10^{-16} seconds, which was much longer than the typical collision time of around $10"-22$ seconds. The second observation was later attributed to resonances occurring at specific incident energies.

Nevertheless, nuclear reactions are used for a variety of goals, not just for the ones mentioned above. They can be used to study the structure of nuclei; sometimes, they can be the only way to probe nuclear structure, especially far from stability. Nuclear reactions also provide information about the interaction between nuclei, either to study the fundamentals of the nuclear force, or to measure reaction rates, which are major inputs in other fields of physics, like nuclear astrophysics, or in a broad range of nuclear applications, like nuclear power or the production of radioactive isotopes for medical purposes.

hOWEVER, When two particles collide, numerous processes may occur, resulting in various outcomes. A typical nuclear reaction is commonly expressed as:

$$A + a \rightarrow B + b + Q \quad \text{(Equation 2.1)}$$

In this notation, A stands for the target nucleus, a signifies the projectile particle, B represents the residual nucleus, and b corresponds to the observed particle. This reaction is symbolized as $A(a, b)B$. Specific isotopes are indicated using the mass number as a superscript to the left of the chemical symbol. Special symbols are assigned to fundamental particles and lightweight nuclei, e.g., e for electrons, π for pions, p for protons, n for neutrons, d for deuterons or ^{2}H, t for tritons or ^{3}H, and α for alpha particles or ^{4}He. Photons or gamma rays are denoted as γ [4].

Furthermore, both B and b may exist in excited states, indicated by subscripts or asterisks, like B^*.

The symbol Q in Equation 2.1 represents the energy released during the reaction. If particles b and B are in their ground states, this energy is denoted as Q_0. Conservation of total energy implies that $Q = 0$ signifies a conversion between kinetic energy and internal excitation energy or rest energy. Positive Q values indicate exoergic reactions, releasing energy, while negative Q values denote endoergic reactions requiring energy input. In the latter case, a minimum energy threshold is necessary for the reaction to occur. In the center-of-mass system (CMS), an incident energy E_i greater than $-Q$ is necessary for the energy in the final state (E_r) to be positive.

Though the primary focus often revolves around two particles b and B in the final state, it's important to acknowledge that collisions can yield three or more products. When ample energy is available, the colliding systems can disintegrate entirely into their constituents, although such events are rare [5]. In scenarios where multiple reaction products emerge, the collision is referred to as a spallation reaction.

1.2 Nuclear Reaction Classification

We classify reactions in many ways, If the incident and outgoing particles are the same (and correspondingly X and Y are the same nuclei), it is a scattering process.

Elastic if Y and b are in their ground state and inelastic if Y or b is in an excited state (from which it will generally decay quickly by γ emission). Sometimes a and b are the same particle, but the reaction causes yet another nucleon to be ejected separately (so that there are three particles in the final state); this is called a knockout reaction. In a transfer reaction, one or two nucleons are transferred between the projectile and target, such as an incoming deuteron turning into an outgoing proton or neutron. thereby adding one nucleon to the target X to form Y, Reactions can also be classified by the mechanism that governs the process. In direct reactions (of which transfer reactions are an important subgroup)[6]. Only very few nucleons take part in the reaction, with the remaining nucleons of the target serving as passive spectators Such reactions might insert or remove a single nucleon from a shell-model state and might therefore serve as ways to explore the shell structure of nuclei. Many excited states of Y can be reached in these reactions. The other extreme is the compound nucleus mechanism, in which the incoming and target nuclei merge briefly for a complete sharing of energy before the outgoing nucleon is ejected, somewhat like the evaporation of a molecule from a hot liquid. Between these two extremes are the resonance reactions, in which the incoming particle forms a " quasi-bound" state before the outgoing particle is ejected [7].

1.2.1 Nuclear Reactions and ISS

Isolde solenoidal spectrometer gives important information about nuclear structure and it is highly connected to the scattering reactions and its kinematics. These reactions hold immense significance in the realm of nuclear physics as they provide a unique avenue to investigate the intricate structures of atomic nuclei. Among these reactions, a particular subset holds great interest in solenoidal spectrometry, mentioned direct reactions. These reactions induce a specific vibrational mode within a nucleus, enabling a direct exploration of its structure. Notable instances are transfer reactions, involving the exchange of single or multiple nucleons between a beam and a target nucleus. Historically conducted under "normal kinematics" (NK), these investigations entail directing a stream of particles at a denser target, with resultant emissions measured using a fixed-angle configuration. This approach, which entails measuring the energy and scattering angle of the ejected particle, offers profound insights into the underlying nucleus's composition.

However, this methodology faces limitations when dealing with short-lived nuclei due to the impracticality of generating appropriate targets. In such scenarios, the transient nucleus must be synthesized and subsequently directed towards a target comprising lighter nuclei, a concept termed "inverse kinematics" (IK). Yet, the inversion of beam and target, particularly when significant mass differences exist, introduces complexities when attempting to measure reactions using a fixed-angle setup.

This synthesis provides an overview of the kinematic intricacies governing transfer reactions within the context of solenoidal spectrometry, exemplified by platforms such as HELIOS [8], ISS [9], and SOLARIS [10]. This study unfolds through classical and relativistic mechanics, highlighting the extraction of critical parameters like scattering angles and nuclear excitation energy through measurements on detectors encircling the beam axis.

Summarising, we will be focused on direct reactions and we will be working in the inverse kinematics.

2 Physics Background

One particularly important application of single-particle transfer reactions, especially (d,p) and (d,n), is the study of low-lying shell-model excited states. Several such states may be populated in a given reaction; We can choose a particular excited state from the energy of the outgoing nucleon. Once we have done so, we would like to determine just which shell-model state it is. For this, we need the angular distribution of the emitted particles, which often gives the spin and parity of the state that is populated in a particular reaction [6]. Angular distributions, therefore, are of critical importance in studies of transfer reactions. (Pickup reactions, for example (p,d), in which the projectile takes a nucleon from the target, also give information on single-particle states.)

2.1 Classical Mechanics Approach

Here, the masses $m_{1,2,3,4}$ correspond to the beam, target, ejectile, and recoil nucleus respectively. In the LAB (CM) frame, the velocities of particles are labelled with u (v). The center-of-momentum velocity for an inertial frame (relative to the CM frame) is labelled with V. Kinetic energies of particles are labelled T ($\overline{T}$) in the LAB (CM) frame. The angle θ_{lab} is the angle of the ejectile particle in the LAB frame *relative to the direction of the beam*. The angle θ_{cm} is measured *relative to the initial direction of its parent (target in IK, beam in NK)*. Additionally, the speed of

light will be set to 1 so that factors of c are not obscuring the underlying mathematics.

First, the kinetic energies of the two initial particles (i.e. the beam energy and the target energy) are

$$T_1 = \frac{1}{2}m_1u_1^2, \quad T_2 = \frac{1}{2}m_2u_2^2 = 0. \tag{1}$$

and the centre-of-momentum velocity before the reaction, V_i is

$$V_i = \frac{m_1u_1 + m_2u_2}{m_1 + m_2} = \frac{m_1u_1}{m_1 + m_2} = \frac{\sqrt{2T_1m_1}}{m_1 + m_2}. \tag{2}$$

The total energy in the LAB frame initially, T_i, is

$$T_i = T_1 + T_2 = T_1. \tag{3}$$

A similar exercise can be performed for the CM frame. The necessary transformation from LAB to CM is

$$v_1 = u_1 - V_i = \frac{m_2u_1}{m_1 + m_2}, \tag{4}$$

$$v_2 = u_2 - V_i = -V_i, \tag{5}$$

such that

$$T1 = \frac{1}{2}m_1v_1^2, \quad T2 = \frac{1}{2}m_2v_2^2, \tag{6}$$

$$Ti = T1 + T2 = \left(\frac{m_2}{m_1 + m_2}\right)T_1. \tag{7}$$

The centre-of-momentum velocity V_i characterises the LAB frame as having velocity V_i relative to the CM frame. The CM frame acts as the fundamental frame for studying reactions, as its special property of having a net momentum of zero makes it useful as a reference from which other frames can be defined. The LAB frame is characterised, not by conserving its velocity V_i, but by *preserving its total momentum relative to the CM frame*. The total momentum of the LAB frame is

$$P_{\mathrm{LAB}} = M_iV_i = (m_1 + m_2)V_i, \tag{8}$$

and P_{LAB} does not change over the course of the reaction.

While the reaction, from a detector's perspective, occurs in the LAB frame, transforming quantities from the LAB frame to the CM frame is more useful, because we can use the special properties of the CM frame to extract the quantities in the LAB frame that we need. Additionally, θ_{cm} is a more fundamental angle when analysing and comparing scattering experiments.

Once the reaction has taken place the CM frame will have a final kinetic energy, Tf, which is

$$Tf = Ti + Q - E_x = \frac{1}{2}m_3v_3^2 + \frac{1}{2}m_4v_4^2, \tag{9}$$

where Q is the energy released in the reaction (this term takes the mass differences into account), and is

$$Q = (m_1 + m_2) - (m_3 + m_4). \tag{10}$$

E_x is the excited state of the recoil nucleus. Note that the ejectile is considered to be in its ground state[1].

Using momentum-conservation from the definition of the CM frame, it can be shown that

$$m_3v_3 + m_4v_4 = 0 \tag{11}$$

so, using Equation 9,

$$Tf = \left(\frac{m_3 + m_4}{m_4}\right)T3. \tag{12}$$

The ejectile velocity, v_3, can be obtained using Equations 9 and 12:

$$v_3 = \sqrt{\frac{2m_4\left(Ti + Q - E_x\right)}{m_3\left(m_3 + m_4\right)}}. \tag{13}$$

Since the reaction has taken place *and* since mass is not conserved (but mass-energy is always conserved!), the velocity of the LAB frame must be considered more carefully. As P_{LAB} is a constant, then

$$P_{LAB} = M_iV_i = M_fV_f \Rightarrow (m_1 + m_2)V_i = (m_3 + m_4)V_f, \tag{14}$$

$$\tag{15}$$

so

$$V_f = \frac{m_1 + m_2}{m_3 + m_4}V_i, \tag{16}$$

which in practice is almost identical to V_i.

At this point, the treatment has been identical for NK and IK. However, Figure 1 highlights the differences that occur when calculating the kinematics in these two situations.

While the ordering of the beam and target are different in NK and IK, the ejectile remains the same,

[1]This is true for many light ejectiles, such as p, d, t, and 3He which do not have bound excited states. While α-particles do have excited states, these are above 20 MeV, so are energetically impossible for typical reactions.

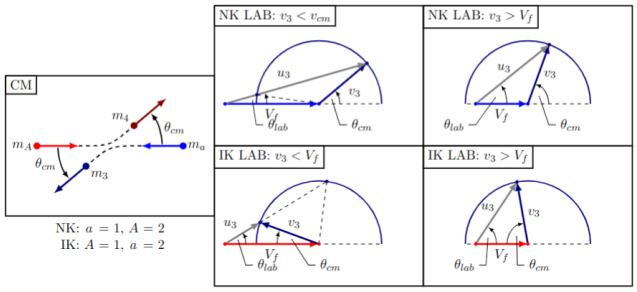

Figure 1: Drawings of the vector triangles formed when transforming from CM to LAB frame in NK and IK. m_a (m_A) is the initial, lighter (heavier) nucleus in the reaction. The vector V_f has been coloured to match the nuclide acting as the beam particle.

which changes the relative direction of the centre-of-momentum velocity relative to the ejectile velocity along the beam axis. Thus, a shallow scattering angle in the CM frame may correspond to a large θ_{lab}.

Also, the size of v_3 relative to V_f is important. When $v_3 < V_f$, θ_{lab} is doubly-valued, and the ejectiles are focused into a cone with

$$|\theta_{cm}| \arccos\left(\frac{v_3}{V_f}\right). \tag{17}$$

When $\frac{v_3}{V_f} > 1$, then θ_{cm} has a single value for every θ_{lab} and ejectiles can be emitted in either the forwards or backwards hemisphere.

In the case of elastic scattering, where beam and target are the same as ejectile and recoil, and where $E_x = 0$, then $\frac{v_3}{V_f} = 1$, and $\theta_{lab} = \frac{1}{2}\theta_{cm}$.

Examining the fraction

$$\frac{v_3}{V_f} = \sqrt{\left(\frac{m_3 + m_4}{m_1 + m_2}\right) \cdot \frac{m_2 m_4}{m_1 m_3} \cdot \left(1 + \frac{Q - E_x}{Ti}\right)}, \tag{18}$$

shows that the most significant term is $\frac{m_2 m_4}{m_1 m_3}$, which determines which of these regimes occurs for a given reaction.

Proceeding carefully, Figure 1 can be used to calculate the energy of the ejectile in the LAB frame:

$$u_3^2 = v_3^2 + V_f^2 \pm 2 v_3 V_f \cos\theta_{cm}, \tag{19}$$

$$\Rightarrow T_3 = T3 + \frac{1}{2} m_3 V_f^2 \pm m_3 v_3 V_f \cos\theta_{cm}, \tag{20}$$

with the positive (negative) sign in this equation corresponding to NK (IK).

So far, the primary distinction between normal kinematics (NK) and inverse kinematics (IK) lies in the definition of angles. However, when transitioning to IK, two additional effects emerge, which are absent in NK.

The first effect, known as the kinematic shift, becomes apparent when comparing regions with equivalent angular sizes in both regimes. Such a comparison corresponds to the angular acceptance of a detector when utilizing a fixed-angle setup. Notably, in IK, the kinematic lines exhibit a significantly broader energy range than in NK, leading to widened peaks in an excitation-energy spectrum.

The second effect, termed kinematic compression, manifests itself in the low θ_{cm} region, where reaction cross sections generally exceed those at higher θ_{cm}. For the given reaction, NK and IK exhibit low θ_{cm} at opposing extremes of the θ_{lab} spectrum. The result is a considerably diminished separation between states in IK compared to NK, specifically within an equivalent θ_{cm} range. This phenomenon reduces the gap between peaks in the excitation-energy spectrum.

While these effects can be mitigated through coincident gamma-ray measurements alongside ejectiles, solenoidal spectrometry presents a sophisticated solution to this challenge.

2.1.1 Kinematics in a solenoidal field

Measurements of nuclear reactions that provide the most interesting information are often challenging to carry out due to the low intensity of short-lived beams. Direct reactions are important because of their selective nature and the relative simplicity of understanding the experimental observables. Single-nucleon transfer reactions, such as (d,p), (α, t), or $(^3He, d)$, particularly probe the basic single-particle structure of nuclei. These studies have been crucial in establishing the framework for understanding stable nuclei in the past, and they are regaining significance in exploring nuclei away from stability using the short-lived beams available at accelerators worldwide. These initial steps are fundamental for studying the more exotic nuclei that will be accessible with future rare isotope beam facilities.

The utility of solenoidal spectrometers for investigating short-lived nuclei under inverse kinematics (IK) was initially outlined in reference [11]. It effectively addresses the issue of kinematic compression compared to the same reaction conducted in IK with a fixed-angle setup. By measuring the position where a particle intersects the beam axis instead of the scattering angle, kinematic lines from two distinct states can be more easily differentiated. From this point, we will

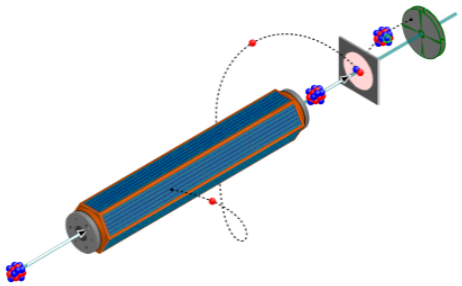

Figure 2: An illustration of a transfer reaction in a solenoidal spectrometer

focus on the most prevalent reaction within solenoidal spectrometers, namely the DP reaction in IK.

Here, the ejectile proton is emitted in the upstream hemisphere from the target, executing a helical orbit that intersects with a detector centered on the beam axis as shown in Figure 2. This helical trajectory arises due to the Lorentz force law. For an ejectile with charge q in a uniform magnetic field B, the centripetal force it experiences while moving with velocity u_3 is given by

$$qu_3 \times B = qu_\perp B = \frac{m_3 u_\perp^2}{R}, \tag{21}$$

where R represents the radius of the helical orbit, and $u_\perp$ is the component of the ejectile velocity perpendicular to the beam axis. The parallel component $u_\parallel$ can be expressed in terms of V_f and v_3 as

$$u_\parallel = u_3 \cos\theta_{lab} = V_f - v_3 \cos\theta_{cm}, \tag{22}$$
$$u_\perp = u_3 \sin\theta_{lab} = v_3 \sin\theta_{cm}. \tag{23}$$

The radius R can be calculated as

$$R = \frac{m_3 u_\perp}{qB} = \frac{m_3 u_3 \sin\theta_{lab}}{qB}. \tag{24}$$

Using Equation 24, the time period for one complete orbit, t_{cyc}, is given by

$$t_{cyc} = \frac{2\pi R}{u_\perp} = \frac{2\pi m_3}{qB} = \frac{2\pi}{\omega}, \tag{25}$$

which is identical for like particles, enabling particle identification with precise timing resolution[2].

The distance along the beam axis, z, is then given by

$$z = u_\parallel t_{cyc} = \frac{2\pi m_3}{qB} \cdot u_3 \cos\theta_{lab}, \tag{26}$$

[2]A 'start' time would also need to be recorded to distinguish particles; this could be obtained from the recoil detector.

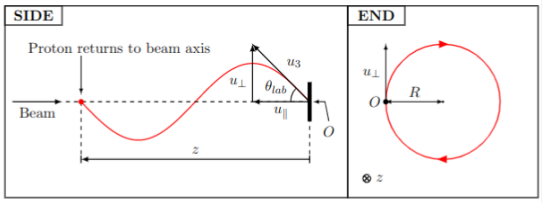

Figure 3: A 2D view of the helical orbit, viewed along and perpendicular to the beam axis.

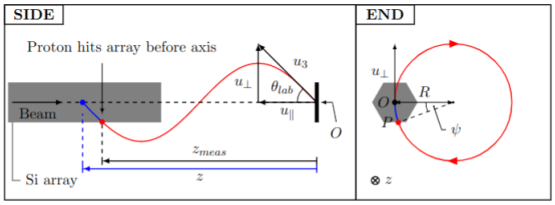

Figure 4: A 2D view of the helical orbit, viewed along and perpendicular to the beam axis.

and this can be substituted into Equation 20 as

$$T_3 = T3 - \frac{1}{2} m_3 V_f^2 + \frac{m_3 V_f z}{t_{cyc}} \tag{27}$$
$$= T3 - \frac{1}{2} m_3 V_f^2 + \frac{qBV_f z}{2\pi}. \tag{28}$$

Particle's trajectory can be illustrated as it is provided in the Figure **??**

2.1.2 Kinematics in a solenoidal field with a finite-size detector

In the presence of a finite-size detector, the ejectile's helical orbit is incomplete, spanning an angle $2\pi - \psi$ as shown in Figure 4.

The ISS silicon array can determine the z position of the proton on the array, denoted as z_{meas}, and provide an approximate measured radius, ρ_{meas}, based on which side of the strip (p-side or n-side) fired. The angle ψ can be determined with

$$\psi = 2 \arcsin\left(\frac{\rho_{meas}}{2R}\right). \tag{29}$$

However, the value of R relies on θ_{lab} from Equation 24, which is itself derived from z using Equation 26. Thus, z cannot be analytically solved but must be extracted using a minimization process.

The Butler algorithm [12] employs a minimization approach to extract z. It minimizes the function

$$f(z) = \left(1 - \frac{\psi}{2\pi}\right) z - z_{meas} \tag{30}$$

utilizing measurements of the ejectile energy T_3 and the measured array position z_{meas}. This yields $u_3 = \sqrt{\frac{2T_3}{m_3}}$, and combining this with the initial estimate of z can provide θ_{lab} using Equation 26. Subsequently, R can be determined using Equation 24, and then ψ using Equation 29 for minimizing Equation 30.

2.2 Design and Architecture of the Web Interface

In the rapidly evolving landscape of scientific research, the integration of web-based toolkits has emerged as a powerful approach to streamline data analysis, visualization, experiment optimization, and collaboration [13]. This work explores the advantages of incorporating HTML, CSS, and JavaScript (JS) in the context of nuclear physics research. A useful toolkit will be developed and used for optimizing researchers' work. Nevertheless, our toolkit aims to provide researchers with a comprehensive platform for refining experimental parameters and predicting outcomes, ultimately guiding them toward more successful and impactful proposals.

By harnessing the potential of these technologies alongside Python and Flask for server-side connectivity, researchers can enhance their capabilities, foster collaboration, and facilitate efficient data dissemination.

In our research endeavor, the selection of programming languages and frameworks was undertaken to ensure optimal efficiency and compatibility for both the front-end and back-end aspects of our project. For the front-end development, we judiciously chose JavaScript, HTML, and CSS as our primary programming languages and frameworks.

JavaScript, with its versatile capabilities, enabled us to create interactive and dynamic web applications that seamlessly engage users. HTML and CSS, on the other hand, facilitated the structuring and styling of our web interfaces, enhancing user experience by providing a visually appealing and intuitive layout.

For the back-end components of our project, we recognized the significance of Python as an ideal choice. Python's widespread adoption within scientific communities, including prominent institutions like CERN, ensured seamless integration with the scientific tools and libraries essential for our research in nuclear physics.

2.3 Frontend Development

As mentioned, we have proceeded with using HTML, CSS, and JS for our development work. Visual representation of frontend and backend division is shown in Figure 5

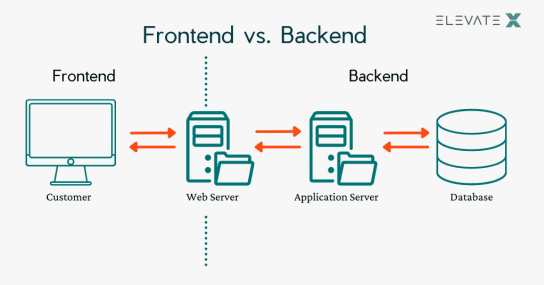

Figure 5: Backend and Frontend visualisation

HTML (Hypertext Markup Language) is the cornerstone of web content, allowing researchers to structure their data and findings in an accessible format. By employing HTML, nuclear physicists can create intuitive user interfaces for data input, experiment control, and result visualization [14]. The structured nature of HTML ensures consistency and clarity in presenting research methodologies and outcomes.

CSS (Cascading Style Sheets) complements HTML by enabling researchers to apply visually appealing styles to their web interfaces. In the realm of nuclear physics research, CSS can enhance the presentation of complex data, graphs, and plots. Customizable styling through CSS enables scientists to tailor their interfaces to specific research objectives, enhancing user experience and facilitating data interpretation.

JS, [15] serves as the engine behind dynamic and interactive web content. By incorporating JS libraries, such as D3.js or Plotly.js, nuclear physicists can create interactive visualizations of complex datasets, enabling deeper insights and facilitating data-driven decision-making [16]. These visualizations offer an intuitive way to explore experimental results, enhancing the accessibility of research findings to a broader audience.

2.4 Backedn Developemnt

Python, a popular programming language in scientific research, can seamlessly integrate with web technologies through the Flask framework. Flask facilitates the development of dynamic web applications by connecting the frontend (HTML, CSS, JS) with backend Python functions [17]. This integration empowers researchers to leverage their existing Python expertise while harnessing the advantages of web-based toolkits.

In Flask, you can store information specific to a user for the duration of a session. Saving data

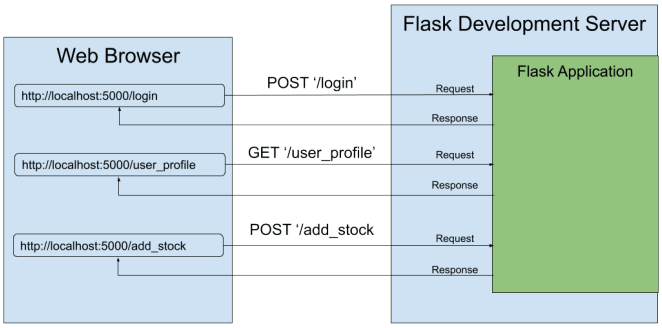

Figure 6: Flas- Web Browser and connection the Development Server.

for use throughout a session allows the web app to keep data persistent over multiple requests – i.e., as a user accesses different pages within a web app. When you open your browser and navigate to http://127.0.0.1:5000/, an HTTP (HyperText Transfer Protocol) request is sent from the browser to our Flask app on the server as it is illustrated in Figure 6

The adoption of web-based toolkits, including HTML, CSS, JS, and Python Flask [18], offers nuclear physicists a versatile and efficient approach to conducting research, analyzing data, and sharing findings. These technologies enable the creation of user-friendly interfaces, interactive visualizations, and collaborative platforms that contribute to advancing the field of nuclear physics. By embracing these web-based solutions, researchers can propel their studies forward while promoting accessibility and engagement within the scientific community.

3 Design and Implementation

By incorporating advanced computational techniques and accurate models of the ISS and transfer reactions, in detail explained in **??** the web-based tool would provide researchers with a comprehensive and user-friendly interface to explore different experimental scenarios. It would allow an iterative proposal refinement, considering various combinations of parameters to maximize the scientific yield of their experiments.

Certainly, here's a revised version of your explanation with improved English and formatting:

We have developed a comprehensive system that encompasses both JavaScript (JS) and Python scripts to facilitate our research objectives. The Python scripts, housed in the `calculations.py` module, serve as the backbone for executing crucial calculations pivotal to our study. These calculations are intricately linked to the server via Flask, a micro web framework for Python. This connection is established in the `main.py` script, which acts as the bridge between the computational backend and the user interface.

Within the `main.py` script, two essential lines of code at the start and end signal the integration of the server and calculations, unifying the system's functionality.

Our overarching goal centers on empowering users with the capability to extract meaningful information. Particularly, we provide users with access to θ_{cm} values based on input parameters. These parameters can be effortlessly tailored to mimic experimental scenarios. The focal motivation behind this endeavor lies in acquiring θ_{cm} values for predefined input values, specifically catering to the adjustable parameters in the experiment. By achieving this, we gain insights into the angular coverage of a detector, enabling us to discern the diverse states within the residual nucleus.

This versatile framework thus empowers researchers to fine-tune parameters, observe their effects on θ_{cm} values, and subsequently decipher the angular coverage of detectors in various scenarios.

The toolkit's backend integrates nuclear mass data using the $read\_nuclear\_mass\_table$ function from the $data\_utils\ module$. The data is loaded into a pandas DataFrame and reset for ease of manipulation. This data serves as the basis for subsequent calculations and analyses.

A core feature of the toolkit is the ability to calculate nuclear mass values. The $calculate\_nuclear\_mass$ route, triggered by a **POST request**, receives a list of particles with **'a'** and **'z'** values. The backend function $extract\_nuclear\_mass$ queries the DataFrame for specific 'a' and 'z' values, returning the corresponding nuclear mass. If no data is found, an appropriate error message is generated. The calculated mass values are then returned to the frontend as a **JSON response**.

The toolkit also enables the calculation of particle velocities in nuclear reactions. The $calculate\_v3$ route receives particle masses, excitation energy, and kinetic energy as inputs. The backend function computes the velocity of the third particle using relevant equations and returns the result to the frontend.

3.1 Calculation of Theta Values based on Detector Setup

3.1.1 Point-like Detector

In the initial phases of our project, we chose a simplified strategy to ensure the effectiveness and coherence

of our code. To achieve this, we treated our detector as a conceptually compact entity, which allowed us to create straightforward functions for our analysis. Our primary focus revolved around determining the most suitable theta values, guided by the user's input for the variable z.

Expanding on this concept, our simulation centered on a scenario where a proton interacts with the detector. To facilitate computational ease, we approximated the impact of the proton as occurring directly along the beam axis. This simplification aimed to streamline our calculations. In our comprehensive discussion of the physics background, outlined within its dedicated chapter, we introduced the variable z_{meas}. Under the aforementioned simplified conditions, z_{meas} was directly comparable to the input z. It's crucial to acknowledge that this simplification does not mirror the intricate nature of real-world scenarios.

As our project advances, our trajectory involves refining our calculations by embracing more precise and intricate considerations. This encompasses the incorporation of genuine particle behavior upon interacting with the detector, accommodating potential deviations from the beam axis, and addressing the genuine spatial distribution of impact events. While our initial work embraced simplifications for validation purposes, our forthcoming phases will usher in heightened accuracy and precision within our calculations, striving to faithfully replicate real-world complexities.

In this section, we describe the implementation of a route that calculates the theta values based on specific input parameters. The route is defined as follows:

```
@app.route("/given_z_find_theta", methods=['
    POST'])
def doCalculationsGivenZtoTheta():
```

This route is configured to listen for POST requests at the URL `/given_z_find_theta`. It is designed to perform calculations related to the determination of theta values in the context of a particle detector setup.

The input data for the calculation is extracted from the incoming JSON request as follows:

```
data = request.get_json()
```

The JSON data is expected to contain the following parameters: `mass1`, `mass2`, `mass3`, `mass4`, `v_3`, `T_1`, `z`, and `B_value`.

The core calculation is performed using the function `given_z_find_theta`, which is invoked as follows:

```
result = given_z_find_theta(
    m1=data['mass1'],
    m2=data['mass2'],
    m3=data['mass3'],
    m4=data['mass4'],
    v_3=data['v_3'],
    T_1=data['T_1'],
    z=data['z'],
    B_value=data['B_value']
)
```

The `given_z_find_theta` function calculates theta values based on the provided input parameters. This calculation involves various physical parameters, such as `m1`, `m2`, `m3`, `m4`, `v_3`, `T_1`, `z`, and `B_value`, along with certain constants like `q` (elementary charge) and `c` (speed of light). The calculation entails determining initial and final velocities (`V_i` and `V_f`) as well as the time of cyclotron motion (`t_cyclotron`). These values are then utilized to compute the angle `theta_cm` using trigonometric functions, providing insights into particle behavior within the detector setup.

The calculated theta value(s) are returned as the result:

```
return jsonify(result)
```

The resulting value(s) are then converted into JSON format and returned as the response to the frontend for further analysis or display.

This functionality allows users to explore the impact of changing input parameters, particularly the value of z, on the resulting theta values within the context of a particle detector setup.

3.1.2 Finite-size Detector

As our project progressed, we transitioned from the point-like detector approximation to a more intricate scenario that reflects real-world conditions. In a practical detector, particles interact over a finite area, introducing complexities that need to be addressed. This involved recalibrating our calculations to consider the actual spatial distribution of impacts and particle interactions on the detector's surface

With the incorporation of realistic detector dimensions, we encountered a challenge in obtaining optimal theta values for a given set of measurements. Unlike the point-like detector case, solving for theta values analytically was no longer feasible due to the increased complexity. To address this, we employed a numerical minimization technique to iteratively calculate optimal theta values that minimized the difference between calculated and measured z values.

We developed a minimization algorithm that iteratively adjusted theta values to match the measured z values on the detector's surface. The algorithm accounted for various factors such as particle velocities, masses, and magnetic field strengths. It aimed to minimize the discrepancy between calculated and measured z values, effectively improving the accuracy of our simulation.

```
# Extracted from main.py
@app.route('/find_optimal_theta', methods=['
    POST'])
def doCalculationsforOptimalTheta():
    # Data extraction from request
    data = request.get_json()
    # Applying the optimization function
    result = find_optimal_theta(
        # ... input parameters ...
    )
    return jsonify(result)
```

Our minimization function, $find\_optimal\_theta$, dynamically adjusted theta values to converge towards optimal solutions. It incorporated real-world factors such as particle velocities, mass, magnetic field strength, and detector dimensions. The goal was to iteratively find theta values that best matched the measured z values, considering the intricate interaction dynamics on the detector's surface.

```
# Extracted from calculations.py
def find_optimal_theta
(z_meas, V_f, v_3, m_3, ro_measured1,
ro_measured2, initial_theta_lab, B,
tolerance=1e-6, max_iterations=100):
    # ... parameter processing ...
    optimal_thetas = []
    for ro_meas in [ro_measured1,
    ro_measured2]:
        # Gradient descent optimization
        theta = initial_theta_lab
        for iteration in range(max_iterations
    ):
            # Calculate z_meas using the
            Provided formula
            # Calculate the difference
            between calculated z_meas and
    given
            z_meas
            # Calculate gradient
            using finite difference
            theta -= gradient * tolerance
            if abs(difference) < tolerance:
                break
        optimal_thetas.append(theta)
    return optimal_thetas
```

The function takes various parameters, including the measured z values (z_{meas}), particle velocities (V_f, v_3), particle mass (m_3), measured radii ($\rho_{measured1}$), ($\rho_{measured2}$), initial theta in the lab frame ($initial_{\theta_{\text{lab}}}$), magnetic field strength (B), and other parameters related to convergence criteria.

The algorithm proceeds as follows:

1. For each measured radius ($\rho_{measured1}$) and ($\rho_{measured2}$), the function performs gradient descent optimization to iteratively adjust the theta value.

2. The calculated z_{meas} is compared to the given measured z value. The difference and gradient are calculated using finite differences.

3. The theta value is updated based on the gradient and a specified tolerance value.

4. The iteration continues until the absolute difference between the calculated and measured z values is within the defined tolerance.

5. The optimal theta values are collected for both measured radii.

The output of this function is a list of optimal theta values for each measured radius. These values represent the angles that best match the measured z values on the detector's surface, accounting for the complexities of particle interactions, velocities, and detector dimensions.

This iterative optimization process enables us to refine our simulation, making it more accurate and reflective of real-world scenarios where particles interact with a detector of finite dimensions. The calculated optimal theta values serve as a valuable output for analyzing the angular coverage and behavior of particles in our experimental setup.

4 Results and Validation

We have seamlessly integrated the Python backend functions and codes, which we previously elucidated, with the frontend interface to provide a cohesive and user-friendly experience on our web platform. Our approach, designed as a streamlined one-page toolkit, centers around a single HTML file and an accompanying CSS file. Additionally, several JavaScript files have been developed, each dedicated to distinct functions that were pre-established within the `calculations.py` framework.

To elaborate, our web application's core functionality is achieved by a modular system composed of the aforementioned components. Among these, the

JavaScript files play a pivotal role. Notably, the `data.js` and `retrieve_mass.js` scripts manage data handling and mass retrieval operations, respectively. Additionally, the `v3.js` script is responsible for conducting calculations related to the variable `v_3`, while the `z_theta.js` and `optimal_theta.js` scripts cater to the determination of optimal theta values, accounting for both point-like and finite-sized detectors.

This stratified structure ensures efficient code management and enhances maintainability as our project evolves. Further insights into these intricacies and the source code itself can be explored in our dedicated Git repository.

It is worth mentioning, that when working with finite size detector we have hexagonal shape and in each angle we have silicon array detector. so our particle can impige on the flat line or in the corner, see figure 4, thus we can have two different values for our $\rho_{measured}$. And then corresponding to this we can have two different values for our θ_{cm} as it is real output of our function.

We have been using equations provided in order to calculate needed quantities. Considering, firstly, pointlike detector, we proceeded in the following way:

We harnessed Python to manipulate atomic mass data and compute nuclear masses for three particles. By applying conservation laws, we derived the nuclear mass of the fourth particle. These calculated values are presented to users through a button press, revealing the results in a convenient popup window. Furthermore, these values are automatically populated into the subsequent form for streamlined calculations.

Central to this functionality is the JavaScript function `retrieveDataAndCalculateMassExcess`. This function orchestrates server interactions to obtain calculated mass excess values. Upon retrieval, it dynamically generates a popup window displaying the nuclear mass excess for each particle. Additionally, these retrieved values are inserted into corresponding form fields, expediting subsequent calculations.

The core portion of the JavaScript code is provided below, illustrating the essence of our implementation. For the complete code, please consult our Git repository.

```
// JavaScript function to retrieve and
    display mass excess values
function retrieveDataAndCalculateMassExcess(
    particleValues) {
    // Create overlay and popup elements
    const overlay = document.getElementById("
    overlay");
    const popup = document.getElementById("
    popup");
    // Fetch calculated mass excess values
    from server
    fetch("http://localhost:5500/
    calculate_mass_excess", {
        method: "POST",
        headers: {
            "Content-Type": "application/json
    ",
        },
        body: JSON.stringify(particleValues),
    })
    // Process response and display results
    .then(response => response.json())
    .then(massExcessValues => {
        // Create popup window components
        const closeButton = document.
    createElement("span");
        // ... Additional logic for popup
    window ...
        // Insert calculated values into form
     fields
        mass1 = document.querySelector('#
    particle1').value = masses['Particle 1'];
        // ... Additional logic for overlay
    and popup ...
    })
    .catch(error => {
        console.error("Error:", error);
    });
}
```

Through the integration of this functionality into our web application, users can seamlessly access and apply nuclear mass excess values, significantly enhancing the efficiency of their calculations and analyses. Figure 7 shows toolkit inputs for calculating nuclear masses while Figure 8 shows output, where all 4 particle masses are included.

Subsequently, the velocity is computed utilizing pre-determined masses alongside user-provided inputs of E_x (excitation energy) and B (magnetic field strength). Notably, our units are standardized to MeV and T, respectively, aligning with prevailing conventions in the realm of nuclear physics.

The `v3.js` script dynamically furnishes essential contextual information, enriching the environment within which calculations take place. Our toolkit gives acceptable output for the ejectile's velocity as shown in Figure 9.

Finally, our journey culminates in achieving the primary objective of our project: the calculation of angular values θ_{cm}. This computation occurs for both scenarios—the point-like detector and the finite-sized detector. The corresponding JavaScript files, $z_theta.js$ and $optimal\_theta.js$, respectively, encapsulate the

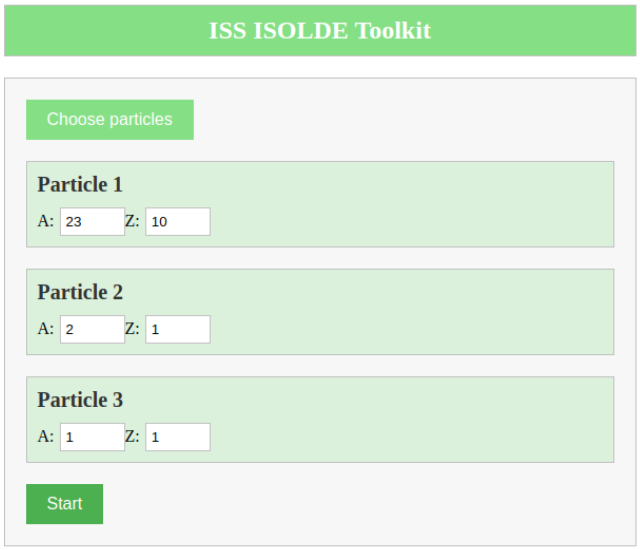

Figure 7: Toolkit input form used for calculating nuclear masses

Figure 8: Toolkit Output providing nuclear masses for 4 particles

logic for these computations, propelling us toward a comprehensive understanding of particle dynamics. This calculation is approached from two distinct scenarios: the finite-sized detector and the point-like detector. Below, we discuss the JavaScript functions responsible for orchestrating these calculations.

4.1 Theta Calculation from given Z - Point-Like Detector

In the point-like detector scenario, the `calculateThetaFromZ()` function is responsible

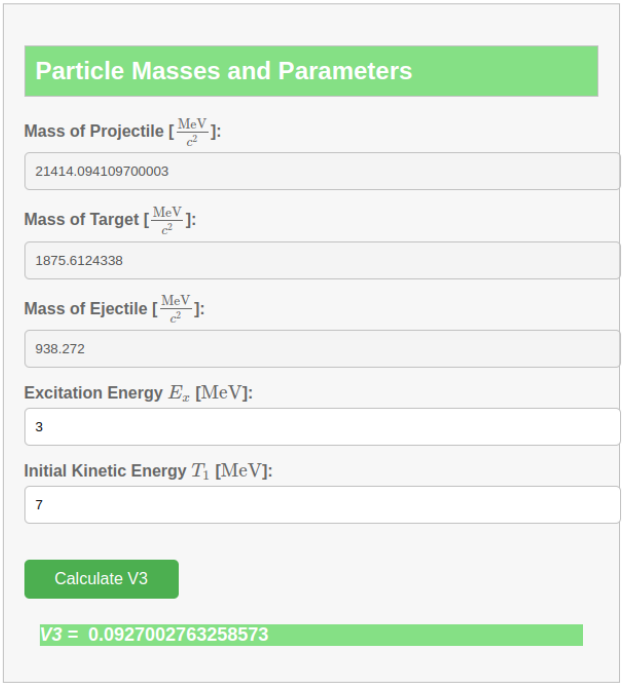

Figure 9: Calculations done on obtaining ejectile's velocity and obtained output

for deriving theta values based on the provided z-value. It interacts with the server using the `given_z_find_theta` endpoint. Similar to the previous function, a POST request is made with relevant data including particle masses, velocities, temperatures, z-value, and other parameters. The calculated theta values are then displayed after receiving and processing the response.

```
function calculateThetaFromZ() {
    // ... POST request and response handling
     ...
}
```

Both of these functions play a pivotal role in realizing our project's goal of comprehending particle dynamics by calculating and visualizing angular values. These functions form the bridge between our backend calculations and the frontend user interface, seamlessly harmonizing intricate computational processes for the end-users convenience. The inputs provided by the user and commonly used within ISS experiment and the corresponding output are shown in figure 11

4.2 Optimal Theta Calculation - Finite-Sized Detector

For the finite-sized detector case, the `optimal_theta_lab()` function orchestrates the calculation of the optimal theta values. It interacts with the server using the `find_optimal_theta`

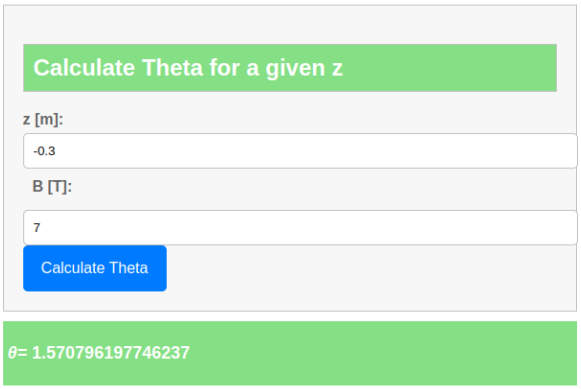

Figure 10: Theta calculation using simplified case and inputs users are able to provide

endpoint. The function sends a POST request with relevant data, including particle masses, magnetic field strength, measured radii, initial theta, charge, and other parameters. Upon receiving the response, it displays the calculated optimal theta values.

```
function optimal_theta_lab() {
    // ... POST request and response handling
     ...
}
```

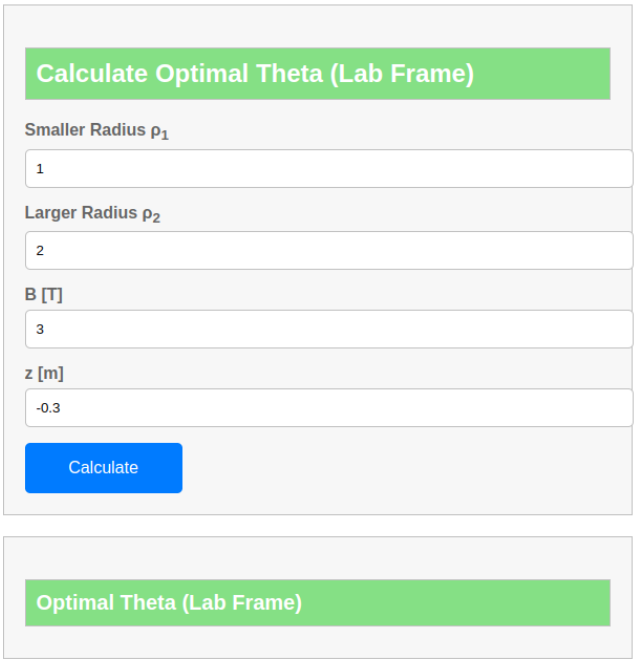

Figure 11: Theta values obtained for a given radii and magnetic field value where size of the detector is taken into account.

4.3 User Interface and Functionality

The toolkit's user interface is designed with user-friendliness and efficiency in mind. It features a toolbar that provides quick access to essential information. Users can easily navigate to the official CERN website, access contact information via email and phone number, and stay connected with the broader scientific community.

The toolkit ensures consistency by employing fixed units throughout its functionalities. For instance, the velocity output is presented in units of the speed of light, offering a relatable perspective for users. Underlying this feature is a sophisticated conversion mechanism that seamlessly transforms the output into the more commonly used SI unit of meters per second (m/s). This conversion enhances the toolkit's accessibility and utility, catering to a wider range of scientific and educational applications. While output for θ is given in radians as that is the default unit in `math` library in `Python`. Using available online converters [19], you can easily access θ values in degrees.

One of the noteworthy features of the toolkit is its educational potential. The toolkit serves as an effective educational aid, allowing users to delve into the intricacies of nuclear physics and gain insights into the properties of nuclei. Whether it's for students aiming to grasp fundamental concepts or researchers exploring the behavior of particles, the toolkit provides a versatile platform for both learning and analysis.

The carefully crafted user interface and the powerful functionalities offered by the toolkit make it a valuable tool for various purposes. From scientific investigations to educational endeavors, the toolkit empowers users to explore nuclear physics with confidence and precision. Its user-centric design and incorporation of essential information contribute to a comprehensive and enriching experience.

5 Conclusions and Future Enhancements

The developed toolkit has proven to be a powerful tool, capable of performing all essential calculations required for nuclear physics experiments. Its user-friendly design and simplicity make it accessible even to individuals without extensive web development experience, thereby enabling non-web-developer personnel working on ISS experiments to easily modify and adapt it. The toolkit's step-by-step explanations guide users through the process, ensuring a smooth experience.

We have taken meticulous care in defining the appropriate units for each calculation, emphasizing the importance of adhering to these units and preemptively handling conversions when necessary. This attention to detail ensures accuracy and consistency in experimental setups.

However, there are areas where the toolkit can be enhanced. Our current model assumes a thin target with no energy loss, overlooking important phenomena such as energy loss during particle interactions. Incorporating more sophisticated physics models, including relativistic mechanics, can improve the toolkit's accuracy and extend its utility to a wider range of experimental scenarios.

Looking ahead, we envision expanding the toolkit's capabilities beyond its current scope. Our aspirations encompass not only an extended multi-page toolkit but also the integration of online calibration and particle trajectory simulations. By providing these additional features, the toolkit could offer researchers a comprehensive platform for experiment planning and validation.

Moreover, the prospect of incorporating machine learning techniques holds promise for optimizing parameter selection in a more efficient manner. This advancement could significantly expedite the experimental design process and lead to more precise outcomes.

In essence, the web-based toolkit serves as a valuable educational resource, illuminating the intricate interplay between theoretical concepts and practical experimentation in nuclear physics. Its seamless fusion of physics principles with computational tools narrows the gap between theory and practice. By delivering real-time insights and feedback, the toolkit empowers researchers to enhance their experimental setups, fine-tune data collection strategies, and ultimately drive the progress of nuclear physics research.

As a culmination of collaborative efforts between physics, web development, and education, this toolkit marks a significant milestone in the field of nuclear physics research.

References

[1] E. Rutherford. "Phil. Mag. ser.6". In: 21 (1911), p. 669.

[2] H. Geiger, J. Harling, and E. Madsen. "Proc. Roy Soc. A82". In: (1909), p. 495.

[3] Fermi et al. "Proc. Roy. Soc. Ser A". In: 146 (1934), p. 483.

[4] S. N. Goshal. "Physical Review". In: 80 (1950), p. 939.

[5] K. S. Krane. "Introductory Nuclear Physics". In: (1987).

[6] A. Gallman et al. "Nuclear Physics". In: *Nuc. Phys.* 88 (1966), p. 654.

[7] W. E. Meyerhof. "Elements of Nuclear Physics". In: (1967).

[8] J.C. Lighthall et al. "Commissioning of the HELIOS spectrometer". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 622.1 (Oct. 2010), pp. 97–106. DOI: 10.1016/j.nima.2010.06.220.

[9] P. T. MacGregor et al. "Evolution of single-particle structure near the $N = 20$ island of inversion". In: *Phys. Rev. C* 104 (Nov. 2021), p. L051301. DOI: 10.1103/PhysRevC.104.L051301.

[10] B. P. Kay, C. R. Hoffman, and A. H. Wuosmaa. *SOLARIS. A Solenoidal Spectrometer Apparatus for Reaction Studies*. White Paper. Argonne National Laboratory, Mar. 2018. URL: https://www.anl.gov/reference/solaris-white-paper.

[11] A.H. Wuosmaa et al. "A solenoidal spectrometer for reactions in inverse kinematics". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 580.3 (Oct. 2007), pp. 1290–1300. DOI: https://doi.org/10.1016/j.nima.2007.07.029.

[12] P. A. Butler. "Enhancing the performance of solenoidal spectrometers for inverse reactions". In: *Proceedings of the Royal Society A* 479.2273 (2023), p. 20230075.

[13] John Smith. "Advancements in Nuclear Physics Research". In: *Journal of Nuclear Physics* 30.2 (2022).

[14] Robert Johnson and Emily Williams. “Interactive Data Analysis in Nuclear Physics using Web-Based Toolkits”. In: *Proceedings of the International Conference on Nuclear Physics*. 2023.

[15] *JavaScript*. Accessed on [insert access date here]. URL: https://www.javascript.com/.

[16] Alice Brown et al. “Web-Based Data Visualization for Scientific Research”. In: *International Conference on Scientific Visualization*. 2021.

[17] Michael Green et al. “Enhancing Collaboration and Dissemination in Nuclear Physics Research using Web-Based Interfaces”. In: *Journal of Nuclear Science* 45.3 (2020).

[18] Pallets Project. *Flask Documentation*. Accessed on August 11, 2023. 2023. URL: https://flask.palletsprojects.com/en/2.3.x/.

[19] *Radians to Degrees Conversion*. Accessed on [Date]. URL: https://www.rapidtables.com/convert/number/radians-to-degrees.html.