SANDIA REPORT
SAND2019-8669

Printed July 24, 2019

Sandia National Laboratories

Code Verification of a Warm Electron Diode Using the EMPIRE Plasma Simulation Code

Gregg A. Radtke

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

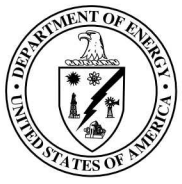
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

This report documents recent code verification exercises for a warm electron diode problem using the EMPIRE plasma code. This computationally expensive test was performed three times, including two different code versions and two different time integration algorithms, and the resulting code responses were analyzed for convergence to the analytical solution and orders-of-convergence using the StREEQ numerical error estimation tool. Significant deviations from the exact solution and expected orders-of-convergence, as well as changes in code behavior over time and due to choice of time integration algorithm were observed, illustrating the need to fix fundamental code issues as well as additional code verification testing before the code can be relied upon to accurately solve critical problems within its application space.

CONTENTS

LIST OF FIGURES

NOMENCLATURE

List of variables

Variable	Definition	Formula	Units
$\boldsymbol{B}$	Magnetic flux density		T
c_0	Vacuum speed of light	$1/\sqrt{\epsilon_0 \mu_0}$	m/s
Co	Courant number	$v_0 \Delta t / \Delta x$	
e	Elementary charge		C
$\boldsymbol{E}$	Electric field	$-\nabla \phi$ (electrostatic)	V/m
J	Current flux through the diode		$\mathrm{A/m^2}$
L	Diode gap length		m
m	Electron mass		kg
n	Electron number density		$\mathrm{m^{-3}}$
n_0	Electron emission number density		$\mathrm{m^{-3}}$
N_c	Number of particles per cell at emission conditions		
R	Number of simulations at a given discretization level		
t	Time		s
Δt	Time step		s
T_0	Electron emission temperature		V
$\boldsymbol{v}$	Electron velocity		m/s
v_0	Electron emission thermal velocity	$\sqrt{2eT_0/m}$	m/s
W	Particle weight		
(x, y, z)	Coordinates along (x), and normal to (y, z), the diode gap		m
β	Coefficients in the discretization error model for StREEQ		
γ	Order-of-convergence		
Δx	Element cell size		m
ϵ_0	Permittivity of free space		F/m
λ_D	Electron Debye length at injection conditions	$\sqrt{\epsilon_0 T_0/(e n_0)}$	m
Λ	Diode gap normalized by the electron Debye length L/λ_D		
μ_0	Permeability of free space		H/m
ϕ	Electrostatic potential		V
Φ	Normalized electrostatic potential		
ω_p	Electron plasma frequency	$\sqrt{e^2 n_0/\epsilon_0 m}$	rad/s

1. INTRODUCTION

Verification [3, 7, 8] is the process of assuring that a scientific computer code produces the correct solution to the system of equations it was designed to solve. Verification consists of two related processes: code verification and solution verification. In code verification, the code is rigorously tested against an exact solution to assure that: (i) the solution converges to the exact solution in the limit of increasing refinement of its discretization, and (ii) that it converges at the appropriate order for the implemented algorithms. In contrast, solution verification uses the code to solve a problem for a specific application at multiple levels of discretization, and predicts the fully-converged solution within estimated uncertainty bounds. The credibility of the solution verification step ultimately relies on the sufficiency of the code verification activities, where this sufficiency is judged in terms of both rigor and completeness. In particular, the completeness of a suite of code verifications should span appropriate physical parameter regimes, and include coupling between all the necessary physics relevant to the code application space.

Code verification itself occurs at multiple levels, which includes the intensive code verification exercises of which the work documented in this report is an example. In particular, this work analyzes a problem with a direct relationship to a critical application of the EMPIRE plasma code [2], and also tests coupling between different physics components rather than just testing a single part of the implemented code physics. It was also analyzed using the StREEQ numerical error estimation tool [4], which provided a greater degree of statistical rigor in comparison with standard approaches. Although these code verification exercises were computationally expensive, they were able to diagnose code issues in EMPIRE than other less-intensive tests failed to diagnose.

Other levels of code verification consist predominantly of nightly and weekly automated testing, which sacrifice generality and rigor to maintain sensible computational cost. These include unit testing, which exercises an isolated function of the code, and integration testing, which tests the correctness of the code output directly. Regression testing is a related activity that only tests for changes in the code behavior, and therefore is most useful to ensure that a code which has been rigorously verified does not suffer degradation of its credibility. Benchmarking, or code-to-code comparison, is another activity that is useful, but not generally as rigorous as code verification, as it relies heavily on the credibility of another code.

The EMPIRE code base [2] consists of electrostatic and electromagnetic field solvers, particle-in-cell (PIC) and fluid plasma representations, and collision chemistry capabilities. EMPIRE is also capable of relativistic particle transport, hybrid operation incorporating particle and fluid in the same simulation, as well as the ability to run on CPU, GPU, and hybrid computing platforms. The warm electron diode problem, which we refer to as the Vlasov-Poisson-Child-Langmuir (VPCL) diode problem, exercises the coupling between the electrostatic field solver and PIC particle transport, as well as algorithms for particle emissions from surface boundaries and output diagnostics. This problem is notable because it

employs three different diagnostics which (ideally) should provide similar results. Different results obtained for different versions of the code, different time integrator options, and different diagnostics showed both changes in code behavior over time, as well as differences between algorithms and diagnostics, many of which significantly deviated from the analytical solution and expected orders-of-convergence.

Stochastic noise in the code responses (output quantities of interest) is an inherent feature of the PIC plasma model which requires special consideration in verification efforts. An additional complexity is due to the fact that code responses generally converge due to three different discretization variables: namely mesh size, time step and particle weight, where particle weight is the number of physical particles represented by each computational particle. This is in contrast to typical deterministic finite element methods, which converge according to mesh size and time step only, and in some cases, mesh size only.

StREEQ, or Stochastic Richardson Extrapolation based Error Quantification [4], is a Sandia-developed tool for performing numerical error estimation, which was specifically designed to deal with the stochastic code responses and multiple discretization variables, which are inherent features in plasma simulations. This tool was used heavily in the code verification exercises in this report to estimate uncertainty intervals for the converged solution and orders-of-convergence, as well as to assess the credibility of equation modeling the convergence behavior itself.

In this report, we first discuss the VPCL diode verification problem, its analytical solution, and how it is simulated in EMPIRE. We then give a brief overview of the StREEQ numerical error estimation tool and how it is used in this code verification exercise. In the main part of the document, we discuss the code verification work for three different sets of simulations, including: an early version of EMPIRE, and a recent version of EMPIRE using the leapfrog time integrator, and the same recent version of EMPIRE using the velocity Verlet time integrator. Finally, we conclude by summarizing the results and providing an assessment of EMPIRE's current capability to accurately solve the VPCL diode problem.

2. WARM ELECTRON DIODE VERIFICATION PROBLEM

The warm electron diode is one of the few non-trivial plasma problems which has an analytical solution. This solution is discussed in Section 2.1, but the details of its derivation are beyond the scope of this report and described elsewhere [5]. We also discuss the computational solution performed using EMPIRE in Section 2.2.

This problem is a good test of a number of plasma physics components and algorithm implementations. The space-charge-limiting current flow behavior relies on a coupling between field solve and particle transport steps, and achieving a high order of accuracy requires proper implementation of the time integration algorithm. This problem also tests the particle emission boundary condition and potentially several output diagnostics including globally-averaged, surface-based, and line-integrated quantities; each of these are discussed more fully in Section 2.2.

2.1. ANALYTICAL SOLUTION

The warm electron diode problem, which we will call the Vlasov-Poisson-Child-Langmuir (VPCL) diode problem, was derived in detail in Reference [5]. This problem consists of a steady-state, one-dimensional, non-magnetized electron gas contained between grounded electrodes separated by gap length L. At the cathode, thermal electrons are emitted at temperature T_0 and number density n_0 yielding an emitted current flux of $J_0 = -en_0v_0/(2\sqrt{\pi})$, where $v_0 = \sqrt{2eT_0/m}$ is the thermal velocity, m is the electron mass, and e is the elementary charge. Of this injected current, only a fraction J^* of the emitted current, shown below, makes it across the gap.

$$J^* = -\frac{en_0v_0}{2\sqrt{\pi}}e^{-\Phi_m} \tag{2.1}$$

In Equation 2.1, the normalized potential extremum Φ_m is determined by inverting the integral shown below, which is a function a single parameter Λ. This parameter is the diode gap normalized by the electron Debye length λ_D at reference conditions, $\Lambda = L/\lambda_D$, where $\lambda_D = \sqrt{\epsilon_0 T_0/(en_0)}$ and ϵ_0 is the permittivity of free space.

$$\Lambda = \int_0^{\Phi_m} d\Phi \Bigg\{ \left[\left(1 + e^{-\Phi} \operatorname{erf} \sqrt{\Phi_m - \Phi} \right) - \left(1 + 2\sqrt{\frac{\Phi_m - \Phi}{\pi}} \right) e^{-\Phi_m} \right]^{-1/2} + \left[\left(1 - e^{-\Phi} \operatorname{erf} \sqrt{\Phi_m - \Phi} \right) - \left(1 - 2\sqrt{\frac{\Phi_m - \Phi}{\pi}} \right) e^{-\Phi_m} \right]^{-1/2} \Bigg\} \quad (2.2)$$

This integral is easily performed to high precision using numerical quadrature and inverted to obtain Φ_m, which is used (Equation 2.1) to obtain the steady current. The code verification problem analyzed in this report was obtained for the specific case: $T_0 = 10$ eV, $n_0 = 10^{16}$ m^{-3}, and $\Lambda = 20$; for which the current was obtained as $J^* = -77.0596$ A/m^2.

2.2. COMPUTATIONAL SIMULATION

Particle-in-cell (PIC) plasma codes in general are expected to converge as mesh size Δx, time step Δt, and macroparticle weight (the number of physical particles represented by each computational particle) W approach zero. Generally, PIC codes are assumed to converge to second-order in mesh and time, and first-order in weight. In our analysis, we normalized these parameters to obtain dimensionless discretization parameters: $\Delta x/\lambda_D$, $Co = v_0 \Delta t/\Delta x$, and $1/N_c = W/(\Delta x^2 n_0)$. Here, Co is the electron Courant number, and N_c is the number of computational particles per cell at emission conditions. The corresponding number of mesh elements in x, the direction normal to the gap, is $N_x = L/\Delta x$, which clearly must be integer valued. Note that the Courant number was specified in terms of the electron thermal velocity, which is appropriate for electrostatic simulations. For electromagnetic simulations, the appropriate velocity scale is the vacuum speed of light c_0, resulting in $Co = c_0 \Delta t/\Delta x$.

These normalized discretization parameters, $(\Delta x/\lambda_D, Co, 1/N_c)$, can be directly related to algorithmic stability and accuracy considerations. For explicit electrostatic PIC, $\Delta x/\lambda_D \leq 1$ is required for stability; although violations of this criterion generally manifests in self-heating which increases λ_D to the point where stability is maintained at the cost of solution inaccuracy. Likewise, $Co \leq 1$ limits particles traversing over multiple cells which leads to gross errors in particle trajectories. These Courant limits are also related to another time step stability criterion required to prevent instability, which requires $\omega_p \Delta t \leq 1$, where $\omega_p = \sqrt{e^2 n_0/\epsilon_0 m}$ is the electron plasma frequency. It is easy to show that this is automatically satisfied whenever $\Delta x/\lambda_D \leq 1$ and $Co \leq 1$ and is therefore redundant to our formulation of the discretization parameters[1].

Many particles per cell, i.e. $1/N_c \gg 1$, are required for adequate resolution of the charge density and computation of the electrostatic potential ϕ; therefore inadequate resolution of $1/N_c$ results in both stochastic noise and bias error. Since ϕ is used to define the electric field via $\boldsymbol{E} = -\nabla\phi$, and $\boldsymbol{E}$

[1]For electromagnetic simulations using implicit time integration, it is common to simulate with Co higher than unity. In this case, the plasma frequency criterion is no longer redundant, and care must be taken not to exceed $\omega_p \Delta t \leq 1$ in order to prevent spurious oscillations.

accelerates the electrons, this also adds noise and bias to the particle trajectories. For electromagnetic simulations, noisy cell charge and current densities induce noise in the electric field $\boldsymbol{E}$ and magnetic flux density $\boldsymbol{B}$ fields directly via Maxwell's Equations without using ϕ as an intermediate variable.

PIC plasma simulations are stochastic in general, and in this problem, the primary source of noise is the thermal emission boundary condition in which Monte Carlo particle samples are injected through the cathode during each time step. For this reason, we also simulate this problem several times for each discretization level, where statistical independence is obtained by varying the random number seed between simulations. We also obtain multiple independent statistical samples by time-averaging the responses over several large time windows[2] following the simulation achieving steady-state conditions.

Steady-steady state conditions for this problem are expected to exist after many batches of emitted electrons have exited the simulation by passing through either electrode, which requires $t \gg L/v_0$. In practice, we required $t \geq 1000L/v_0$ which was derived from experience [5] with this test simulated in the Aleph [1] plasma code, although this criterion was relaxed in the automated tests implemented into the EMPIRE repository in order to reduce computational cost.

While this is a one-dimensional problem, there is no one-dimensional simulation capability in EMPIRE. In practice, we used a $N_x \times 2$ mesh in the (x, y) dimensions, where aspect ratio of the mesh was fixed at unity. One-dimensional output quantities were then obtained by spatially-averaging over the y-direction. For electrostatic simulations run to steady state, there are three equivalent ways to measure the current, which we will identify as the global J_G, right-hand surface J_R and left-hand surface J_L current diagnostics.The global current diagnostic is calculated by averaging the x-momentum over the entire domain

$$J_G = -\frac{e}{mL} \int_0^L dL \int_{\mathcal{R}^3} d^3\boldsymbol{v}\; m v_x\, f(x, \boldsymbol{v}, t), \tag{2.3}$$

where $\boldsymbol{v}$ is the vector electron velocity with components (v_x, v_y, v_z), and $f(x, v, t)$ is the velocity distribution function. In the EMPIRE code, this is obtained by computing the average particle velocity over the entire spatial domain at each time step.

The surface diagnostics are defined relative to the absorbed currents $J_{A,R}$ and $J_{A,L}$ in the outward normal direction (i.e. $J_{A,R}$ and $J_{A,L}$ are defined as positive in the $\pm x$-direction respectively). These absorbed currents are computed by counting the the total charge absorbed by the surface during a time step and normalizing by the time step. The total current is obtained in each case by adding the emitted current (0 and $-J_0$ for J_R and J_L respecitvely) and transforming to the $+x$-direction yielding

$$J_R = J_{A,R} \tag{2.4}$$
$$J_L = J_0 - J_{A,L}. \tag{2.5}$$

[2] Large time windows are required for steady-state time-averaging of response quantities, since there may be considerable correlation between successive samples (in time) otherwise, which will lead to violations of our variance model (see Equation 3.2).

For electromagnetic simulations, the current can also be obtained by using a line integral of $\boldsymbol{B}$. This problem would require a three-dimensional simulation with a loop in the (y, z)-plane. The steady-state integral version of Ampere's Law then reduces to

$$J_B = \frac{1}{\mu_0 A} \oint_{\mathcal{C}} d\boldsymbol{\ell} \cdot \boldsymbol{B} \tag{2.6}$$

where μ_0 is the permeability of free space. Here, $\mathcal{C}$ is any closed loop along the edges of a cross-section (y, z)-plane at a fixed value $0 < x < L$ which has enclosed area A and a surface normal (using the right-hand rule) in the $+x$-direction. At the time of this report, EMPIRE does not have the capability to simulate the VPCL diode problem electromagnetically, primarily due to the lack of enforcing that both electrodes have a common ground, which requires enforcing

$$\int_0^L dx \; E_x = 0, \tag{2.7}$$

where E_x is the x-component of the electric field. When this capability becomes available, an electromagnetic version of this heavy verification problem should be performed, which would allow verification of four different current diagnostics with the same set of simulations.

3. STREEQ NUMERICAL ERROR ESTIMATION

The StREEQ, or Stochastic Richardson Extrapolation based Error Quantification, numerical error estimation tool was developed primarily for verification of plasma and other codes which have inherently stochastic responses. This tool addresses a fundamental need in the computational plasma community, since previous studies [6] make use of standard tools developed for deterministic simulations, which may not be appropriate to the application.

While documentation exists for earlier versions [5], the best reference is the current version of the user's manual [4]. The basic formulation of the StREEQ method includes multiple fits to a model for the numerical discretization error. While this form is easily generalized to any number of discretization variables, the formulation used in this work is shown below in terms of the variables considered in the electron diode problem $(\Delta x/\lambda_D, Co, 1/N_c)$.

$$J = \beta_0 + \beta_x \left(\frac{\Delta x}{\lambda_D}\right)^{\gamma_x} + \beta_t Co^{\gamma_t} + \beta_w \left(\frac{1}{N_c}\right)^{\gamma_w} + \beta_{xt} \left(\frac{\Delta x}{\lambda_D}\right)^{\gamma_x} Co^{\gamma_t} + \beta_{tw} Co^{\gamma_t} \left(\frac{1}{N_c}\right)^{\gamma_w} + \beta_{wx} \left(\frac{1}{N_c}\right)^{\gamma_w} \left(\frac{\Delta x}{\lambda_D}\right)^{\gamma_x} + \epsilon \tag{3.1}$$

Here, the β coefficients and γ orders-of-convergence are variable parameters determined by global optimization, and ϵ is the residual.

A single least-squares fit is obtained based on the original data set and used to perform a lack-of-fit F-test, which is used as a credibility assessment. This step obtains a probability value (p-value), which when greater than a critical value, provides evidence that Equation 3.1 is appropriate for the data. For the results presented in this report, the commonly-used 5% critical value was used. When the F-test fails, especially when the p-value is very small, the StREEQ analysis should be viewed with increased skepticism.

A large number of additional fits are performed with various residual error norms and weight[1] strategies, and for multiple bootstrap samples of the underlying response data. The residual error norms consist of L_1, L_2, and L_∞, which have different assumptions on the treatment of outliers. Three different weighting schemes emphasize the less- and more-refined response data, in addition to neutrally-biased (variance-weighted). For these options, there are a total of nine fitting models, which are fit for a large number of bootstrap samples of the local mean responses, providing a large pool of fits.

[1] The weights referred to here are unrelated to the particle weights used in EMPIRE and other PIC plasma simulations.

Variance estimates of the data are made by fitting to a model. For the electron diode problem, the variance is expected to be inversely proportional to the total number of computational particles in the simulation, leading to

$$\mathrm{var}(J) \propto \frac{\Delta x}{\lambda_D} \frac{1}{N_c}. \tag{3.2}$$

This assumed variance relationship is tested on an auxiliary variable using of the Brown-Foresythe non-parametric equality-of-variances test. When this test passes with a p-value of 5% or greater, the variance is represented by fitting to Equation 3.2; otherwise, the unbiased sample variance is used. This is ordinarily used to improve the variance estimate for small sample sizes for which the sample variance estimator can be noisy, which is not an issue for this analysis due to the large number of simulated responses. However, deviations from the variance model (indicated by low p-values) indicates unexpected performance, and indicates potential problems with the code or data post-processing techniques.

This pool of fitted values of β and γ values are used to perform the numerical error analysis. Most importantly, the distribution of β_0 is used to estimate the fully-converged solution with uncertainty. When an analytical solution is available, this distribution is compared to it to determine if the simulations are converging to the correct solution. Likewise, the distributions of the γ values are used to determine if the orders-of-convergence are consistent with theoretical predictions.

Finally, StREEQ has a method of performing the above analysis on a sequence of subdomains of the full data set. These subdomains are chosen by eliminating discretization levels that produce outliers in ϵ, which is interpreted as evidence of being outside the asymptotic region. This process is continued until there is no longer a sufficient number of discretization levels to perform the necessary fits. From this list of domains, the optimal one is chosen as the one having the lowest uncertainty, while still passing the F-test. For cases where no subdomain passed the F-test, the optimal domain is chosen as the one with the largest F-test p-value. Since the optimal domain is chosen based on the uncertainty level (when the F-test passes), there may be a different optimal domain for β_0 and each γ.

StREEQ has several advantages over traditional verification methods. In particular, it was formulated specifically for simulations with stochastic noise, which is characteristic of plasma codes like EMPIRE. It has also been shown to be more robust at capturing the exact solution than methods based on least-squares fitting alone [4]. The credibility assessment provides some statistical evidence of whether to trust the analysis, and it is also capable of finding which discretization levels should be excluded in the analysis, providing the analyst some guidance to where additional simulations should be performed.

4. PRELIMINARY CODE VERIFICATION

This chapter discusses the first code verification effort for the VPCL diode problem and includes details of the EMPIRE simulations as well as the numerical error and convergence estimation using the StREEQ code.

4.1. EMPIRE SIMULATIONS

The preliminary analysis was performed with an early code version compiled in April 2018, soon after the required code features were implemented in EMPIRE. Initially, the steady current response was extracted from the mesh output and determined by spatially averaging the electron momentum. The simulations were run to a time of $2000L/v_0$ where the final half of each simulation was time-averaged into 100 bins.

The mesh discretization parameter was varied as $\Delta x/\lambda_D = 1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}$, and the time step discretization was varied as $Co = 1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}$. In this early version of the electron emission boundary condition, direct control of the particle weight was not possible; instead, the analyst had control of N_e, which is the (integer) number of particles emitted per element face and per time step. This parameter was set as $N_e = \frac{1}{4}N_c'Co$ where $N_c' = \frac{2}{\sqrt{\pi}}N_c \approx N_c$ was varied as $N_c' = 24, 48, 96, 192, 384, 768$.

We expected the variance in the responses to show inverse proportionality to the number of computational particles in the simulation, where the number of samples in the simulation scales as $\sim N_c'\lambda_D/\Delta x$. To achieve a constant variance in the mean of the pool of responses at each unique discretization level[1] requires multiple simulations for each case. For this work, we simulated R independent simulations per discretization level, where R is defined below, resulting in a total of 3780 simulations.

$$R = 4608\,\frac{\Delta x}{\lambda_D}\,\frac{1}{N_c'} \tag{4.1}$$

The simulated responses are plotted in Figure 4-1. Here the error bars represent 95% confidence intervals of the mean response for each discretization level, where the statistical noise is too small to be noticeable in this plot. In contrast, the simulation results for only a single $R = 1$ simulation at each discretization level are shown in Figure 4-2, where the statistical noise is more apparent. The clusters of error bars represent unique mesh and time step discretizations, but variable weight discretization, where the weight discretization becomes more refined (larger N_c') from right to left.

[1] Although StREEQ has no requirement for constant variance in the mean of the underlying data set, this is generally considered good practice when available computational resources allow.

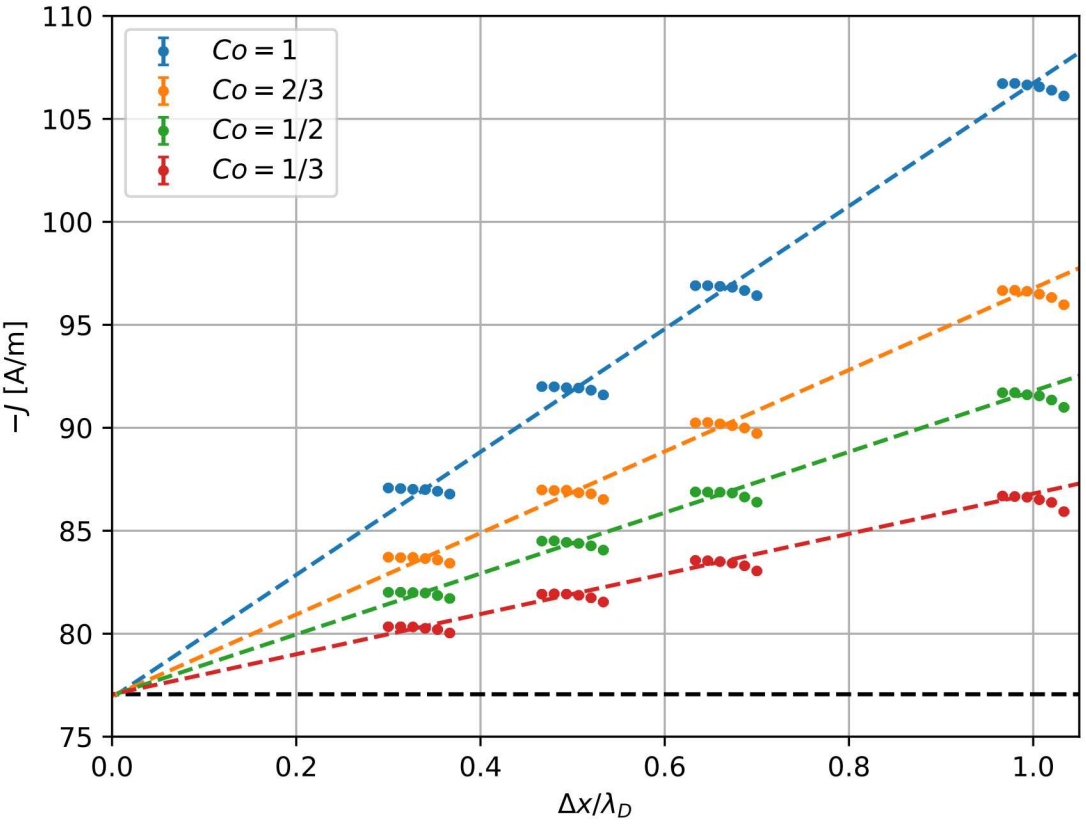


Figure 4-1. Preliminary VPCL diode simulation results based on all 3780 simulations. The error bars represent 95% confidence intervals in the mean response, for which the noise level is too small to be discerned. Each cluster of error bars is the data for a single mesh and time step discretization, but variable weight discretization.The exact solution is shown as a black dashed line, and the remaining dashed lines are discussed in Section 4.2.

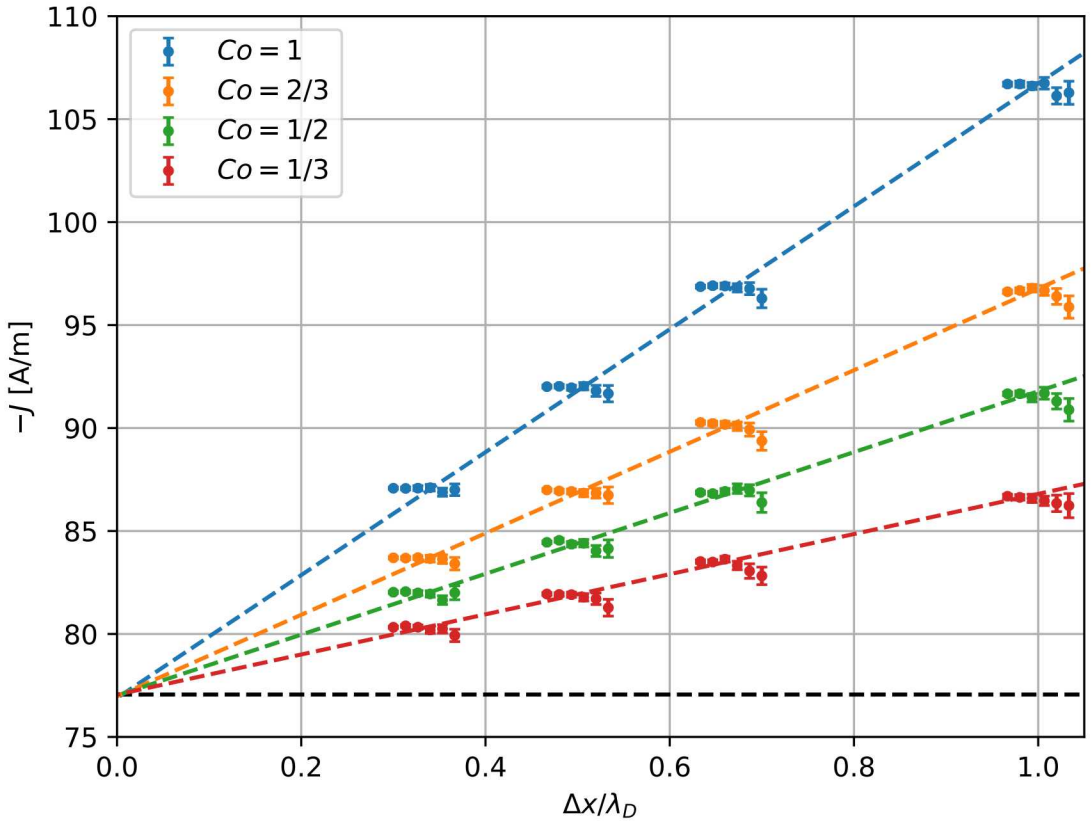


Figure 4-2. Preliminary VPCL diode simulation results based on a single simulation per discretization level. The error bars represent 95% confidence intervals in the mean response. Each cluster of error bars is the data for a single mesh and time step discretization, but variable weight discretization. The exact solution is shown as a black dashed line, and the remaining dashed lines are discussed in Section 4.2.

4.2. STREEQ ANALYSIS

Numerical error analysis was performed on response data (Figure 4-1) using the StREEQ tool. By analyzing multiple subdomains of the response data, we found the most favorable results for β_0, γ_x, and γ_t by fitting a subset of 49 of the original 96 discretization levels; while the optimal results for γ_w were obtained by fitting 51 discretization levels. For both of these subsets, credibility test (F-test) passed, providing evidence that the discretization error model (Equation 3.1) is appropriate. The variance model test also passed, indicating that the variance in the responses are consistent with Equation 3.2.

The predicted fully-converged solution and convergence rates are shown in Figure 4-3. The results (here, and throughout the document) are shown as both a kernel density estimation (KDE) smoothed distribution of fit values, and 95% estimated uncertainty intervals, where the predicted fully-converged solution estimate (obtained from the distribution of fitted β_0 values) bounds the exact solution to reasonable precision. Likewise, the mesh, time step, and weight orders-of-convergence are consistent with first-order convergence, although the degree of uncertainty for weight convergence is large. Note that the hard edges at $\gamma_w = 0.25$ and 2 are due to optimization constraints in StREEQ used for robustness and computational speed, which can be relaxed, if desired; however, this would not change the overall interpretation of the results. Likely, this large uncertainty is due to the smaller sensitivity of the response due to changes in N_c', as is easily discerned from Figures 4-1 and 4-2.

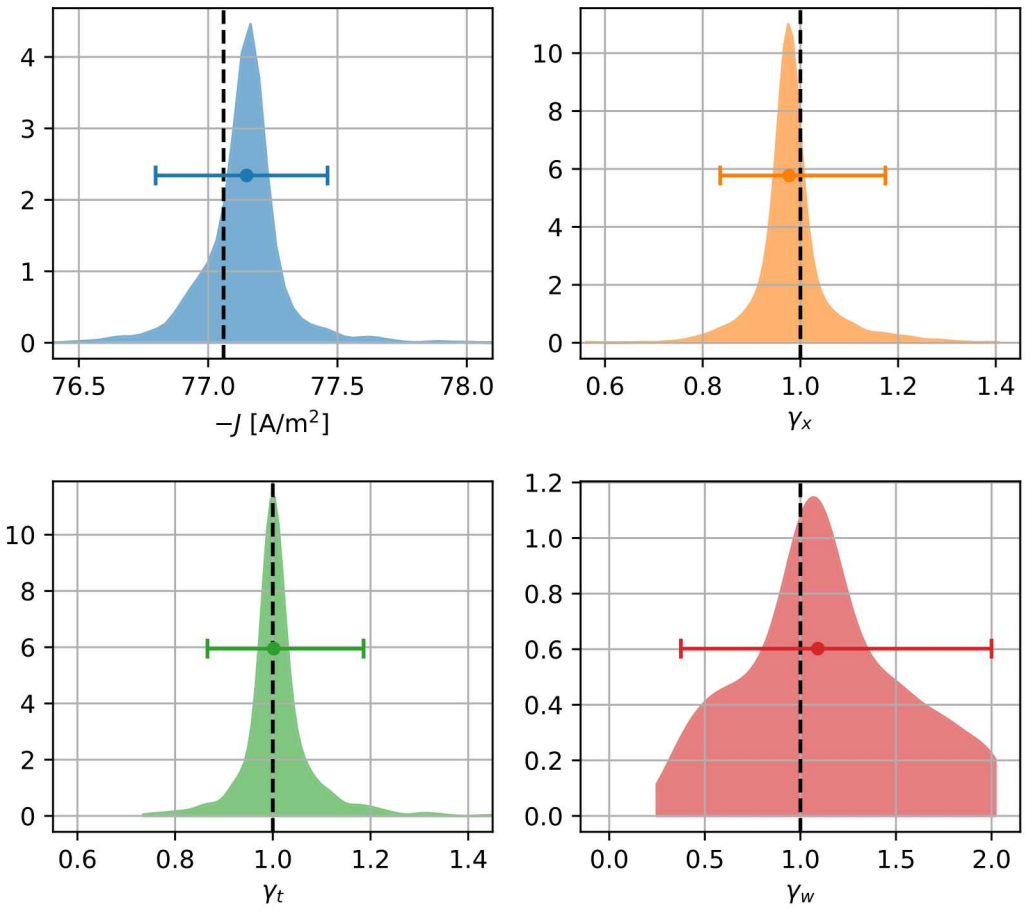


Figure 4-3. Preliminary StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence. Black dashed lines indicate exact solution and first-order convergence.

The dominant coefficient in Equation 3.1 (other than β_0) was found to be β_{xt}, which indicates strong cross-coupling between mesh and time step convergence. Of secondary importance were the pure mesh and time step terms with leading coefficients β_x and β_t. The dominance of these terms is readily apparent in Figures 4-1 and 4-2, where the dashed colored lines were obtained by plotting

$$J = \beta_0 + \beta_x \left(\frac{\Delta x}{\lambda_D}\right)^{\gamma_x} + \beta_t Co^{\gamma_t} + \beta_{xt} \left(\frac{\Delta x}{\lambda_D}\right)^{\gamma_x} Co^{\gamma_t} \tag{4.2}$$

where representative fitted values were used for $(\beta_0, \beta_x, \beta_t, \beta_{xt})$ and all convergence rates were fixed at unity. This observed convergence behavior was in stark contrast to expectations, both because second-order mesh and time step convergence is expected from the algorithms implemented in the code, and because the cross-coupling β_{xt} term is generally expected to be small.

4.3. INITIAL REGRESSION AND CODE VERIFICATION TESTS

Due to the relatively high computational cost of the large number of simulations required by this analysis, an automated regression and a code verification test were developed for the EMPIRE-PIC and ExtendedTesting repositories, respectively. In both cases, the computational expense was greatly reduced in order to allow the tests to be run more regularly. For the regression test, each simulation was run to $t = 2L/v_0$, where the final half of the simulation was time-averaged into a single response evaluation. The code verification test was run to $t = 200L/v_0$, where the latter half was time-averaged into 10 responses per simulation. Likewise, the effect of weight convergence was ignored for the verification test by fixing $N_c' = 768$, for which the simulations are relatively converged in N_c' as shown in Figure 4-2. These tests were implemented in a similar manner to what is described in Section 4.1, except the the global output diagnostic $J = J_G$ described in Section 5.1 was used for the response outputs.

For the regression test, four simulations are performed at $\Delta x/\lambda_D = 1, \frac{1}{2}$ and $Co = 1, \frac{1}{2}$, and the ratio of bias errors $\Delta J = J - J^*$ are tested to ensure that the dominant behavior is consistent with $J = \beta_0 + \beta_{xt} \left(\Delta x/\lambda_D\right)^{\gamma_x} Co^{\gamma_t}$, and that it converges to first-order in mesh and time step. This is an approach that was generalized from the use of specifically chosen *spatial and temporal refinement factors* [3] used to test convergence behavior on a small number of simulations.[2] Specifically, we expected the following ratios to hold:

$$\frac{\Delta J(\Delta x/\lambda_D = 1, Co = 1)}{\Delta J(\Delta x/\lambda_D = \frac{1}{2}, Co = 1)} = 2 \tag{4.3}$$

$$\frac{\Delta J(\Delta x/\lambda_D = 1, Co = 1)}{\Delta J(\Delta x/\lambda_D = 1, Co = \frac{1}{2})} = 2 \tag{4.4}$$

$$\frac{\Delta J(\Delta x/\lambda_D = 1, Co = 1)}{\Delta J(\Delta x/\lambda_D = \frac{1}{2}, Co = \frac{1}{2})} = 4 \tag{4.5}$$

For the regression test implemented in the repository, we require all of these ratios to be within 10% of the expected values. The results obtained for a recent version of EMPIRE found these ratios to be 1.90, 2.09, and 3.89 respectively, providing evidence of the previously observed convergence behavior. We note that the imposed 10% tolerance for agreement was chosen for the sole reason that it is a little larger than the discrepancies observed for the test at the time it was implemented. For this reason, the test behaves as a regression test rather than a true code verification test.

For the automated code verification test, 16 simulations were performed at $\Delta x/\lambda_D = 1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}$ and $Co = 1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}$, and a StREEQ analysis was performed verifying that the predicted fully-converged

[2] In contrast to the typical use of spatial and temporal refinement factors, where the β_x and β_t terms are dominant and only two simulated responses are used, we adapted the approach to test the non-standard behavior to first-order in mesh and time step, which required four simulations.

solution and mesh and time step convergence rates captured the analytical value and first-order convergence respectively. Recent results for this test are shown in Figure 4-4, for which the StREEQ estimates bound the exact solution and mesh and time step convergence rates, albeit with increased uncertainty, particularly for the fully-converged solution. The pass criterion for this test required that the F-test passes for at least one subdomain, and that the the StREEQ estimates appropriately bound the exact solution, and first-order convergence for mesh, time step, and weight.

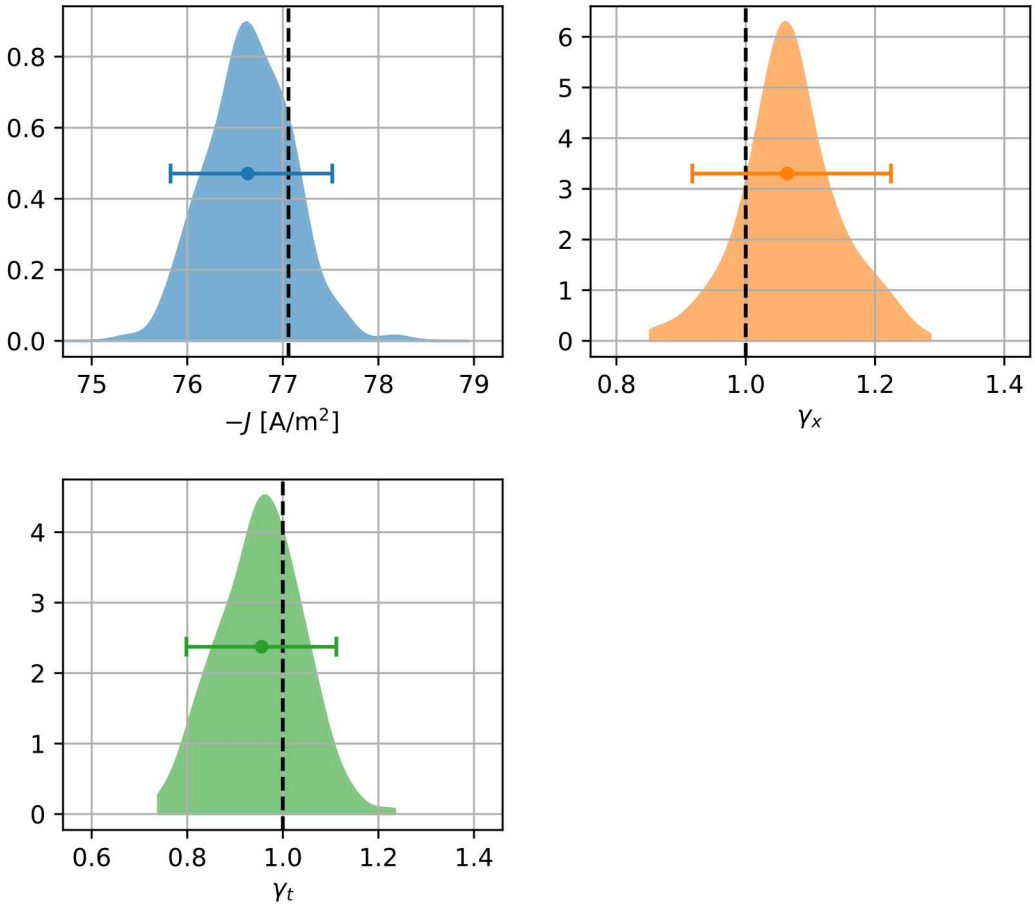


Figure 4-4. StREEQ estimates of the fully-converged solution (blue), and mesh (orange) and time step (green) orders-of-convergence for the implemented code verification test in the EMPIRE ExtendedTesting repository. Black dashed lines indicate exact solution and first-order convergence.

When the necessary code changes and bug fixes are implemented in order to achieve the expected second-order convergence, this test will perform differently, ideally showing results consistent with second-order convergence but inconsistent with first-order convergence. However, this is not guaranteed since the test size (i.e. number of simulations, and time to steady state, which sets the relative stochastic noise level) and the appropriate number of discretization variables will need to be refactored on the basis of a much heavier test; the analysis presented in Chapters 5 and 6 document the current state of this refactor for additional output features and implementation of a new time integrator respectively.

While these test implemented in the EMPIRE repository were useful, they lacked the precision to determine code changes that the heavier tests documented in the report were able to detect. Moreover, automating such a process with StREEQ analysis is difficult for a rapidly changing code, since exceptionally large computational resources are required to perform code verification to assess large variations in convergence behavior; however, for a more stable code base, the analysis can be better targeted.

5. CODE VERIFICATION USING MULTIPLE DIAGNOSTICS

In contrast to the preliminary analysis (Section 4), which was performed on responses extracted from the mesh output, the analysis performed in this chapter was performed after new output diagnostics were implemented (see Section 2.2). The global output diagnostic J_G is believed to be equivalent to the mesh-based output used in the preliminary analysis, but requires much less post-processing. The surface particle diagnostics allowed the steady current to be evaluated by measuring the electrons absorbed by the electrodes on each side of the diode gap (see Section 2.2).

Other than the extraction of the response output quantities of interest, the EMPIRE simulations were performed as described in Section 4.1 with one additional change. This change used a new feature in the particle emission to allow better control of the particle weights, permitting direct variation of N_c rather than N_c'. For this case we chose a smaller set of values for particle weight, $N_c = 32, 64, 128, 256$, where the number of simulations per discretization level R was initially chosen according to the formula below (c.f. Equation 4.1), which required 900 total simulations. The simulated responses in this chapter, as well as Chapter 6, were run with a version of EMPIRE compiled on June 6, 2012.

$$R = 1536 \frac{\Delta x}{\lambda_D} \frac{1}{N_c} \tag{5.1}$$

While this was adequate for analyzing the global diagnostic, the surface diagnostics exhibited substantially less discretization error and converged at higher order causing the numerical error to be hard to distinguish from the stochastic noise. Therefore, the numerical error analysis for the surface currents was performed on the basis of a $10\times$ larger set of data using R chosen below, which required 9000 total simulations.

$$R = 15360 \frac{\Delta x}{\lambda_D} \frac{1}{N_c} \tag{5.2}$$

During the long time period since the preliminary analysis, EMPIRE has undergone numerous changes and bug fixes in addition to the implementation of new output diagnostics. It was originally theorized that the observed first-order convergence in the global diagnostic was due to inaccurate representation of the electric field within the elements adjacent to the boundary. A fix for this issue was implemented, but second-order convergences were not observed in the global diagnostic. A more probable reason behind the low-order convergence of global diagnostics is due to the leapfrog time integrator algorithm, which maintains the particle velocities at a half time step advanced from the corresponding particle positions. A fix for this lack of time synchronization of the particle population was only partially

addressed with the initial implementation of the velocity Verlet time integrator. For the simulations in the present chapter, the leapfrog algorithm was used, while Chapter 6 documents the same analysis with velocity Verlet enabled.

5.1. GLOBAL OUTPUT DIAGNOSTIC

The raw simulation data for the global output diagnostic is shown in Figure 5-1. As before, the error bars represent 95% confidence intervals of the mean response for each discretization level, where the statistical noise is too small to be discerned. The clusters of error bars represent unique mesh and time step discretizations, but variable weight discretization, where the weight discretization becomes more refined (larger N_c) from right to left. The dominant terms in the discretization error, due to coefficients $(\beta_0, \beta_x, \beta_t, \beta_{xt})$ are plotted based on representative fitted values as dashed lines assuming first-order mesh and time step convergence; as before, β_{xt} was the dominant term. (See Equation 4.2)

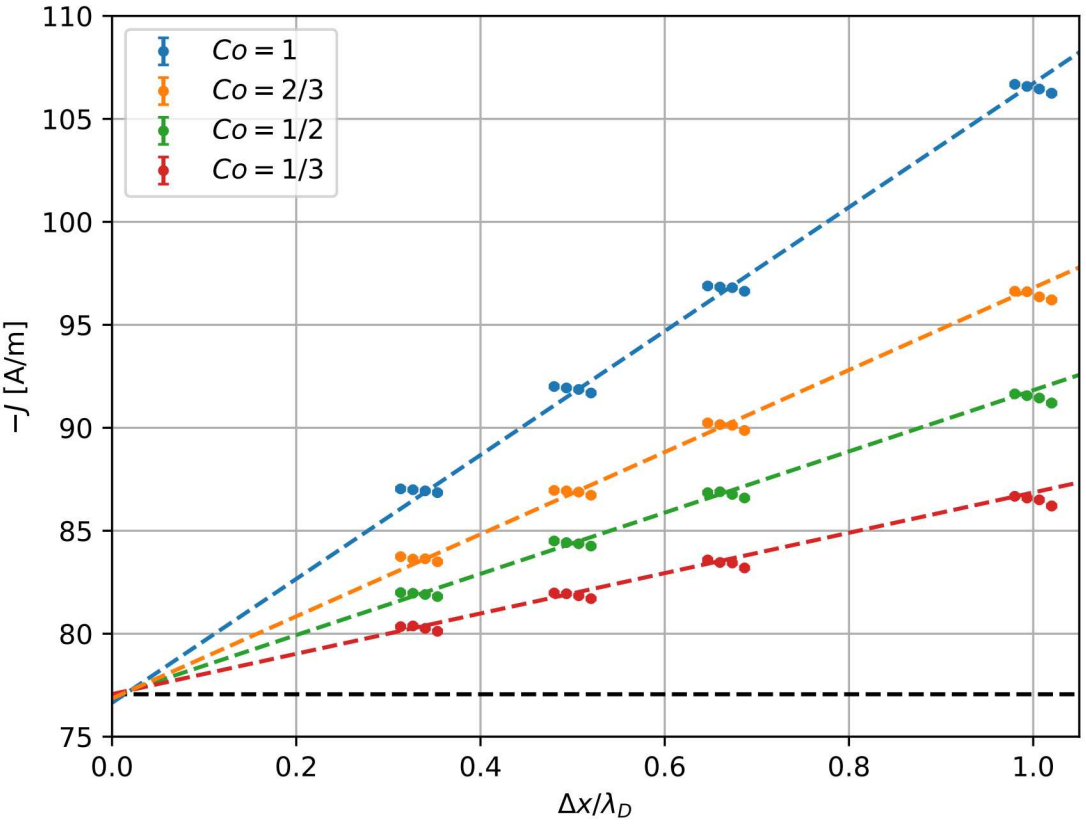


Figure 5-1. Simulation results using the global output diagnostic for 900 simulations with the leapfrog algorithm. The error bars represent 95% confidence intervals in the mean current response, for which the noise level is too small to be discerned. Each cluster of error bars is the data for a single mesh and time step discretization, but variable weight discretization. The exact solution is shown as a black dashed line, and the remaining dashed lines are the dominant discretization error model terms.

The StREEQ analysis readily passed the credibility test and the variance model test, and the optimal results for all fit parameters were obtained using the full set of simulation data with 64 discretization levels. The results, shown in Figure 5-2, exhibited degraded convergence behavior compared to the preliminary analysis (see Figure 4-3). In particular, Figure 5-2 shows that the StREEQ estimates do not bound the exact solution, albeit with tighter uncertainty intervals. The problem again appears to be converging at approximately first-order in mesh and time step, although the StREEQ estimates fail to bound $\gamma_x = 1$. As before, the weight order-of-convergence has a large uncertainty, but is consistent with first-order.

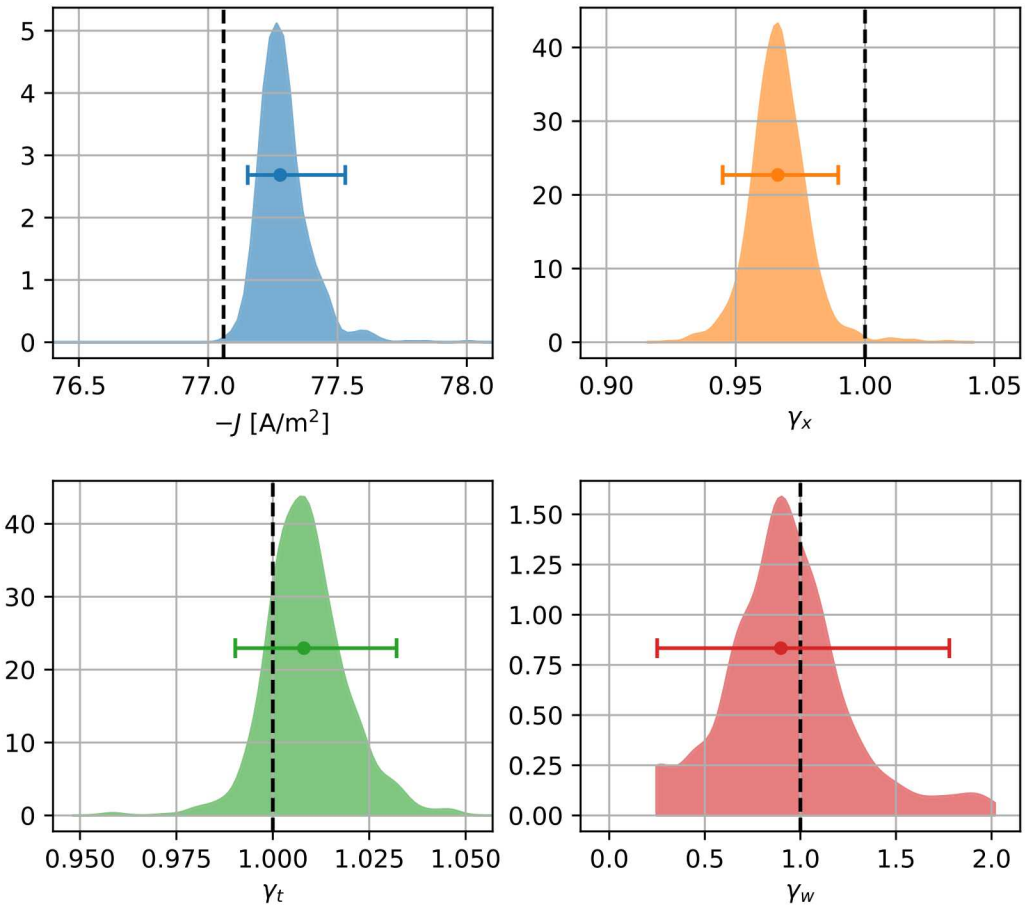


Figure 5-2. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence using the global output diagnostic for 900 simulations with the leapfrog algorithm. Black dashed lines indicate exact solution and first-order convergence.

5.2. RIGHT SURFACE DIAGNOSTIC

The simulated responses for the right surface current diagnostic based on 9000 simulations respectively are shown in Figure 5-3 below, which show considerably less numerical error compared to the global diagnostic. Here mesh and weight discretization show the clearest trend, while the time step discretization trend is less clear. In contrast to similar plots presented previously (c.f. Figure 5-2), the clusters of error bars with the same color represent unique mesh and weight discretizations, while within each cluster, the time step discretization becomes more refined (smaller *Co*) from right to left.

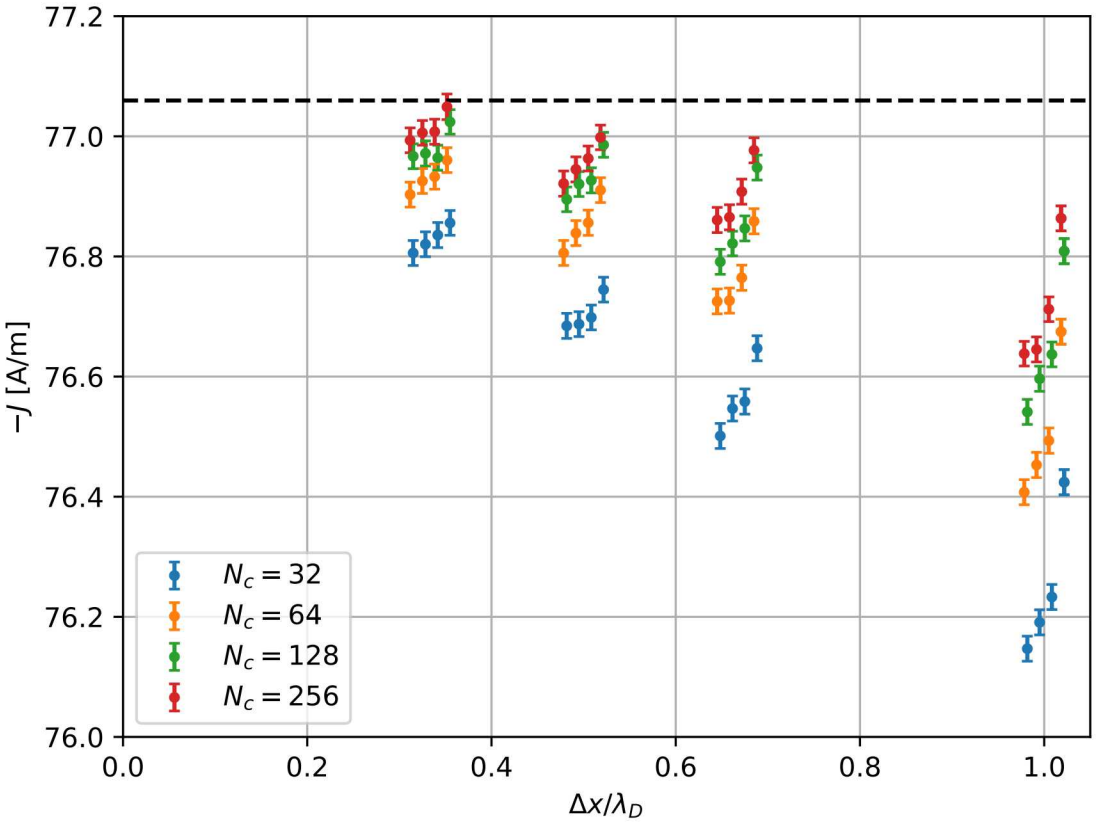


Figure 5-3. Simulation results for the right surface current diagnostic based on 9000 simulations with the leapfrog algorithm. The error bars represent 95% confidence intervals in the mean current response. Each cluster of error bars is the data for a single mesh and weight discretization, but variable time step discretization. The exact solution is shown as a black dashed line.

The StREEQ analysis found optimal results by fitting to all 64 discretization levels for all parameters except γ_t, for which a lower uncertainty result was obtained by using only 56 out of the original 64 discretization levels; in both cases, the credibility and variance model test passed. These results are shown in Figure 5-4 below. Unlike the results for the global diagnostic, these results for the right surface diagnostic show better bounding of the exact solution, and with smaller uncertainty. While the mesh convergence seems to be of lower-than-expected order, it is clearly higher than first-order. While the time step convergence has a large uncertainty, it is nearly centered about second-order. The weight convergence also bounds first-order, but with lower uncertainties, likely due in part to the fits being performed using about ten times the simulation data.

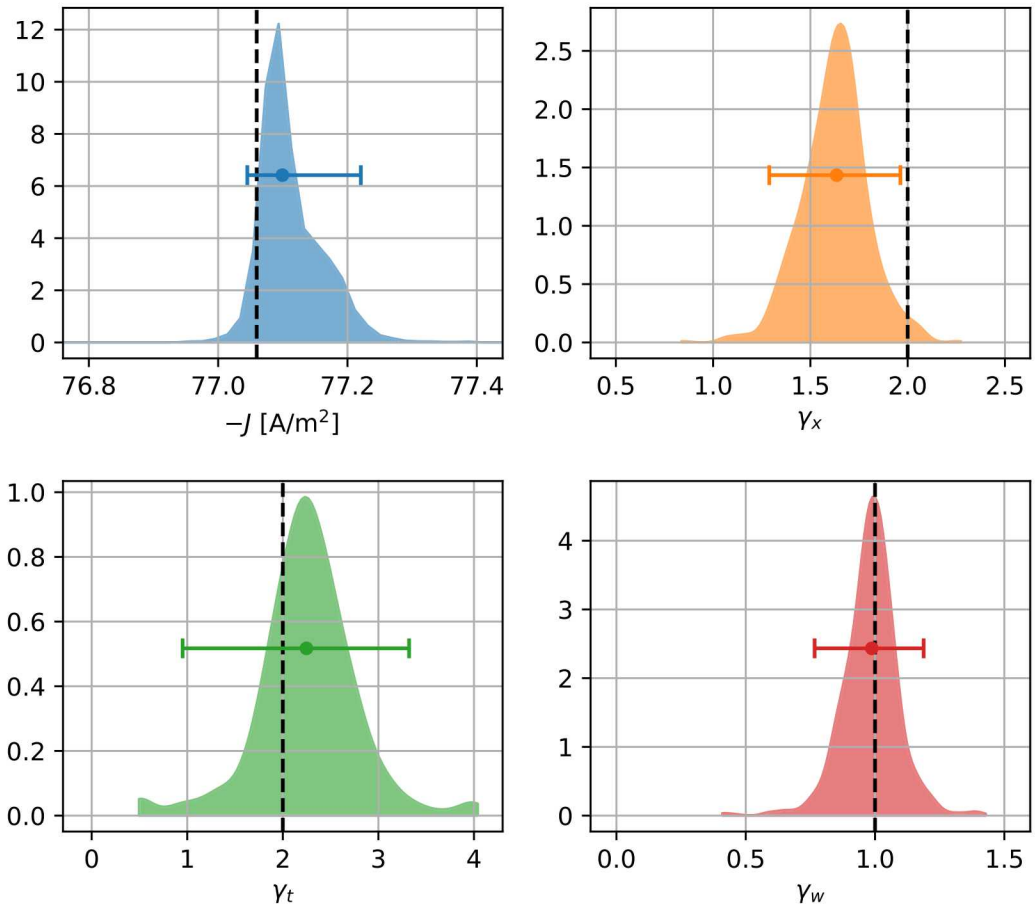


Figure 5-4. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence using the right surface current diagnostic for 9000 simulations with the leapfrog algorithm. Black dashed lines indicate exact solution, second-order mesh and time step convergence, and first-order weight convergence.

5.3. LEFT SURFACE DIAGNOSTIC

The simulation data for the left surface diagnostic are shown in Figure 5-5, which show similar levels of numerical error but significantly larger variances. A likely explanation for this increase variance is due to the fact that the two diagnostics were averaged from very different populations. Specifically, while the right diagnostic was averaged from the high-energy tail of the emitted electron population, the left diagnostic was formed from the remaining majority of the distribution.

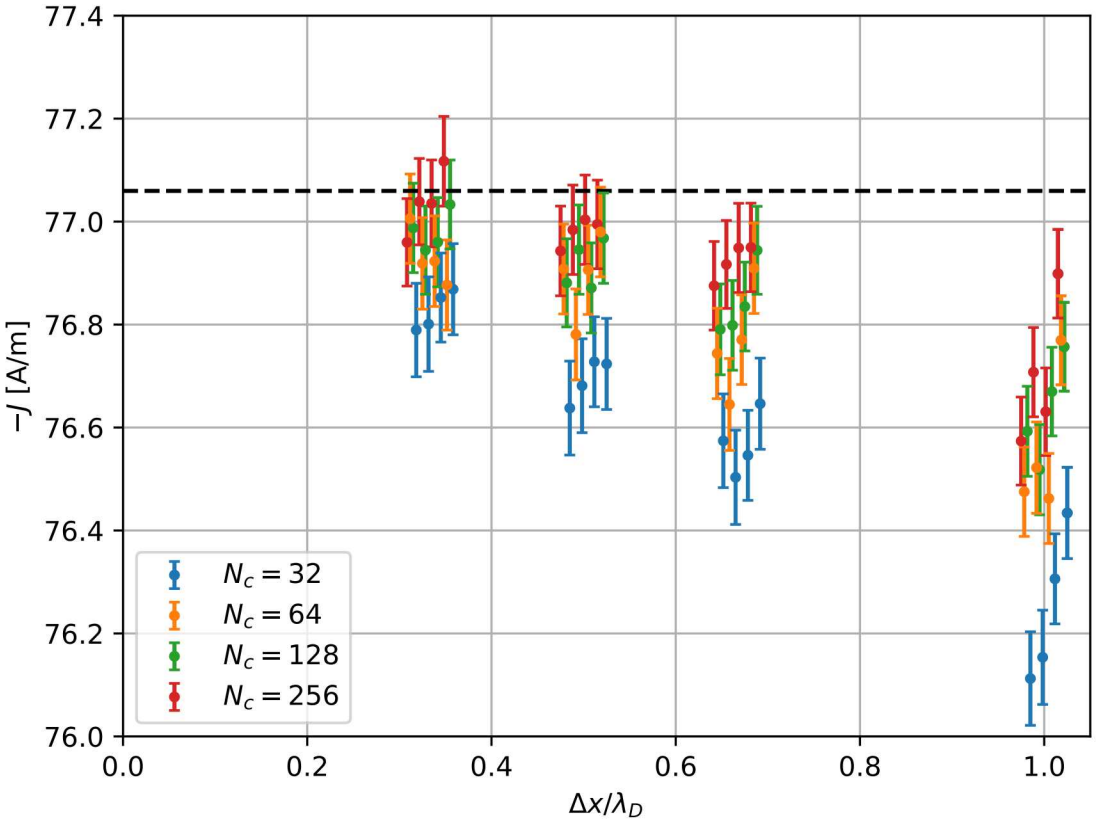


Figure 5-5. Simulation results for the left surface current diagnostic based on 9000 simulations with the leapfrog algorithm. The error bars represent 95% confidence intervals in the mean current response. Each cluster of error bars is the data for a single mesh and weight discretization, but variable time step discretization. The exact solution is shown as a black dashed line.

The StREEQ analysis for the left diagnostic found optimal results by fitting to all 64 discretization levels for all parameters except γ_x, for which a lower uncertainty result was obtained by using only 56 out of the original 64 discretization levels; in both cases, the credibility test passed, but the variance model failed with a vanishingly small p-value. The reason behind the failure of the variance model test is not known at this time, but given the very large number of data (9000 simulations, each yielding 100 independent responses), even a relatively small discrepancy will cause the test to fail. Note that this is not expected to have any derogatory effect on the analysis, since StREEQ will default to the sample variance estimator when the variance model test fails, which given size of the data set, will be very reliable.

The numerical error analysis results are shown in Figure 5-4 below. Compared to the right diagnostic (Figure 5-4), the uncertainty is much larger, although the exact solution, second-order mesh and time step, and first-order weight convergence are all captured within the estimated uncertainty intervals.

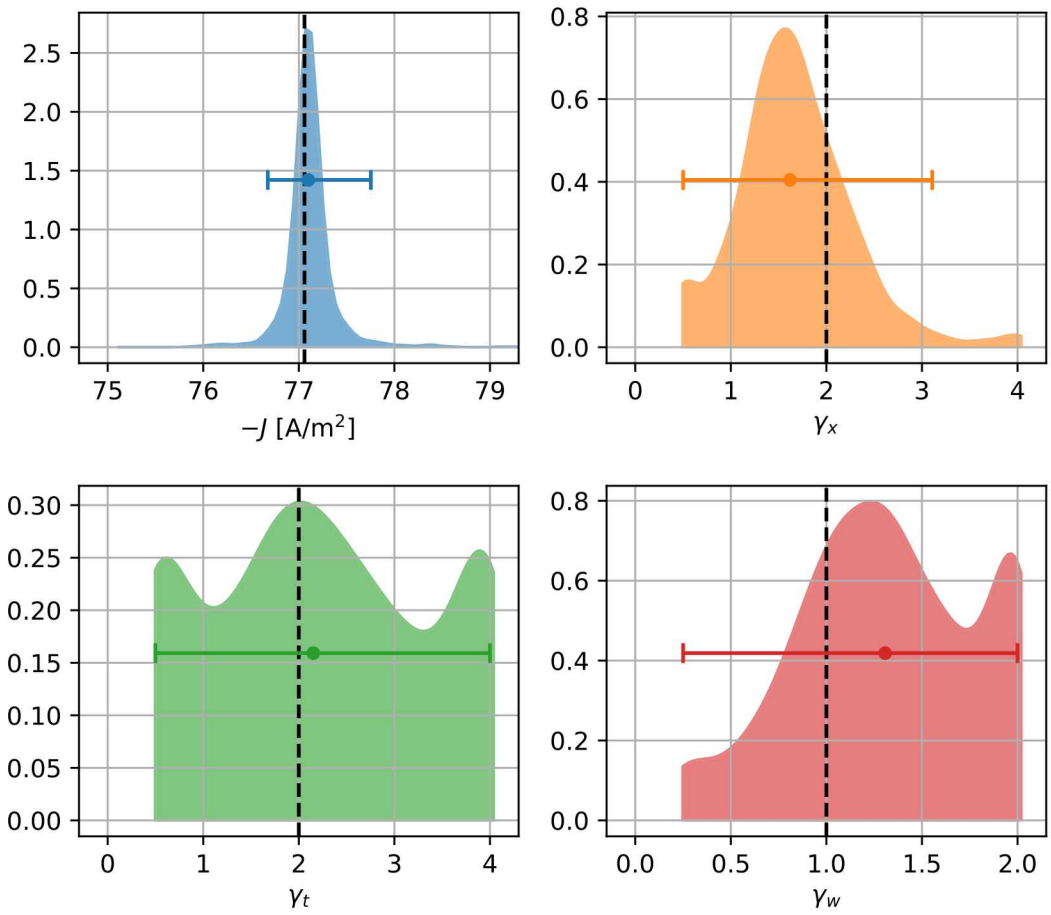


Figure 5-6. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence of the left surface current diagnostic for 9000 simulations with the leapfrog algorithm. Black dashed lines indicate exact solution, second-order mesh and time step convergence, and first-order weight convergence.

5.4. DISCUSSION

The key differences between the results presented in this chapter and Chapter 4 are due to differences in the output diagnostics, as well as a large number of other code changes in EMPIRE during more than a year of very active development. We also used a different range of discretizations in particle weight, although this choice is probably not very important since the largest differences for the global diagnostics were observed for estimates of the fully-converged solution and the mesh and time step convergence. Specifically, comparing Figures 4-3 and 5-2, we find the indication of bias errors in the current and mesh convergence that are not apparent for the preliminary analysis. Ongoing work to correct the time synchronization of the global diagnostics is the most logical next step in the work to correct this issue.

Numerical error estimation results for the more recently implemented surface current diagnostics were also compared directly to the global diagnostic, which seemed to demonstrate higher-order convergence for the surface diagnostics. Since the anticipated second-order mesh and time step convergence was not convincingly verified, additional work is indicated for these diagnostics as well. While there are good reasons to expect differences between the results for the different diagnostics (especially regarding the relative noise level), we do expect second-order convergence for all of them.

6. CODE VERIFICATION OF THE VERLET ALGORITHM

In comparison with the analysis discussed in Chapter 5, the results presented in this chapter differ only because the velocity Verlet time integration algorithm is enabled. The implementation of the Verlet algorithm is a first step towards correcting the lack of time synchronization between particle positions and velocities, which is believed to be the primary reason behind the observed first-order convergence in the global output diagnostic. For this reason, we anticipated seeing an improvement in the convergence behavior of the global diagnostic, but essentially no change in the surface diagnostics.

6.1. GLOBAL OUTPUT DIAGNOSTIC

The raw simulation data for the global diagnostic in Figure 6-1 shows a very similar behavior with the Verlet algorithm compared to the previous version (Figure 5-1), although the dominant discretization error model terms do not fit the entire set of data as well. This is further exemplified in the StREEQ analysis, for which the credibility test fails miserably; the results shown in Figure 6-2 are for a subset of 48 out of the original 64 discretization levels for which the maximum p-value for the F-test of $0.014\% \ll 5\%$ was obtained. This is strong evidence that the discretization error model used in this analysis does not characterize the convergence behavior of the response data, and the results should be viewed with suspicion. The variance model test did pass, however. Compared to the analysis with the leapfrog time integrator, the uncertainties are much larger, and the differences with respect to the analytical solution and first-order convergence are greatly increased.

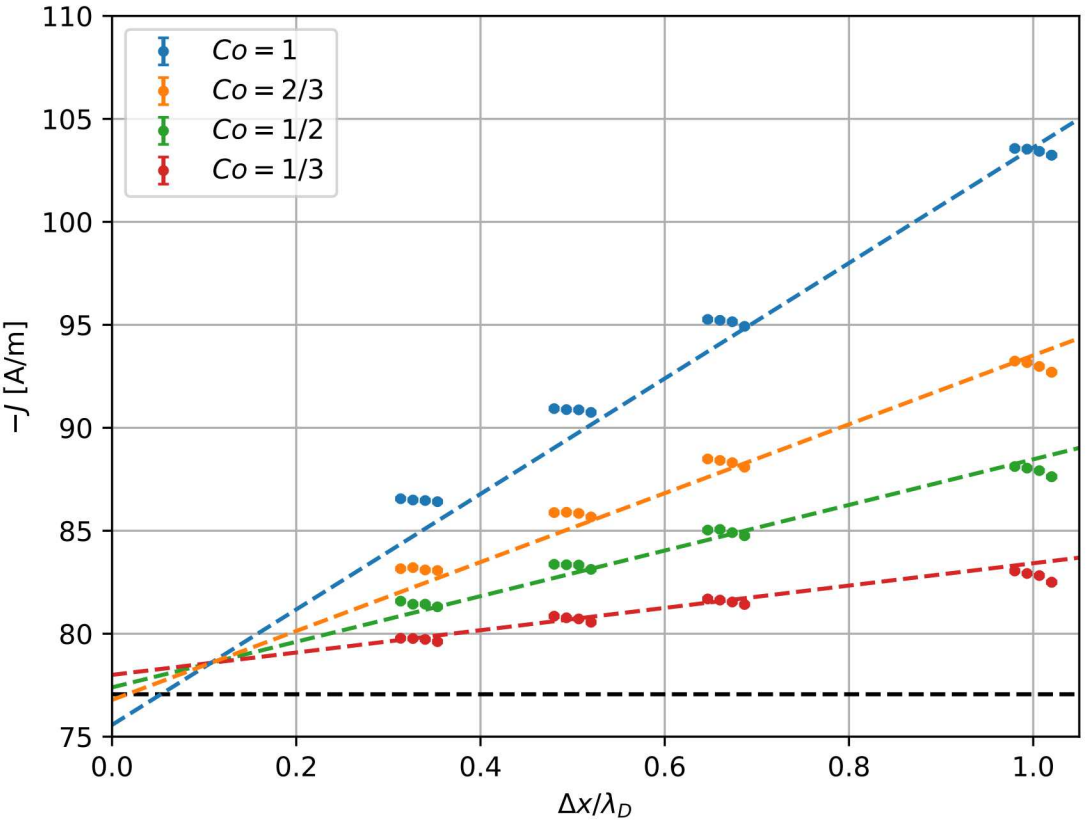


Figure 6-1. Simulation results using the global output diagnostic for 900 simulations with the Verlet algorithm. The error bars represent 95% confidence intervals in the mean current response, for which the noise level is too small to be discerned. Each cluster of error bars is the data for a single mesh and time step discretization, but variable weight discretization. The exact solution is shown as a black dashed line, and the remaining dashed lines are the dominant discretization error model terms.

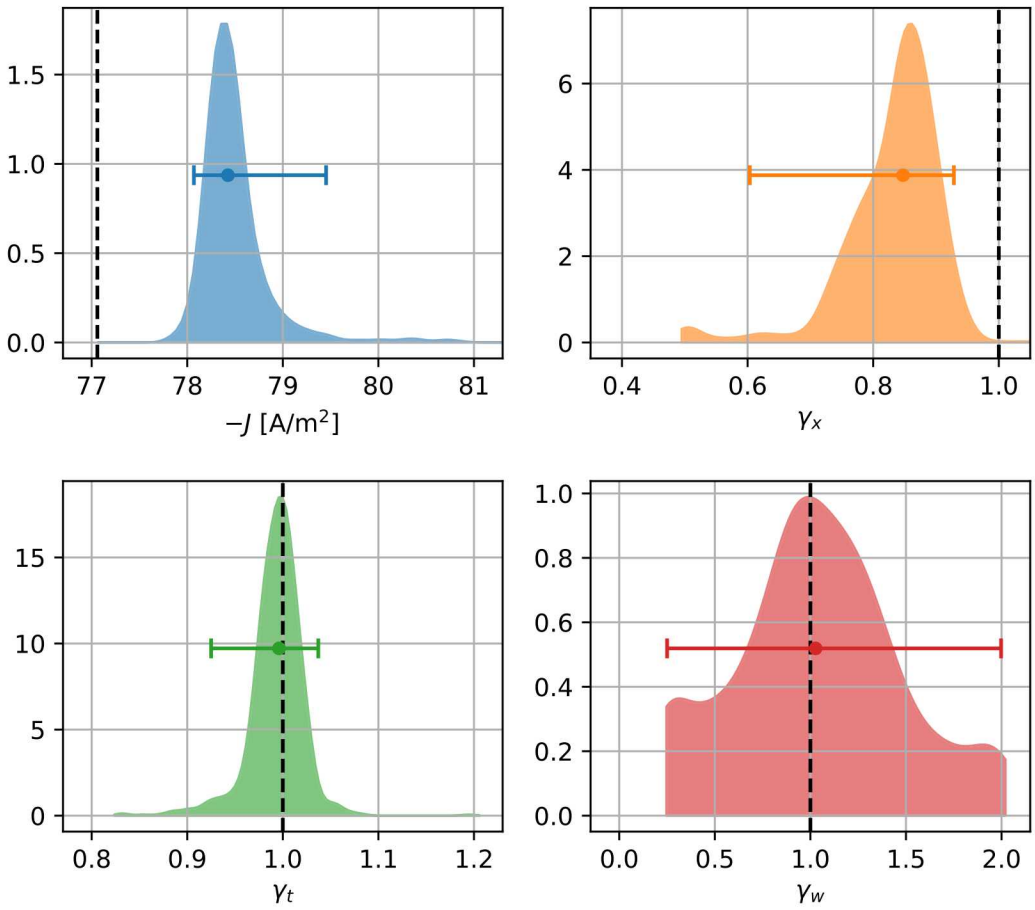


Figure 6-2. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence using the global output diagnostic for 900 simulations with the Verlet algorithm. Black dashed lines indicate exact solution and first-order convergence.

6.2. RIGHT SURFACE DIAGNOSTIC

Enabling the Verlet algorithm had a striking effect on the surface current diagnostics. For the right diagnostic, the raw data for the velocity Verlet is shown in Figure 6-3. Compared to the previous responses shown in Figure 5-3, the magnitude of the numerical bias error is greatly increased. Nonetheless, for the StREEQ analysis shown in Figure 6-4, the credibility and variance model tests passed for the optimal domain consisting of 45 out of the full set of 64 discretization levels. Compared to the results in the previous chapter, the convergence behavior is badly degraded. In particular, the uncertainty intervals no longer capture the exact solution, and the discrepancy with the anticipated second-order mesh convergence is now much larger.

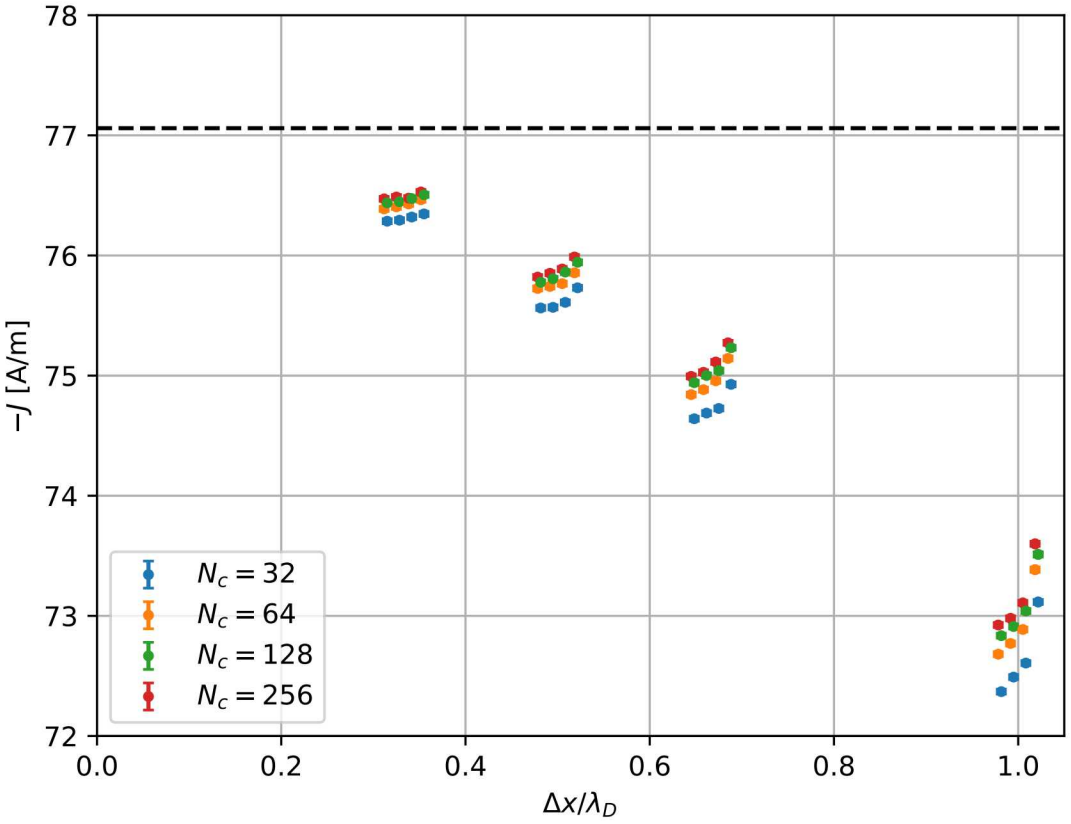


Figure 6-3. Simulation results for the right surface current diagnostic based on 9000 simulations with the Verlet algorithm. The error bars represent 95% confidence intervals in the mean current response. Each cluster of error bars is the data for a single mesh and weight discretization, but variable time step discretization. The exact solution is shown as a black dashed line.

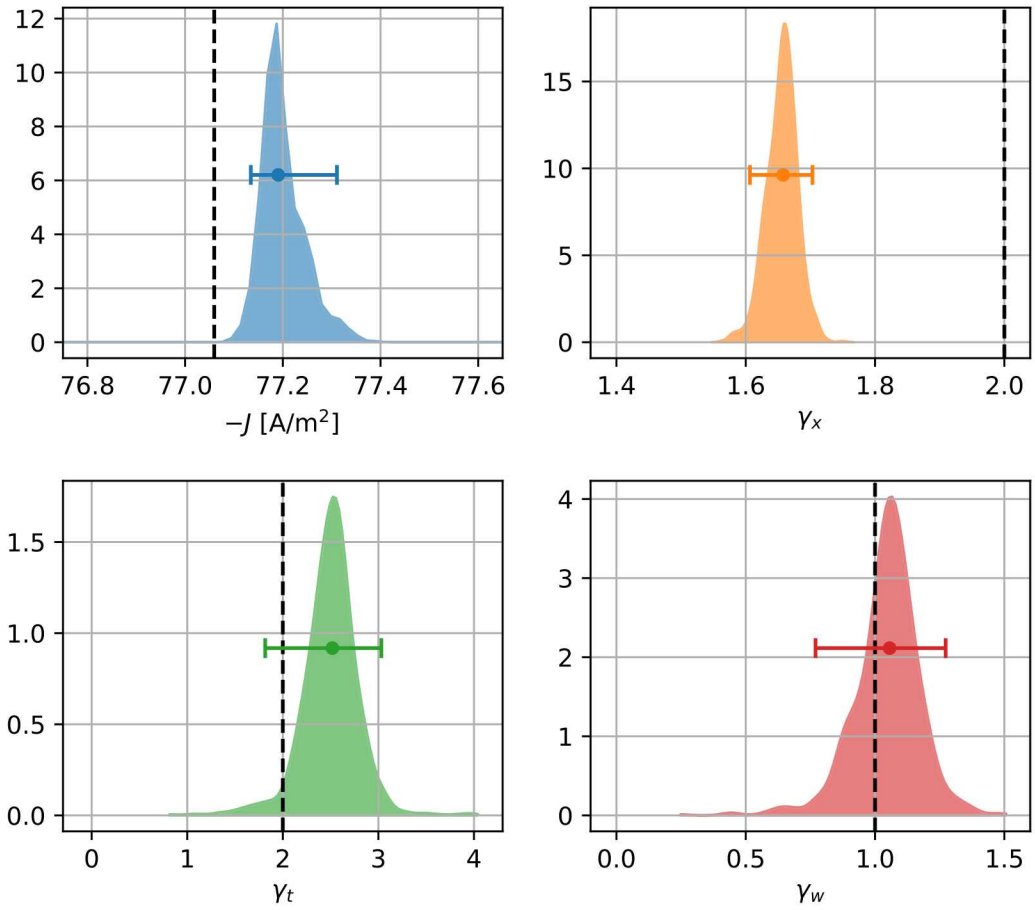


Figure 6-4. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence using the right surface current diagnostic for 9000 simulations with the Verlet algorithm. Black dashed lines indicate exact solution, second-order mesh and time step convergence, and first-order weight convergence.

6.3. LEFT SURFACE DIAGNOSTIC

Most of the same conclusions can be drawn when comparing the raw responses with and without the velocity Verlet algorithm enabled for the left diagnostic as were observed for the right diagnostic. The response data, shown in Figure 6-5, also shows the large increase in numerical bias error. For the StREEQ analysis, an optimal domain was found consisting of 50 out of 64 discretization levels, for which the credibility test passed. The variance model test failed, as was true for the left diagnostic with the velocity Verlet disabled. The analysis for this case is shown in Figure 6-6, which compared to Figure 5-6 indicates degraded mesh convergence which is no longer consistent with second-order.

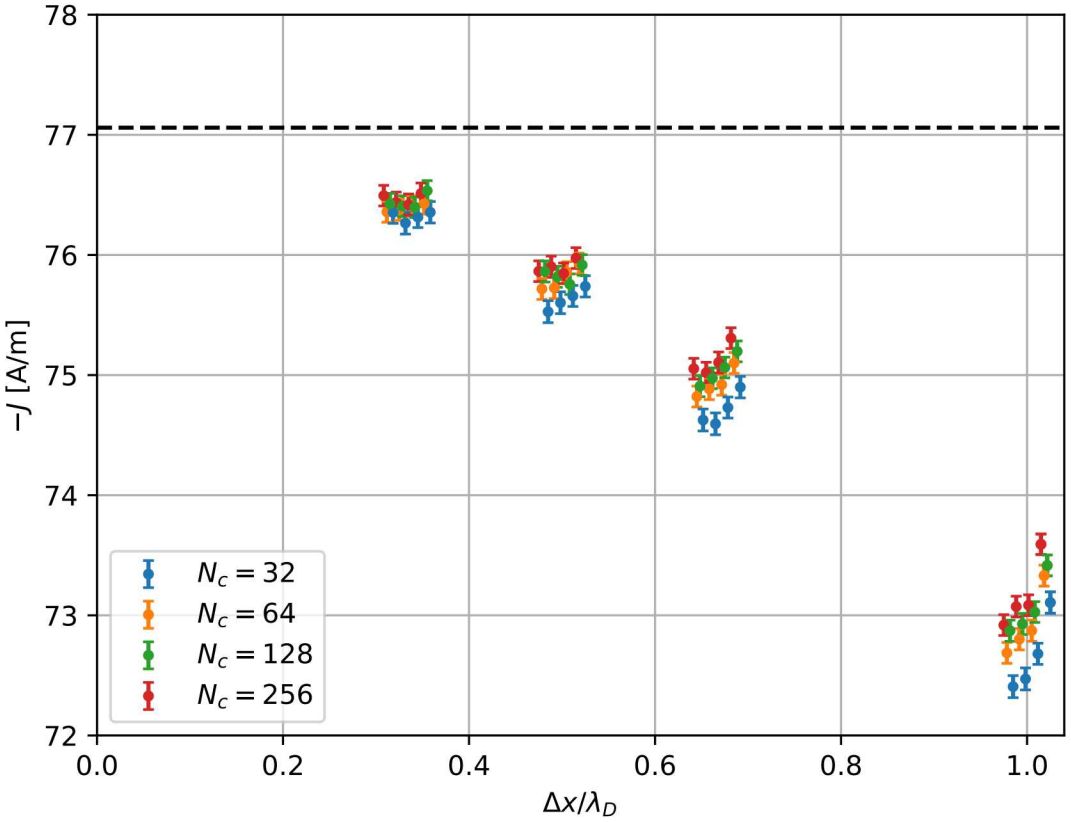


Figure 6-5. Simulation results for the left surface current diagnostic based on 9000 simulations with the Verlet algorithm. The error bars represent 95% confidence intervals in the mean current response. Each cluster of error bars is the data for a single mesh and weight discretization, but variable time step discretization. The exact solution is shown as a black dashed line.

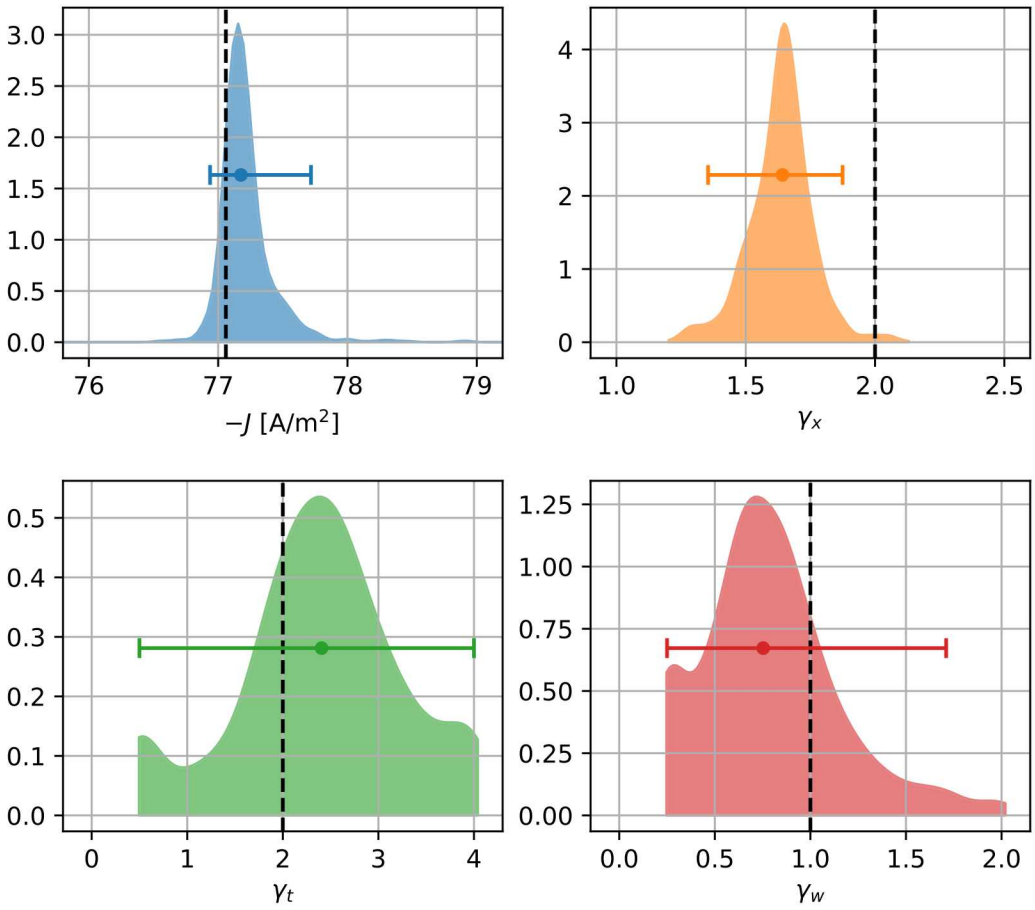


Figure 6-6. StREEQ estimates of the fully-converged solution (blue), and mesh (orange), time step (green), and weight (red) orders-of-convergence of the left surface current diagnostic for 9000 simulations with the Verlet algorithm. Black dashed lines indicate exact solution, second-order mesh and time step convergence, and first-order weight convergence.

6.4. DISCUSSION

The analysis in this chapter differed with that in Chapter 5 only due to the change of time integrator, and therefore we expected different behavior for the global diagnostic but essentially no change to the surface diagnostics. Surprisingly, significant degradation was observed for all three current diagnostics. Specifically, the StREEQ analysis showed a residual numerical bias that was introduced or worsened when simulated with the Verlet time integrator for all three diagnostics, as well as reduced mesh order-of-convergence for two of the diagnostics. For the global diagnostic, the credibility test failed for the Verlet time integrator, while it passed for leapfrog. Differences are also readily observed by comparing the raw data sets directly, particularly for the left and right diagnostics, where the magnitudes of the numerical error $|J - J^*|$ are approximately five times larger for the Verlet algorithm.

7. CONCLUSION

We performed verification tests of EMPIRE simulations of a warm electron diode problem using the StREEQ numerical error estimation tool. This intensive code verification analysis is much more expensive than typical automated tests, but show a much finer level of detail in how the code is performing. In this report, we discussed preliminary code verification efforts (Chapter 4) leading to an initial regression test and verification test, as well as similar tests for a relatively current version of EMPIRE, where three different diagnostics were compared for two different choices of time integrators (Chapters 5 and 6). The overall StREEQ predictions for all three levels are compared in Figure 7-1. Here, the preliminary (prelim), baseline case using the leapfrog (leap) and Verlet algorithms are compared; in the latter cases, for the global (G), right (R), and left (L) surface diagnostics. Comparing the preliminary results to the global diagnostic for the leapfrog and Verlet cases, we see increased bias error, and a slight degradation from first-order mesh convergence. The surface diagnostics do, however, do show increased mesh and time step convergence, although the mesh convergence is still below second-order.

EMPIRE has numerous unit, integration, and regression tests to cover individual aspects of the electron diode problem, including the time integrator, surface emission, and field solve; however, this problem tends to put greater stress on the coupling between these aspects, as well as the diagnostics component, which is a newer, and rapidly developing portion of the code. The lack of time-synchronization of particle positions and velocities in the output is an issue partially addressed by the Verlet algorithm, but at least for the version of the code used to perform the analysis in Chapter 6, this implementation seems to interact with other code components causing degraded convergence behavior. These issues are likely due to both code bugs and technical debt[1] and addressing them should become more of a priority for the development effort. These issues are less likely due to the underlying algorithms themselves, as excellent results [5] for this problem were demonstrated for the Aleph code [1], which employs very similar algorithms to electrostatic EMPIRE.

It would be worthwhile to redo this computationally expensive analysis[2] after substantial code changes until the expected behavior (i.e. second-order mesh and time step convergence, and lack of residual bias error) is observed. After which, this analysis can be used to construct computationally cheaper simulations to ensure that this desired behavior is maintained.

[1] Technical debt refers to the future cost of implementing an easy limited solution rather than a correct approach in order accelerate the development schedule. This concept is critically important for code verification, as incorrectly or incompletely implemented algorithms give suitable results for a limited set of simulations, and would therefore be judged as adequate, but show more severe defects for rigorous code verification exercises.

[2] Here, the expense is due primarily to the large number of EMPIRE simulations, while the computational expense of the StREEQ tool itself is negligible by comparison.

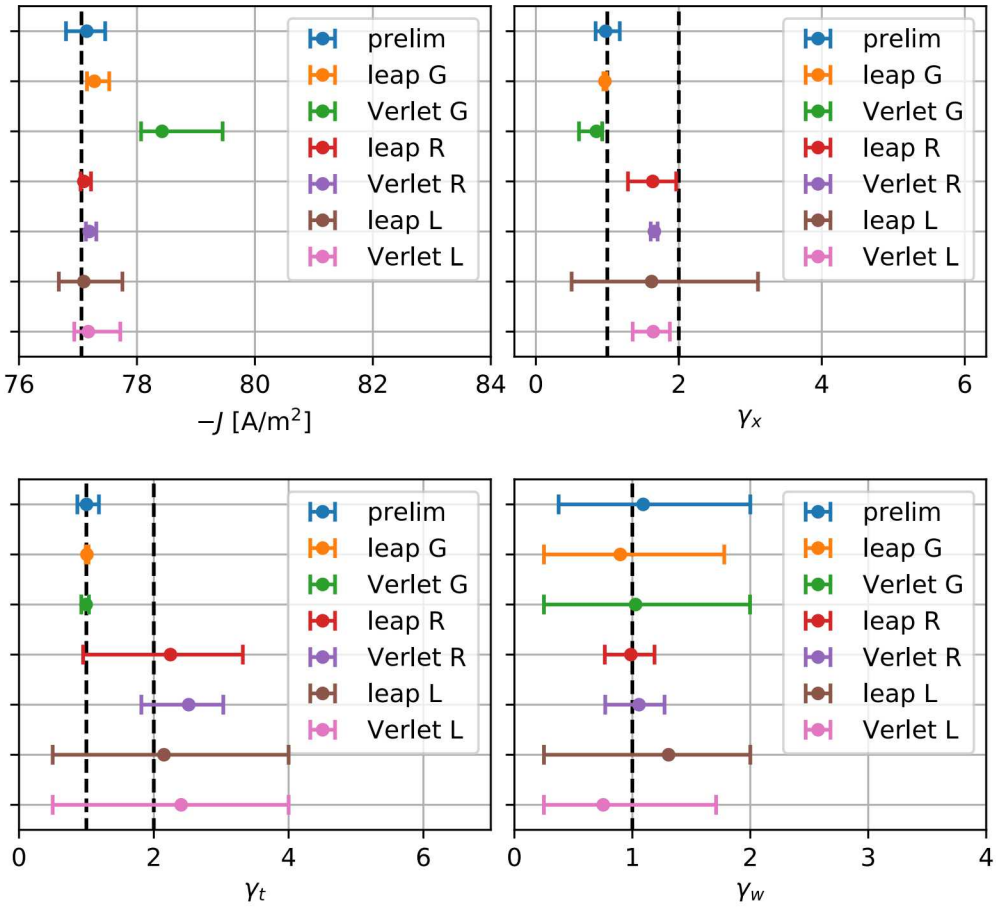


Figure 7-1. StREEQ estimated uncertainty intervals for the fully-converged response and mesh, time step, and weight orders-of-convergence for the code verification test discussed in this report.

REFERENCES

[1] Matthew T. Bettencourt, Jeremiah J. Boerner, Paul S. Crozier, Andrew S. Fierro, Anne M. Grillet, Russell W. Hooper, Matthew M. Hopkins, Thomas P. Hughes, Harold E. Meyer, Christopher H. Moore, Stan G. Moore, Lawrence C. Musson, and Jose L. Pacheco. Aleph Manual. Technical Report SAND2017-10343, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, 2017.

[2] Matthew T. Bettencourt, Richard M. J. Kramer, Keith L. Cartwright, Edward G. Phillips, Curtis C. Ober, Roger P. Pawlowski, M. Scot Swan, Irina Tezaur, Eric Phipps, Sidafa Conde, Eric Cyr, Craig D. Ulmer, Todd H. Kordenbrock, Scott L. Levy, Gary J. Templet, Jonathan J. Hu, Paul T. Lin, Christian A. Glusa, Christopher M. Siefert, and Micheal W. Glass. ASC ATDM level 2 milestone #6358: Assess status of next generation components and physics models in EMPIRE. Technical report SAND2018-10100, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, 2018.

[3] W. L. Oberkampf and C. J. Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[4] Gregg A. Radtke, Keith L. Cartwright, Christopher H. Moore, Nevin Martin, and Andy Huang. StREEQ users manual. Technical report SAND2019-xxxx, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, 2019.

[5] Gregg A. Radtke, Keith L. Cartwright, and Lawrence C. Musson. Stochastic Richardson extrapolation based numerical error estimation with application to kinetic plasma simulations. Technical report SAND2017-4165, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, 2017.

[6] Fabio Riva, Carrie F. Beadle, and Paolo Ricci. A methodology for the rigorous verification of particle-in-cell simulations. *Physics of Plasmas*, 24(055703), 2017.

[7] Patrick J. Roache. *Fundamentals of Verification and Validation*. Hermosa Publishers, 2009.

[8] Christopher J. Roy. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, 205:131–156, 2005.

DISTRIBUTION

Email—Internal [redacted]

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.