Upgraded Swimmer for Computationally Efficient Particle Tracking for Jefferson Lab's CLAS12 Spectrometer

Lydia Lorenti
Advisor: David Heddle

April 29, 2018

Abstract

The CLAS12 spectrometer at Jefferson Lab consists of a large suite of particle detectors that collect data used in the study of nucleon structure. One set of detectors, called drift chambers, record position data on charged particles that fly through them. In data analysis, software reconstructs the trajectories of these particles via a process called tracking, which involves fitting the theoretical trajectory of a charged particle in a magnetic field to the real position data. This theoretical trajectory is produced by a program called a swimmer. However, the swimmer that is currently used for CLAS12 particle tracking employs a mathematical algorithm that is not computationally fast enough to keep up with the large amount of data, creating a bottleneck problem. In this project, a new, faster swimmer has been created as a replacement for the old one. Along with a more efficient mathematical algorithm, it implements an optimized set of equations proposed in a paper from the HERA-B experiment at the DESY accelerator center in Germany. This causes the new swimmer to be far more computationally efficient than the old one, giving it a much higher computation speed than the old swimmer while maintaining the same accuracy of results.

1 Introduction

This project is meant to produce a faster data analysis tool for the CLAS12 spectrometer at Jefferson Lab. Speed is critical in data analysis at Jefferson Lab. Due to the vast amounts of data collected, high computational efficiency is required in order to avoid bottlenecks in data processing. The CLAS12 software team is currently facing such an issue. Below is a diagram of the CLAS12 spectrometer.

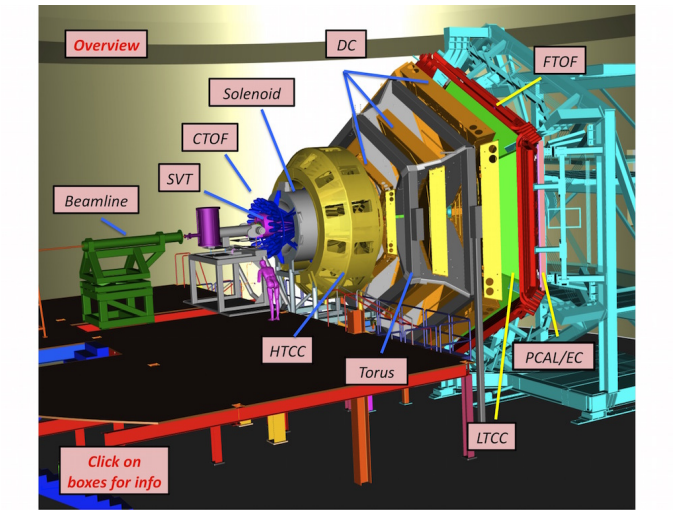

Figure 1: Diagram of CLAS12 Spectrometer [1]

During experiments, the electron beam from the accelerator collides with a stationary nuclear target placed at the front end of spectrometer. The collision causes relativistic particles to scatter off in a forward cone and enter the spectrometer. As the particles fly through the spectrometer, its various detectors record data on these particles. One particular detector system, the drift chambers or DC, records position data on charged particles as they pass through the magnetic field of a large torus magnet built around the DC system. Each drift chamber is a large, gas-filled box filled with fine, parallel, high-voltage wires. As a charged particle flies through the drift chambers, it ionizes the gas in its path. The free electrons in the particle's wake then collect around the positive high-voltage wires, creating electronic signals called "hits." These "hits" indicate the general location and direction of the particle's path.

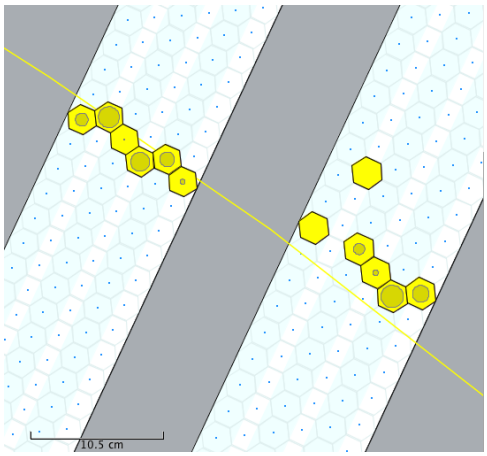

Figure 2: Diagram of Drift Chambers with Particle Trajectory and Hits

During data analysis, software called a tracker must reconstruct the trajectories, or tracks, of charged particles from the hits in the drift chambers. The tracking process is accomplished by fitting the theoretical prediction of a particle's path to the real hit data from the detector to produce a best-fit trajectory. The theoretical trajectory is obtained by solving the differential equations of motion of a charged particle in a magnetic field. However, the equations of motion for a charged particle in the highly non-uniform magnetic field of a torus have no analytic solution; therefore, the equations of motion must be solved by a numerical approximation. This numerical integration yields the predicted trajectory of the particle, and the fitting of the prediction to the data reproduces the actual trajectory of the particle.

The program within the tracker that creates the predicted trajectory is called a particle swimmer, or just a swimmer for short. The swimmer implements a numerical approximation method to solve the equations of motion for a charged particle in CLAS12's toroidal magnetic field. Currently, the swimmer uses a method called Euler's Method to solve for the trajectory. Euler's method is the simplest, most primitive method for approximating the solution to a differential equation. It begins at a known point, takes a small, fixed-size step in the independent variable (e.g. x), and approximates the value of the dependent variable (e.g. y) at the new location based on the slope of the graph at the initial location. Then the procedure repeats. Typically, when solving equations of motion for particles, the independent variable is the time t; however, the current swimmer uses the path length of the particle's travel as the independent variable. This means that each approximation of the particle's location along its path is made after the particle has traveled a certain distance from the previous approximation point. Due to the highly non-uniform field of the torus magnet, the curvature of a charged particle's path through the field varies significantly depending on the strength of the field in a given region. Therefore, the swimmer must take very small steps along the path when numerically integrating in order to achieve sufficient accuracy in the regions of highest curvature. When analyzing large quantities of data, this process is computationally intensive and time consuming, creating a computation bottleneck. Further compounding the problem, the tiny step size required in regions of high curvature is not necessary in regions of low curvature, so the swimmer spends excess computation time when approximating low-curvature sections of a particle's path. Consequently, the

CLAS12 software team wants to use a different swimmer that implements a better, faster, more efficient numerical approximation method. Another piece of CLAS12 software already contains another swimmer which implements a highly efficient approximation method called a Runge-Kutta method. However, this swimmer does not propagate the uncertainties in its approximation, which are required by the tracker. Therefore, this swimmer cannot be used to solve the tracking inefficiency problem.

While it might suffice to complete this other swimmer by implementing the propagation of its uncertainties and then using it to replace the old, inefficient swimmer, a paper for the German HERA-B experiment has proposed an even more efficient swimmer than the faster CLAS12 one [2]. Its main feature is that it uses the straight-line forward coordinate z as the independent variable of integration rather than the path length. This is useful because the drift chambers are oriented in parallel planes (Fig. 3). Consequently, it is logical to stop "swimming" (approximating the path) at a particular plane, such as the back of one of the drift chambers. The coordinate of this stopping plane is easily specified by the coordinate z, as the z-axis is perpendicular to the detector planes. However, since both existing swimmers use path length as the independent variable, z is currently a dependent variable. This means that the swimmer must check to see if it has reached the stopping plane after each step. If it passes the stopping plane, it must zig-zag back and forth across the stopping plane, checking the z-coordinate after each step, until the stopping plane is reached. This costs computation time. Using z as the independent variable, on the other hand, allows the stopping location to be determined by the value of the independent variable, which significantly simplifies the stopping process. Once the specified final z-coordinate is reached, the swimmer can stop running with no further checks. If it takes a step past the final z, it can retake that step straight to the final z without having to zig-zag like the existing swimmers. Therefore, it is more efficient to make z the independent variable rather than path length. Additionally, using z as the independent variable reduces the number of dependent variables that need to be approximated.

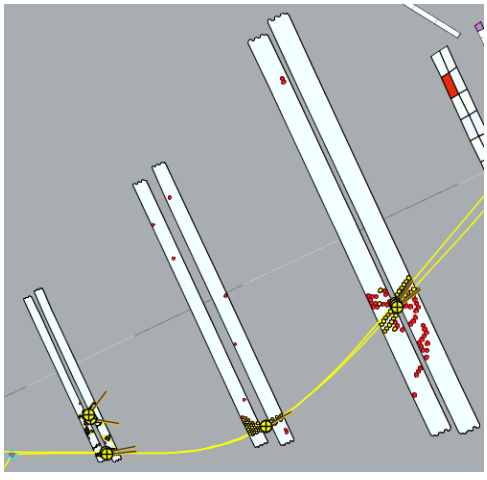

Figure 3: 2-D Cross Section of Drift Chambers with Track

Due to the increased efficiency of the swimmer described in the HERA-B paper [2], I have implemented it as a replacement for the old, slow swimmer. This new swimmer uses

the same implementation of the same Runge-Kutta approximation method as the existing faster swimmer. Runge-Kutta methods are highly accurate methods for approximating the solutions to differential equations, allowing them to achieve the same accuracy in far fewer steps than Euler's method. Additionally, the particular Runge-Kutta method used by the swimmer includes a feature called adaptive step size. Instead of taking fixed-size steps along the independent variable, the size of each step is adjusted so that the largest step possible is taken while maintaining the specified accuracy. This means that smaller steps are used in regions of high path curvature, while larger steps are used in regions of little to no path curvature. As a result, the adaptive step size feature enables the approximation to be computed to the desired accuracy in as few steps as possible, eliminating unnecessary computations.

The new swimmer was expected to meet two criteria. First, it was expected to produce results as accurately as the other two existing swimmers. Second, it was expected to be computationally faster than both the existing fast swimmer and, by implication, the old swimmer currently used in the tracker.

2 Theory

The most general vector equation of motion for a relativistic charged particle in a magnetic field is given by:

$$m\gamma\ddot{\vec{r}} = q(\vec{v} \times \vec{B}) \tag{1}$$

where m is the mass of the particle, γ is the Lorentz factor, $\ddot{\vec{r}}$ is the acceleration of the particle as a function of time, q is its charge, $\vec{v}$ is the particle's velocity, and $\vec{B}$ is the magnetic field. When solving this equation of motion, it is typically split up into a system of position components and velocity or momentum components. The component equations then describe the motion of the particle with respect to time t. However, for application in the swimmer, the equations of motion must describe the motion of the particle with respect to the z-coordinate. The time-dependent equations of motion can be converted to z-dependence by two changes of variable. The first change of variable is from the time t to the path length s via the relation $s = vt$. The second change of variable is from the path length s to the coordinate z via the following derivative relations:

$$ds^2 = dx^2 + dy^2 + dz^2 \tag{2}$$

$$ds = \sqrt{f}dz \qquad f = 1 + \left(\frac{dx}{dz}\right)^2 + \left(\frac{dy}{dz}\right)^2 \tag{3}$$

$$\frac{ds}{dz} = \sqrt{f} \qquad \frac{dz}{ds} = f^{-1/2} \tag{4}$$

These changes of variable result in the following equations of motion, taken from the HERA-B paper [2]:

$$\begin{aligned} dx/dz &= t_x, \\ dy/dz &= t_y, \\ dt_x/dz &= q \cdot \upsilon \cdot A_x(t_x, t_y, \vec{B}), \\ dt_y/dz &= q \cdot \upsilon \cdot A_y(t_x, t_y, \vec{B}), \\ q &= q_0, \end{aligned} \tag{2}$$

where parameter υ is proportional to the velocity of light and is therefore defined as

$$\upsilon = 0.000299792458 \; (GeV/c)\, kG^{-1}\, cm^{-1}$$

and the functions A_x,A_y are

$$\begin{aligned} A_x &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot [t_y \cdot (t_x B_x + B_z) - (1 + t_x^2) B_y] \quad , \\ A_y &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot \left[-t_x \cdot (t_y B_y + B_z) + (1 + t_y^2) B_x\right] \quad . \end{aligned}$$

This system of equations depends on the forward coordinate z. In this system, a particle has integer charge Q_e, momentum $\vec{p}$, and Cartesian coordinates (x, y, z). The state vector describing the particle's position and momentum at any given value of z is

$$\vec{\psi}(z) = (x, y, t_x, t_y, q) \tag{5}$$

where $t_x = p_x/p_z$, $t_y = p_y/p_z$, and $q = Q_e/|\vec{p}|$. Here, t_x and t_y are unitless versions of the transverse momentum components. Also, q is a constant, which is why $q = q_0$ in the equations of motion. Regarding units, the coordinates (x, y, z) are in centimeters, the momentum p is in GeV/c, Q_e is in integer units of the elementary charge e, and the magnetic field B is in kiloGauss.

3 Methods

The new z-dependent swimmer is implemented as an object-oriented Java software package, since the rest of the CLAS12 software is also written in Java. The swimmer contains a few overarching methods to perform the overall "swimming" process. The solution approximations for the equations of motion are computed by the existing code package that implements the adaptive-step-size Runge-Kutta approximation method as discussed previously. The swimmer also contains other objects to set up the equations of motion according to the initial conditions, store individual state vectors, and store the results of swimming a particle. The result of swimming a particle is a list of state vector objects describing individual points along the particle's trajectory.

The input to the swimmer program includes the integer charge of a particle, the magnitude of its total momentum, its initial state vector, the final z-coordinate for stopping, the initial step size for the numerical approximation, and the error tolerance for the approximation. The error tolerance is usually given as a power of 10, indicating the number of decimal places to which the approximation must be accurate. Accuracy is achieved when approximations at two different step sizes yield, to the required number of decimal places, identical solution values at a particular z-coordinate. When swimming a particle, the step size is adjusted as necessary to be as large as possible while producing solution values that,

within the specified tolerance, agree with the values produced at the next smaller step size increment. After each step, the solution state vector at that point is stored. Once the swimming process is complete, the swimmer returns the list of state vectors describing the trajectory of the particle.

The swimmer was tested for both accuracy and speed. Since the equations of motion for a charged particle in a *uniform* magnetic field have analytic solutions, the swimmer was tested for accuracy by running it with a uniform magnetic field instead of the actual toroidal one. The analytic solutions for the particle in a uniform magnetic field were implemented in code, and the results of the swimmer were compared to the results of the analytic solutions. To test the computation speed of the swimmer, its computation time was compared to the computation time of the existing fast swimmer. The toroidal magnetic field was used.

4 Data

The following four plots show the results of the accuracy testing for the new swimmer. Four different final parameters, x, y, θ, and ϕ, were compared between the swimmer and the analytic solutions. The parameters x, y are position coordinates, and the parameters θ, ϕ are angular coordinates describing the direction of the particle's momentum. θ is the polar angle measured from the positive z-axis, and ϕ is the azimuthal angle around the z-axis measured from the positive x-axis. The standard deviations of the parameter values resulting from the swimmer were computed relative to the corresponding parameter values resulting from the analytic solutions. These standard deviations were calculated at four different error tolerance values, and the following plots show the standard deviation of each parameter as a function of the error tolerance value. The tolerance value is plotted on a logarithmic scale.

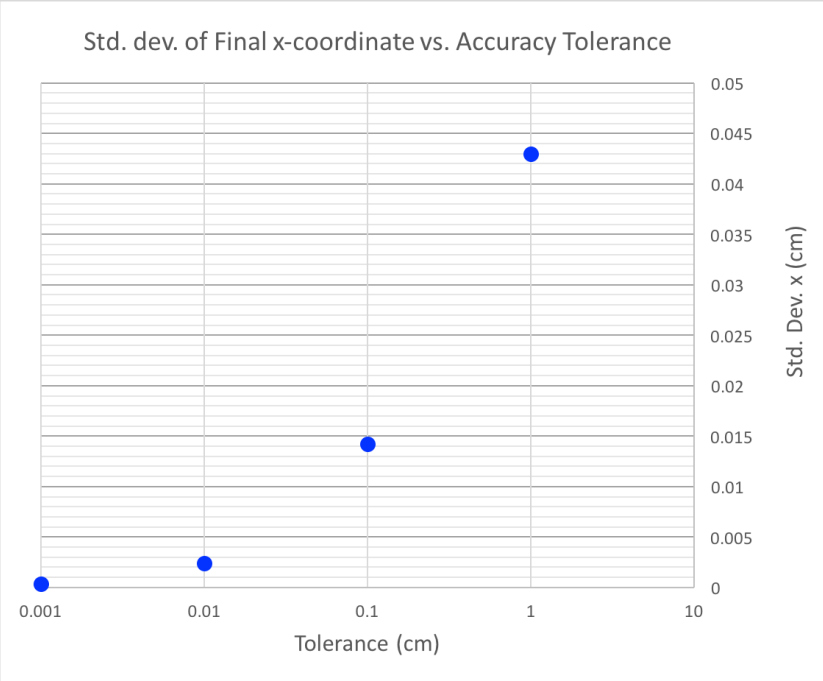

Figure 4: Accuracy Test Plot for Final x-value

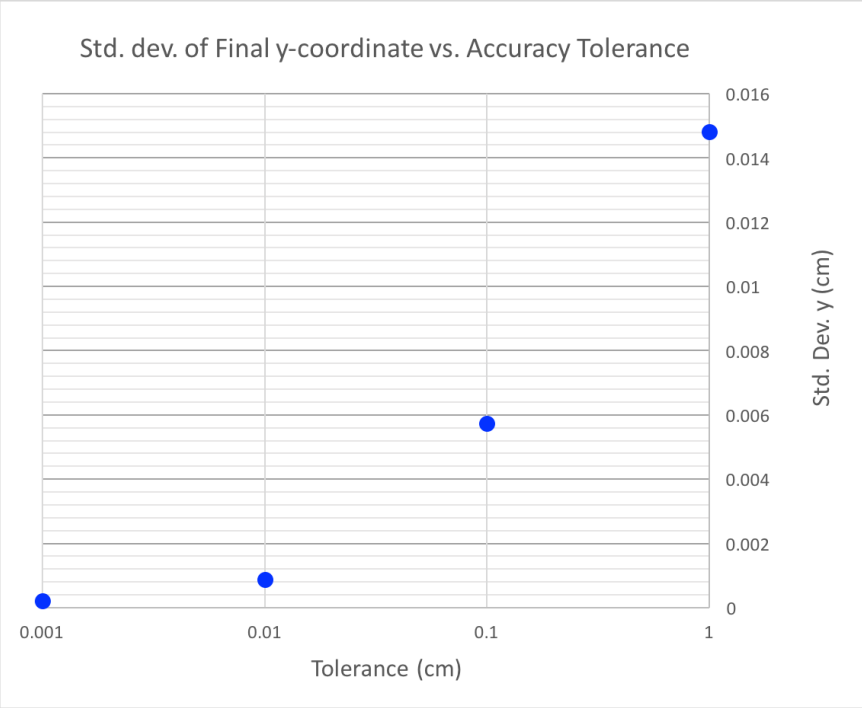

Figure 5: Accuracy Test Plot for Final y-value

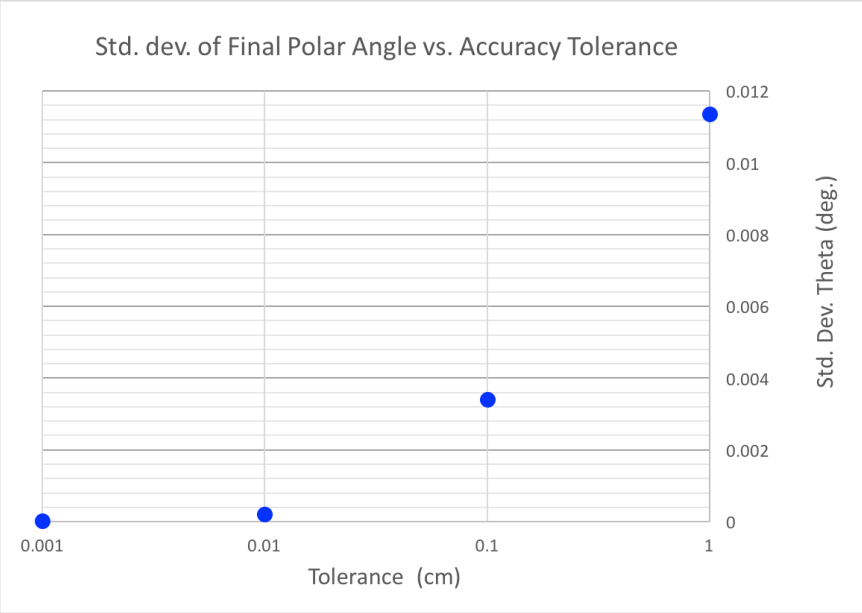

Figure 6: Accuracy Test Plot for Final θ-value

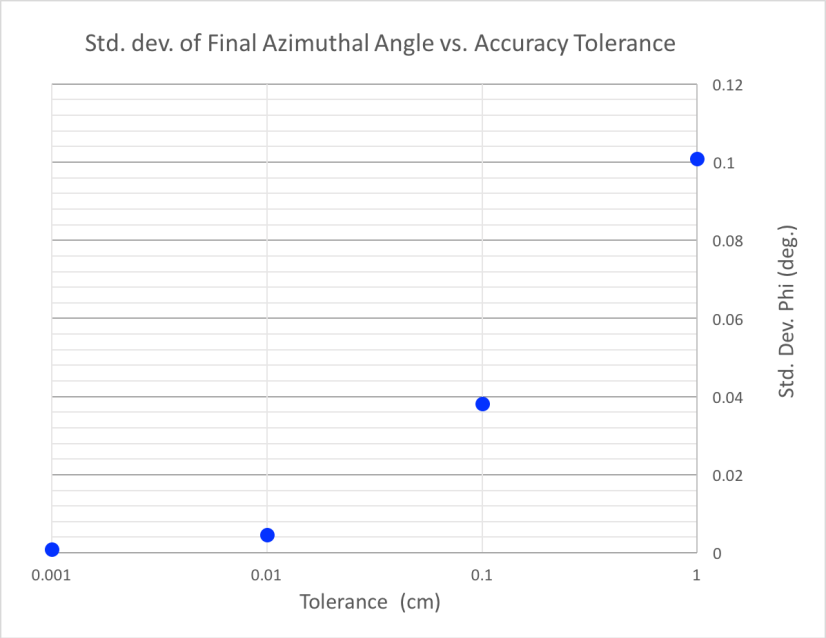

Figure 7: Accuracy Test Plot for Final ϕ-value

The following two plots show the results of the computation speed testing for the new swimmer. The computation speed of the new swimmer was compared with the computation speed of the existing fast swimmer over ten different distances in the z-direction at four different combinations of momentum and initial θ. The first plot shows computation time vs. z-distance at a momentum of 1 GeV/c, while the second plot shows the same at 2 GeV/c. Both plots contain data taken at initial angles of $\theta = 10^{\circ}$ and $\theta = 25^{\circ}$. It is worth noting that the label "Old Swimmer" in the plot legends does not refer to the slow swimmer that is being replaced, but to the existing fast swimmer.

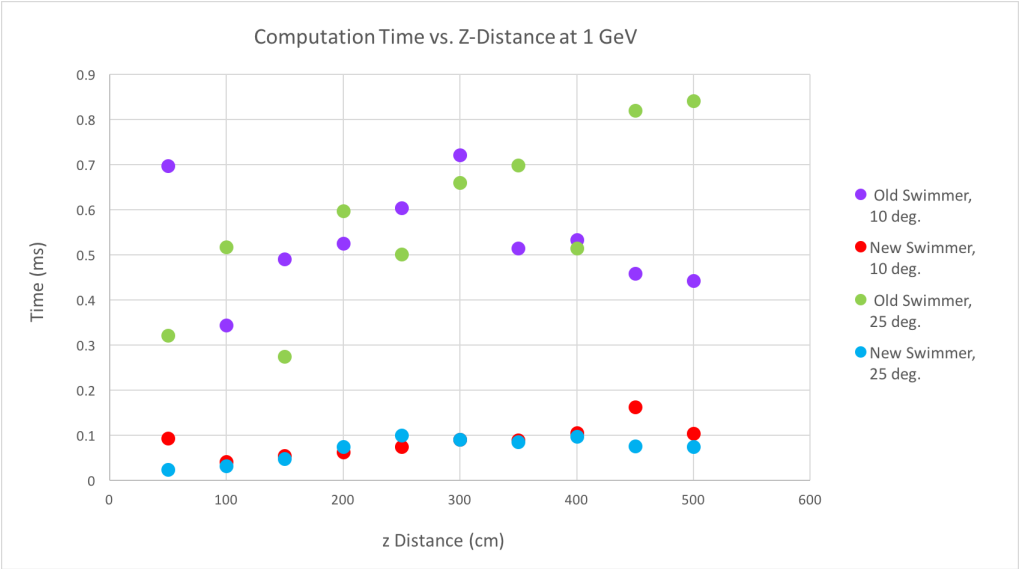

Figure 8: Speed Test Plot at $p = 1$ GeV

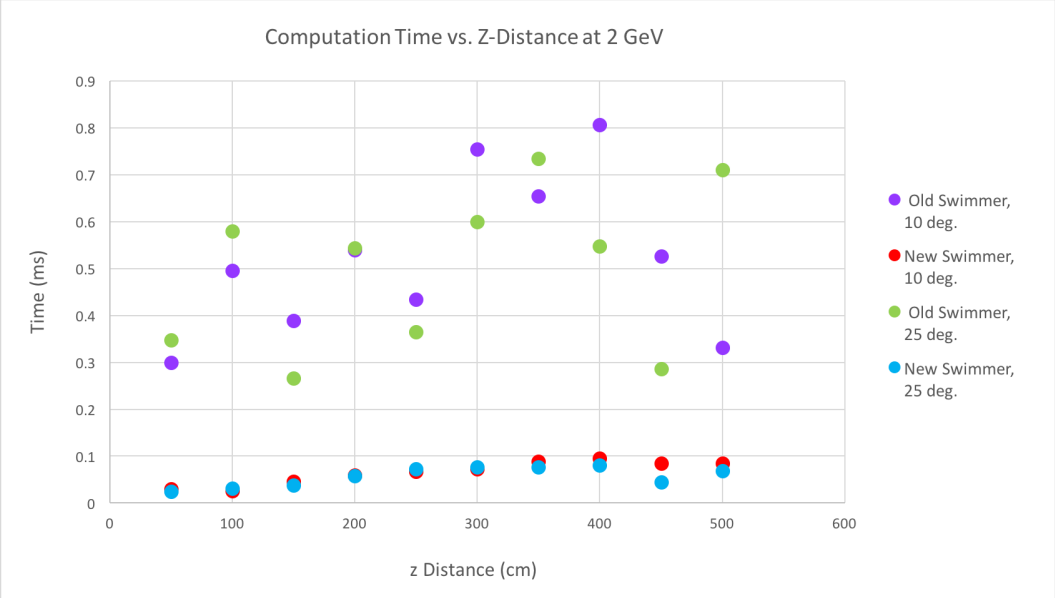

Figure 9: Speed Test Plot at $p = 2$ GeV

5 Discussion and Conclusions

The swimmer performs as expected. In the accuracy data plots (Figs. 4-7), the standard deviation increases as the error tolerance value increases. The currently accepted error tolerance for position is 10 μm = 0.001 cm. This corresponds to a numerical error tolerance value of $1 \times 10^{-3} = 0.001$, which is the smallest tolerance value on the accuracy plots. It is also desired that the swimmer be accurate to at least 1 in 1000 units. The plots show that this criterion is satisfied at an error tolerance of 0.001. This means that the new swimmer is sufficiently accurate at the accepted error tolerance level and is therefore consistent with the accuracy of the existing swimmers.

In the computation speed data plots (Figs. 8-9), the new swimmer (blue and red data points) is consistently faster than the existing fast swimmer (green and purple data points). Upon inspection, the plots reveal that the new swimmer takes about one order of magnitude less time to swim a particle a given distance than the existing fast swimmer. As expected, the new swimmer is computationally faster than both the existing fast swimmer and, necessarily, the old, slow swimmer that it is meant to replace.

It is of some interest to note that the initial direction of the momentum vector had no apparent effect on the computation time of either swimmer. There is also little to no correlation between computation time and distance traveled. Additionally, the existing fast swimmer, which is path-length-dependent, had a much wider range of computation times than the new z-dependent swimmer, with no apparent trend. This seemly random fluctuation is likely due to the zig-zagging behavior of the path-length-dependent swimmer when it locates the final stopping plane, as described in the Introduction. It is likely that the time taken by this zig-zagging behavior does not follow any consistent pattern, resulting in random fluctuations in computation time.

In its current state, the new swimmer still requires a small amount of cleaning up and minor debugging before it will be ready for use in other CLAS12 programs. Also, the propagation of uncertainties has not been implemented, so it is not yet ready to replace the old swimmer in the tracker. However, the HERA-B paper that supplied the equations of motion with respect to z also provides equations for the propagation of uncertainties [2]. Therefore, preparing the new swimmer for use in the tracker will only be a matter of implementing the equations provided; no new derivations will be required.

References

[1] "clas12-design.jpg". Jefferson Lab Experimental Hall B, CLAS12.
url: https://www.jlab.org/Hall-B/clas12-web/. Date of Access: 9/26/17

[2] Alexander Spiridonov. "Optimized Integration of the Equations of Motion of a Particle in the HERA-B Magnet." 07/16/1998. Revised 10/25/2005. PDF.
url: arxiv.org/pdf/physics/0511177.pdf