INL/JOU-23-74288-Revision-0

Verification and Validation of the Open-Source Plasma Fluid Code: Zapdos

October 2023

Corey DeChant, Casey T Icenhour, Shane Keniley, Grayson Gall, Alexander D Lindsay, Davide Curreli, Steven Shannon

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/JOU-23-74288-Revision-0

Verification and Validation of the Open-Source Plasma Fluid Code: Zapdos

Corey DeChant, Casey T Icenhour, Shane Keniley, Grayson Gall, Alexander D Lindsay, Davide Curreli, Steven Shannon

October 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Verification and Validation of the Open-Source Plasma Fluid Code: Zapdos

Corey DeChant[a], Casey Icenhour[a,c], Shane Keniley[b], Grayson Gall[a], Alexander Lindsay[c], Davide Curreli[b], Steven Shannon[a]

[a]*Department of Nuclear Engineering, North Carolina State University, Raleigh NC, USA*
[b]*Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA*
[c]*Computational Frameworks, Idaho National Laboratory, Idaho Falls, ID, USA*

Abstract

Zapdos is an open-source finite element plasma fluid solver based on the MOOSE multiphysics framework. This paper outlines Zapdos verification, benchmarking, and validation efforts for 1D and 2D RF capacitively-coupled plasma discharge models for mid-range pressures (0.1 - 1 Torr). The verification process involved using the method of manufactured solutions to assess Zapdos spatial and temporal error convergence. L_2 errors ranged from 10^{-2} to 10^{-4}, while the convergence's slope were in agreement with the predicted slopes for the tested variable and time integration orders. The benchmarking process involved comparisons to previously results from the validated finite element code, LSODI [1, 2, 3]. These works included 1D and 2D simulations for a range of plasma parameters (densities, temperatures, voltage, etc.). For the 1D cases, Zapdos and LSODI results were in very good agreement. In the 2D cases, variable behaviors matched, with slight discrepancies in peak values. The validation process involved comparisons to experimental works including electron density measurements by microwave interferometry [4] and metastable density measurements by planar laser-induced fluorescence imaging [5]. Results shown reasonable agreement at higher pressure, with results starting to diverge at low pressures. Probable causes for this diverges are the limitation of the fluid assumption for plasmas at low pressure, or the need for more robust boundary conditions. Overall, Zapdos shown reasonable results for the verification, benchmarking, and validation efforts, and Zapdos can be downloaded at `https://github.com/shannon-lab/zapdos`

Keywords: Plasma, Finite Element Method, Open-Source, Verification and Validation, Benchmarking, Method of Manufactured Solutions, Fluid Approximation

1. Introduction

Open-source software has grown in popularity within the plasma community [6, 7, 8]. This is due to the lack of license fees, accessibility of code for modification or specialization, and often community-driven development processes, just to name a few. Zapdos, based on the open source Multiphysics Object Oriented Simulation Environment (MOOSE), uses the finite element method (FEM) to solve the plasma flux continuity and momentum equations in order to simulate low temperature plasma discharges in atmosphere and vacuum. MOOSE was originally developed for the modeling and simulation of nuclear reactor systems, which involve highly nonlinear, tightly coupled sets of partial differential equations (PDEs) representing the required physics. However, due to the general nature of physics implementation within MOOSE code, multiple applications and physics modules have been developed for different fields of physics [9, 10, 11, 12].

Due to a unified code base with respect to the basic syntax, solvers, and FEM implementation, all MOOSE codes can easily be coupled together. In order to add an expanded plasma chemistry capability, Zapdos was coupled to CRANE, a chemical reaction network solver [13]. Alongside the ease of coupling to other applications, MOOSE (and therefore Zapdos) is designed around solve flexibility. This enables coupling of Zapdos to other physics domains available natively in the MOOSE framework or developed through the dozens of tracked open source MOOSE applications, allowing Zapdos to study currently critical plasma research topics such as plasma material interactions, controllable plasma chemistry, and plasma design within complex process flow systems.[14, 15] While most users of MOOSE will run their code on user workstations with a handful of cores, MOOSE has also demonstrated scalability on over 30,000 of CPU cores. [9] Previous publications demonstrating the capability of Zapdos include a plasma-water simulation involving a DC discharge done by Lindsay et al [16] and plasma-enhanced thermionic energy converters simulations done by Haase and Go [17].

In order for a community to trust the results of any given software, said software should over go the process of verification and validation (V&V). While the plasma community has discuss and demonstrated the importance of V&V in the past [18, 19, 20, 21, 22, 23], the American Society of Mechanical Engineers (ASME) has a well defined guide on the V&V process [24]. In short, verification can be summarized as the testing of the computational process, and validation is the testing of the physical model.

Verification can be divided into two parts, code verification which test if the mathematical model is solving correctly, and calculation verification which test if the mathematical model is accurate. Both parts of verification can be done with the Method of Manufactured Solutions (MMS) and error convergence testing. Validation is done by comparing simulation results to validation experiments. In the low-temperature plasmas, the GEC reference cell was designed for this purpose [25]. This reference source gives the plasma simulation community a well define chamber geometry and discharge parameters for replicating measurements in mid-to-low pressure regimes. Though not official apart of the V&V process, there is often a step between verification and validation, known as benchmarking. Benchmarking is the process of comparing a new software to previous validated software. This test simulations with the same physical assumption, without introduction any errors that maybe introduction with experimental comparisons (both experimental errors and errors due to physical assumptions).

This paper gives an overview of the physics within Zapdos (Section 2.1), the bases of the FEM used (Sections 2.2), the user interface (Section 2.3), the V&V process with benchmarking (Section 3) and conclusions (section 4).

2. Software Description

2.1. Governing Equations

2.1.1. Multi-Fluid Approach

Zapdos uses the multi-fluid approach to model plasma behaviors. At pressures were collisional behavior dominate over particular-to-particular interactions, the different plasma species (electrons, ions, metastables, etc.) can be treated as separate fluids. These fluid equations are then simplified to drift-diffusion equations, where the advection term is driven the E-fields. The flux continuity equations are in the form of:

$$\frac{\partial n_j}{\partial t} + \nabla \cdot \mathbf{\Gamma_j} = \sum \mathbf{S_j} \tag{1}$$

$$\mathbf{\Gamma_{e,i}} = \mu_{e,i}\mathbf{E}n_{e,i} - D_{e,i}\nabla n_{e,i} \tag{2}$$

$$\mathbf{\Gamma}_* = -D_*\nabla n_* \tag{3}$$

where n is the density, j is the subscript for either electrons (e), ions (i) or metastables ($*$), $\mathbf{\Gamma}$ is the flux of the species, $\mathbf{S_j}$ is the source terms, μ is the mobility coefficient, D is the diffusivity coefficient and $\mathbf{E}$ is the electric field. The source terms are handled by MOOSE's plasma chemistry application, CRANE [13]. CRANE is a separate software that may be used to solve global (0D) or spatial (1D-3D) chemical reaction networks based on chemical kinetics (Equation 4). Rate coefficients may be constant values, functions of other variables in the system of equations, or tabulated. Since CRANE is compiled with Zapdos, the source term due to chemical reactions are fully coupled to the transport equations, Equation 1.

Within Zapdos, the source terms can be expressed using either rate coefficients or Townsend coefficients (Equations (4) and (5), respectively).

$$\mathbf{S_j} = vk\prod_j n_j \tag{4}$$

$$\mathbf{S_j} = \alpha|\Gamma_e| \tag{5}$$

where v is the stoichiometric coefficient, k is the rate coefficient, n_j is the densities of the reactant species j, and α is the Townsend coefficient. Since the benchmarking examples in the present work only focus on rate coefficient, the equations representing the plasma chemistry will assume rate coefficients as inputs. Currently, Zapdos solves for the electric field using the electrostatic field approximation, such that $\mathbf{E} = -\nabla V$ (where V is the potential). To solve for the potential, Gauss's Law combined with the electrostatic approximation becomes a Poisson's equation in the form of:

$$-\nabla^2 V = \frac{e(n_i - n_e)}{\varepsilon_0} \tag{6}$$

where ε_0 is the permittivity of free space. The electron mean energy is solved for and is used to determine the value of electron energy dependent variables (such as electron mobility, electron diffusion, and rate or Townsend coefficients). The equation for the electron energy is in the form of:

$$\frac{\partial n_e\epsilon}{\partial t} + \nabla \cdot \mathbf{\Gamma}_\epsilon = e\mathbf{\Gamma_e} \cdot \nabla V - 3\frac{m_e}{m_g}n_e n_g k_{elastic} T_e - \sum_j E_j K_j \tag{7}$$

$$\mathbf{\Gamma}_\epsilon = \frac{5}{3}\epsilon\mathbf{\Gamma_e} - \frac{5}{3}D_e n_e \nabla\epsilon \tag{8}$$

Where ϵ is the mean electron energy, e is the elementary charge, m_e is the electron mass, m_g is the mass of the background gas, n_g is the background gas density, $k_{elastic}$ is the elastic rate coefficient, T_e is the electron temperature (i.e. $T_e = \frac{2}{3}\epsilon$), E_j is the threshold energy of inelastic collisions, and K_j is the inelastic rate coefficient. With the electron energy, the value of energy dependent coefficients are determined by interpolating a user-provided tabulated text file that relates energy to a coefficient. These text files are usually pre-calculated by a Boltzmann solver (such as BOLSIG+ [26]).

Though not shown in this current validation effort, Zapdos can model the plasma surface interaction between boundary (such as plasma-water interactions). This is done by modeling the reactive species on either side of the plasma-water interface, and equating the ratio of the electron density on either side of the interface to a Henry's law coefficient. For more information on this, please visit the work done by Lindsay et al [16].

2.1.2. Boundary Conditions

There are several boundary conditions (BCs) within Zapdos, both integrated and nodal. Integrated BCs are,

as the name implies, integrated over the boundary surface, while nodal BCs are defined at the nodes. The BCs of the particle densities are generally in the form of a flux BC, or a Neumann-style BC, but can also take the form of a Dirichlet BC. For clarity, the different flux BCs in this paper will be defined in "sets" corresponding to the papers from which they were implemented, but this does not mean the user has to use all of the flux BC within one set per simulation (e.g. an electron BC from one set could be used with any ion BC from any other set due to the modularity of Zapdos syntax). The simplest set of flux BC in Zapdos are defined as Lymberopoulos BC, based on the BC used by Lymberopoulos et al [1] and are electron energy independent. These BC are in the from of:

$$\mathbf{\Gamma_e} \cdot \mathbf{n} = \mp k_s n_e - \gamma \mathbf{\Gamma_i} \cdot \mathbf{n} \tag{9}$$

$$\mathbf{\Gamma_i} \cdot \mathbf{n} = -\mu_i n_i \nabla V \cdot \mathbf{n} \tag{10}$$

where k_s is the electron surface recombination coefficient assuming an electron sticking coefficient of unity and γ is the secondary electron coefficient. The next set of flux BC adds electron energy dependence. These BC are defined as Sakiyama BC and are presented on works of Sakiyama et al [27] and Lymberopoulos et al [3]. These BCs are in the from of:

$$\mathbf{\Gamma_e} \cdot \mathbf{n} = \frac{1}{4}\sqrt{\frac{8k_bT_e}{\pi m_e}} n_e - \gamma \mathbf{\Gamma_i} \cdot \mathbf{n} \tag{11}$$

$$\mathbf{\Gamma_i} \cdot \mathbf{n} = -\mu_i n_i \nabla V_{eff} \cdot \mathbf{n}, \text{ if } \nabla V_{eff} \cdot \mathbf{n} \geqslant 0 \text{ else } \mathbf{\Gamma_i} \cdot \mathbf{n} = 0 \tag{12}$$

$$\mathbf{\Gamma}_\epsilon \cdot \mathbf{n} = \left(\frac{5}{3}\epsilon\right)\frac{1}{4}\sqrt{\frac{8k_bT_e}{\pi m_e}} n_e - \left(\frac{5}{3}\epsilon\right)\gamma \mathbf{\Gamma_i} \cdot \mathbf{n} \tag{13}$$

where k_b is Boltzmann's constant. The final set of BCs are defined as Hagelaar BCs and are based on the work of Hagelaar et al [28] as well as Sakiyama and Graves [29], respectively. This set of BC are not present in this work, but they have been used in previous Zapdos publications [16].

For the potential, there are three main BCs that capture DC discharges, radio-frequency (RF) discharges, and dielectric materials. Since the benchmarking simulations in this paper are of RF discharges, we will quickly mention that the DC BCs utilize Kirchoff's voltage law to simulate an external driving circuit. More information on previous Zapdos simulation with DC discharges can be found in the work of Lindsay et al [16]. RF discharges and grounded surfaces are simulated using a Dirichlet-style BC. With Dirichlet BCs, the potential can be set to a constant zero or as a sinusoidal function in time for a grounded or RF surface, respectively. The RF BC is usually in the form of:

$$V = V_{amp}\sin(ft) \tag{14}$$

where V_{amp} is the peak amplitude of the voltage, f is the frequency, and t is the time.

Lastly, dielectric surfaces are based on the dielectric BC used by Lymberopoulos et al [3]. This BC is in the form of:

$$\frac{\partial V}{\partial t} = \frac{d_i}{\varepsilon_i}\left(e\Gamma_i - e\Gamma_e + \varepsilon_0 \frac{\partial \nabla V}{\partial t}\right) \tag{15}$$

Where ε_i and d_i are the permittivity and effective thickness of the insulator.

2.1.3. Log Form of Variables and Scaling Methods

To improve convergence of the residual by making all calculated density values similar in order and to prevent negative densities, the densities are cast in log form and the mean electron energy is cast as a log of mean electron energy density (i.e. as the product of electron density and mean electron energy). This means Zapdos tries to solve the density, mean energy, and potential in the plasma in the form of:

$$N_j = \ln(n_j) \tag{16}$$

$$E_n = \ln(n_e\epsilon) \tag{17}$$

$$V = V \tag{18}$$

Along with casting the density variables into log form, Zapdos also has scaling methods in place. These include having the option to solve for the potential in terms of volts or kilovolts, setting the units of the density to moles/m^3, and creating a unity mesh then scaling the any gradients accordingly. The idea behind these scaling methods is to reduce the variation in magnitudes of the variables being solved for (and thus the entries in the Jacobian matrix) which should improve the matrix conditioning and thus help improve the convergence of the iterative solver.

2.1.4. Acceleration Scheme

In order to properly resolve plasma oscillations, very small time steps must be taken when using a traditional constant timestepping scheme. Because neutral species (such as metastables) move much slower than electrons, this means that *many* timesteps must be taken in order to properly resolve their motion. This can lead to simulations that take many hours or days on standard workstation hardware. To reduce the run time of simulations of RF discharges involving neutral particles, a modified shooting method technique was implemented. Zapdos' version of the shooting method scheme is based on work presented by Lymberopoulos and Economou [1], but a more general description of the method can be found by Gogolides, Sawin and Brown [30]. This scheme takes advantage of the fact that an RF discharge will eventually have a periodic steady-state, in the form of:

$$N_j(0) - N_j(T) = 0 \tag{19}$$

where $N_j(0)$ is the density at the beginning of the steady-state cycle and $N_j(T)$ is the density at the end of the cycle. Now, to obtain a guess at the periodic solution of $N_j(0)$,

the Newton-Raphson method can be applied and results in:

$$N_j(0)^{new} = N_j(0)^{old} - J^{-1}(N_j(0)^{old} - N_j(T)) \quad (20)$$

$$J = I - \left(\frac{\partial N_j}{\partial N_j(0)}\right) \quad (21)$$

where I is the identity matrix and $\frac{\partial N_j}{\partial N_j(0)}$ is known as the sensitivity matrix. The sensitivity matrix can be calculated by:

$$\frac{d}{dt}\left(\frac{\partial N_j}{\partial N_j(0)}\right) = \frac{\partial F}{\partial N_j}\frac{\partial N_j}{\partial N_j(0)} \quad (22)$$

where F is the force function of the accelerated density and $\frac{\partial N_j}{\partial N_j(0)} = I$ is at the beginning of the cycle. With this method, the acceleration scheme is as follows:

- The main simulation runs for X amount of RF cycles.
- After X cycles, the simulation runs for one RF cycle to calculate $N_j(T)$ and the sensitivity matrix.
- The metastable density is then accelerated by Equation (20) and the new density is sent to the main simulation.
- The main simulation runs for another X amount of RF cycles and accelerates again.

2.2. Numerical Method

To better understand the format of the input files used in MOOSE/Zapdos, a base understanding of the FEM would be beneficial. A fundamental concept of the FEM is the process of converting the differential form of an equation (known as the strong from) into a weak form, where the equation has solutions only with respect to certain "test functions" that also are able to loosely satisfy the strong form of the equation. This paper will demonstrate the generation of a weak form and how it is reflected in MOOSE code and input files by using the classic diffusion problem, in the form of:

$$\nabla \cdot -D\nabla u = s(\mathbf{r}) \quad (23)$$

Where D is the diffusion coefficient, u is the species of diffusion, and $s(\mathbf{r})$ is the source term. Equation (23) would be classified as the strong form of the diffusion problem. To convert this into the weak form for FEM, one must multiply the equation by a test function ψ_i existing in the same space as the solution variable and integrate over the whole domain of interest. All the terms will also be moved over to the left-hand side (LHS), resulting in the following:

$$\int_\Omega \psi_i \nabla \cdot -D\nabla u dV - \int_\Omega s(x)dV = 0 \quad (24)$$

Where Ω is the extent of the domain. Now the first term can be treated by using integration by parts, which gives:

$$\int_\Omega \nabla\psi_i \cdot D\nabla u dV - \int_{\partial\Omega} \psi_i D\nabla u \cdot \mathbf{n} dS - \int_\Omega s(x)dV = 0 \quad (25)$$

where $\partial\Omega$ is the domain boundaries and $\mathbf{n}$ is the normal vector to the boundary. This equation can also be called the "weighted residual form", and the simulation goal is to find a solution which can minimize the LHS to zero. Equation (25) contains what MOOSE calls "kernels" (e.g. the first and third term) and boundary conditions (e.g. the second term) that are entered into MOOSE/Zapdos. The rest of the FEM involves the discretization of the equation from domain integrals into a summation over mesh elements. Further, the test function applied within each element must be determined. The choice of test function usually corresponds to the type of problem being solved (e.g., whether the solution will be continuous or discontinuous). For Zapdos, the continuous Galerkin finite element method is applied, utilizing linear Lagrange test functions. The solution of the resulting system of equations is obtained via Newton's Method.

2.3. User Interface

The input files for Zapdos can be broken down into sections known as "object blocks". A summary of these object blocks are:

- GlobalParams: Defines parameters that appear in multiple blocks in a given input file (such as scaling options).
- Mesh: Creates a mesh, reads existing mesh files, or modifies existing mesh.
- Problem: Defines the type of problem (e.g. FE Problem, Eigenvalue problems, etc.) and defining the coordinate system.
- Variables: Defines the variable being solved for and their test functions.
- Kernels: Defines the physics of the problem.
- AuxVariables and AuxKernels: Similar to Variables and Kernels, except used for parameters that are not directly solved for (e.g. current, power, etc.).
- BCs: Defines the boundary conditions.
- ICs: Defines the initial conditions as a uniform constant or as a function of position.
- Materials: Defines material properties (e.g. mass, mobility, diffusion, rate coefficients, etc).
- Postprocessors: Defines a variety of postprocessing operations.

- Preconditioning: Defines any preconditioning processes needed to solve the problems (the default is SMP (Single Matrix Preconditioner)).
- Executioner: Defines whether the problem to be solved is transient or steady-state, as well as various solver settings.
- Outputs: Defines the type of file used for the output file (the default is an exodus file).

To demonstrate how an object block is structured, the following is an example of a Kernel block with only the electron flux continuity. By transforming the strong form of the electron flux continuity (i.e. Equation (1)) into the weak form results in:

$$\underbrace{\int_\Omega \psi_i \frac{\partial n_e}{\partial t} dV}_{\text{ElectronTimeDerivative}} + \overbrace{\int_\Omega \nabla\psi_i \cdot \mu_e \nabla V n_e dV}^{\text{EFieldAdvectionElectrons}} + \underbrace{\int_\Omega \nabla\psi_i \cdot D_e \nabla n_e dV}_{\text{CoeffDiffusionElectrons}} + \overbrace{\int_{\partial\Omega} \psi_i \nabla\mathbf{\Gamma_e} \cdot \mathbf{n} dS}^{\text{BCs}} - \underbrace{\int_\Omega \psi_i \sum \mathbf{S_j} dV}_{\text{CRANE Kernels}} = 0 \tag{26}$$

with the labels designating the Zapdos object corresponding to the term in the PDE. Listing 1 shows an example of a residual coded into MOOSE and Zapdos.

Listing 1: Residual for CoeffDiffusionElectrons

```
CoeffDiffusion::computeQpResidual()
{
    return -_diffusivity[_qp] * std::exp(_u[_qp]) *
    _grad_u[_qp] * _r_units * -_grad_test[_i][_qp] *
    _r_units;
}
```

In the attempt to make adding all of the required physics easier for users, there are object blocks called Actions. Actions represent a form of automation in MOOSE/Zapdos, where input file objects and parameters can be added and set, respectively. For example, an action could be created to add the appropriate variables, materials, and kernels corresponding to Equation (26). In Zapdos and CRANE, the "DriftDiffusionAction" and "Reaction" actions exist to assist the user, which are utilized to input the plasma drift-diffusion equations (outlined in 2.1) and plasma chemistry, respectively. The "Reactions" block automatically parses the listed reactions into reactants, products, and rate coefficients, and adds the necessary source and sink terms to each species (Equation 4). Rate coefficients denoted "EEDF" are tabulated against the mean electron energy and are linearly interpolated during the simulation. If the reaction acts as an energy source or sink for electrons, the change in mean electron energy associated with that reaction may be listed in brackets (in units of eV), and the resulting source or sink term is added to Equation 7. Listing 2 shows an example of these actions and their syntax for an argon discharge. A detailed list and description of all current Zapdos objects can be found on the Zapdos website [31].

Listing 2: Actions for an Argon Plasma

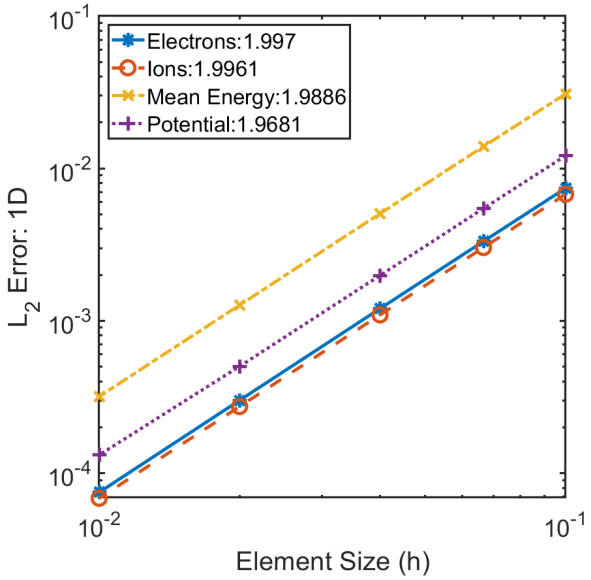
```
# Action that sets the Drift Diffusion Equations
# and Potential Equations
[DriftDiffusionAction]
[./Plasma]
electrons = em
charged_particle = Ar+
Neutrals = Ar*
potential = potential
Is_potential_unique = true
mean_energy = mean_en
position_units = ${dom0Scale}
Additional_Outputs = 'ElectronTemperature Current EField'
[../]
[]
#Action that sets the Source Terms of the species
[Reactions]
[./Argon]
species = 'Ar* em Ar+'
aux_species = 'Ar'
reaction_coefficient_format = 'rate'
gas_species = 'Ar'
electron_energy = 'mean_en'
electron_density = 'em'
include_electrons = true
file_location = 'rate_coefficients'
potential = 'potential'
use_log = true
position_units = ${dom0Scale}
block = 0
reactions =
'em + Ar -> em + Ar*         : EEDF [-11.56] (reaction1)
 em + Ar -> em + em + Ar+    : EEDF [-15.7] (reaction2)
 em + Ar* -> em + Ar         : EEDF [11.56] (reaction3)
 em + Ar* -> em + em + Ar+   : EEDF [-4.14] (reaction4)
 em + Ar* -> em + Ar_r       : 1.2044e11
 Ar* + Ar* -> Ar+ + Ar + em  : 373364000
 Ar* + Ar -> Ar + Ar         : 1806.6
 Ar* + Ar + Ar -> Ar_2 + Ar  : 39890.9324'
[../]
[]
```

3. Verification and Validation

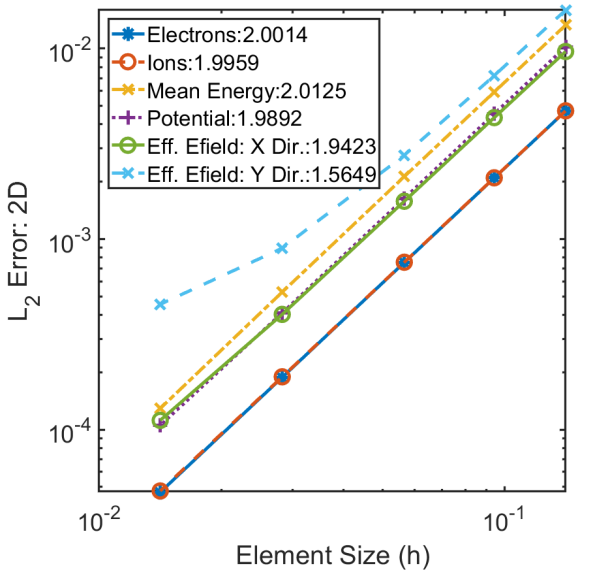
3.1. Method of Manufactured Solutions

MMS is a common and straight forward way of verifying code [18, 19, 20]. The principle behind MMS is that one can verify a code by manufacturing an artificial solution and deriving the necessary inputs needed for one's code to simulate the same solution. By using MMS, one can verify within a relative amount of certainty that the code is solving the intended system of equations correctly. Within Zapdos, there are tests for different degrees of coupling the three main types of variables (i.e. particles, mean energy, and potential). The different MMS tests include: basic diffusion, diffusion with advection, single charged density with potential, multiple charged densities with potential, and multiple charged density with potential and electron mean energy. When constructing the manufactured solutions, it is important for the solutions to be smooth and have non-zero derivatives to the order required by the system. To avoid this in Zapdos, trigonometric functions were

used. A physical constraint on the chosen verification solution required that the densities and mean energy be non-negative and non-zero. To simulate the oscillation that would occur in the electrons under an RF field, the chosen solutions of the electron density and mean energy were set to oscillate in a sinusoidal fashion with time, similar to the potential boundary conditions.



(a) 1D Results



(b) 2D Results

Figure 1: MMS Spacial Regression Plot

Code performance and faithful reproduction of the MMS solution can be verified by reviewing the error behavior over a range of timestep sizes and element size. One should expect a convergence rate of two for first order elements and a second order time integration, as is used in these examples [20]. For brevity, the convergence study for only

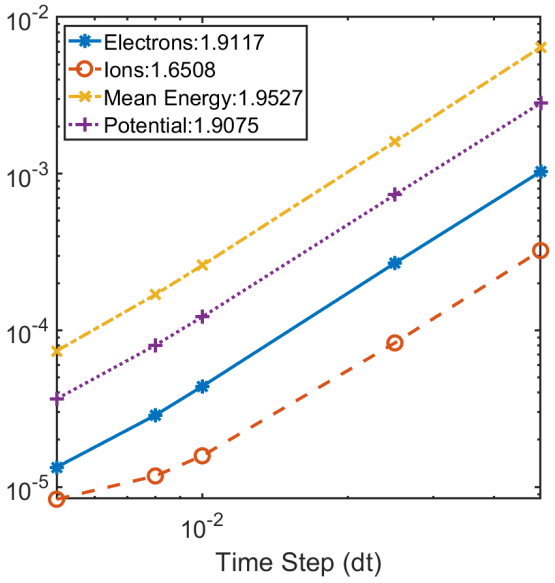


Figure 2: MMS Temporal Regression Plot

the fully coupled plasma fluid equations are presented in this work.

3.1.1. MMS: Results

Figure 1 demonstrates the spatial convergence, while figure 2 shows the temporal convergence. For the 2D simulations, an "effective" electric field was added for the ion motion, as defined in Equation (27). The converge rates match closely to the theoretical values, with outliers of the y-component of the effective electric field for the spacial regression and the ions for the temporal regression. An explanation for these outliers is that some terms for these solutions are small compared to other terms (e.g. for ions the diffusion term being greater than the advection term). Ideally, one would want all the terms to be near the same order when deriving solutions for MMS, but this becomes more difficult as the number of coupled equations increases and an optimal value for each coefficient must be found. When coupling and solving for multiple solutions at once, this can become tedious or improbable while upholding the constraints mentioned before.

3.2. Benchmarking Cases

The benchmarking cases in this paper involve comparisons to RF discharge simulations done by Lymberopoulos in 1D [1] and 2D [2, 3]. Both RF simulations are of an argon discharge with the plasma chemistry shown in table 1 (the only exception is for the 1D case, which excludes elastic collisions to better match the simulation conditions in the paper [1]).

3.2.1. 1D Discharge Conditions

The 1D demonstration case was a simple 2.54 cm CCP discharge for argon at 1 Torr and 300 K ($N = 3.22 \times$

Reaction	E_j (eV)	Rate Coefficient	Ref.
Ar+e→ Ar+e	Elastic	Energy Dependent	[3, 32]
Ar+e→ Ar*+e	-11.56	Energy Dependent	[1, 32]
Ar+e→ Ar^++2e	-15.7	Energy Dependent	[1, 32]
Ar*+e→ Ar^++2e	-4.14	Energy Dependent	[1, 32]
Ar*+e→ Ar+e	11.56	Energy Dependent	[1, 32]
Ar*+e→ Ar^r+e	...	$2 \times 10^{-13}\ \frac{m^3}{s}$	[1, 33, 34, 35]
Ar*+Ar*→ Ar^++Ar+e	...	$6.2 \times 10^{-16}\ \frac{m^3}{s}$	[1, 33, 34, 35]
Ar*+Ar→ 2Ar	...	$3 \times 10^{-21}\ \frac{m^3}{s}$	[1, 33, 34, 35]
Ar*+2Ar→ Ar_2+Ar	...	$1.1 \times 10^{-43}\ \frac{m^6}{s}$	[1, 33, 34, 35]

Table 1: Reactions Rates

Name	Symbol (unit)	Value	Ref.
Electron diffusivity	$ND_e(ms)^{-1}$	3.86×10^{24}	[1, 36]
Electron mobility	$N\mu_e(Vms)^{-1}$	9.66×10^{23}	[1, 37]
Ion diffusivity	$ND_+(ms)^{-1}$	2.07×10^{20}	[1, 38]
Ion mobility	$N\mu_+(Vms)^{-1}$	4.65×10^{21}	[1]
Metastable diffusivity	$ND_*(ms)^{-1}$	2.42×10^{20}	[1, 36]
Secondary electron coefficient	γ	0.01	[1, 36]

Table 2: Flux Coefficients

$10^{22}\ m^3$). The mesh was the same as presented in [1]. The mobility and diffusion coefficients for all species were constant and are summarized in table 2. The boundary conditions for the electrons and ions used in the 1D case were the Lymberopoulos BC set with a k_s of $1.19 \times 10^5\ \frac{m}{s}$ and a γ of 0.01 (Equations (9) and (10)). The potential at the electrodes was set with the RF BC for $V_{amp} = 100\,V$ and $f = 2\pi(13.56 \times 10^6)s^{-1}$ on one electrode and grounded ($V_{amp} = 0$) on the other electrode. The metastables at the electrodes were defined by a Dirichlet BC equaling close to zero (in all cases, this was $100\,m^{-3}$), and mean electron energy was also defined by a Dirichlet BC set to $\frac{2}{3}(0.5)eV$. Two sets of runs were performed, one without metastables and the other with metastables. To help speed up the simulation to steady-state for the run case with metastables, the acceleration method mentioned in Section 2.1.4 was employed. To help with the stability of the accelerator, the acceleration method employed an offset to the sensitivity matrix such that the max entry for J^{-1} in Equation (21) is 100. This value was chosen based on several runs with differing offsets, where a value of 100 shown to maximize acceleration without inducing instability for the set of conditions in this work.

3.2.2. 1D Discharge: Results

The simulation ran for 5000 RF cycles while accelerating the metastable density every 50 RF cycles. Argon simulation with metastables can take up to 10^5 RF cycles to reach steady-state, due to the comparatively slow response time of the metastable to the charged particles [1]. While 5000 RF cycles to reach steady-state is significantly less than the 10^5 cycles need without an accelerator, this is approximately 5 times large then the cycles needed to reach steady-state by the previous work, which stated a steady-state solution at about 1000 RF with a similar accelerator scheme [1]. While the exact cause of the discrepancy in required RF cycles is not know yet, a probably cause is the difference in how the transient solution is solved between Zapdos and LSODI (the FEM software used for benchmarking). Zapdos simulations were solved used a second order backward difference formula, while LSODI used backward difference formula with automatically variable order. While this difference in time integrals should not drastically affect the steady-state solutions, this could affect the transient results and the effectiveness of the accelerator. Figure 3 shows the spatial average growth for the densities with the accelerator. The hump present in the figure has been shown before in other accelerated simulations by Kushner et al [39, 40]. They attribute this phenomenon to an "over acceleration" of the accelerated density with is cased by a high value for their acceleration factor. This would be analogous to computing a high values in the sensitivity matrix near the beginning of the simulations.

Figure 4 shows the results for the case without metastables, while figures 5 and 6 are the results for the case with metastables. All cases are in good agreement between Zapdos results (the line data) with Lymberopoulos's results (the point data). There is a slight increase for the metastable density near the electrodes when compared the previous simulation efforts. This is mainly due to converting the densities into log form. Converting to log form requires higher mesh refinement to properly capture trajectory of variables heading close to zero, such as the boundary condition for the metastables.

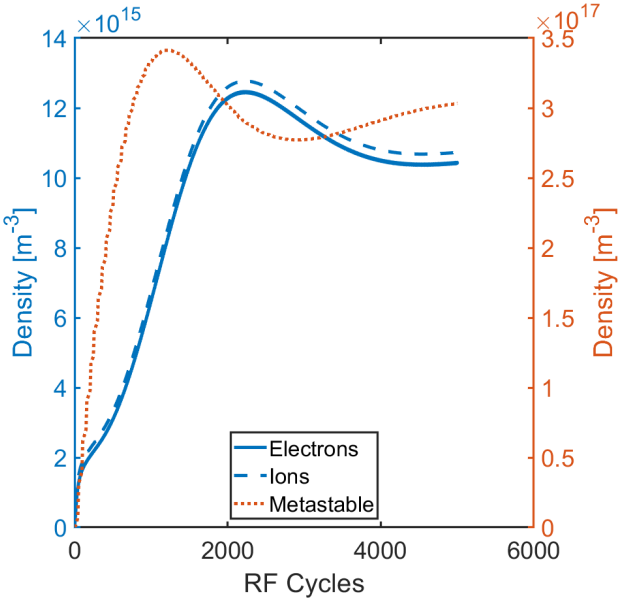
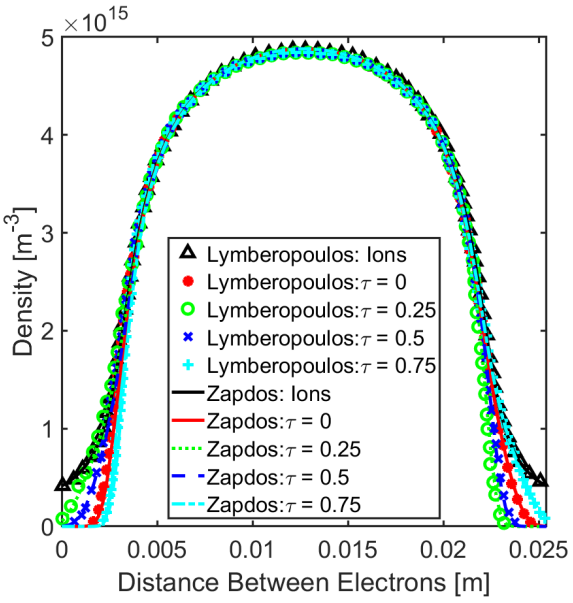
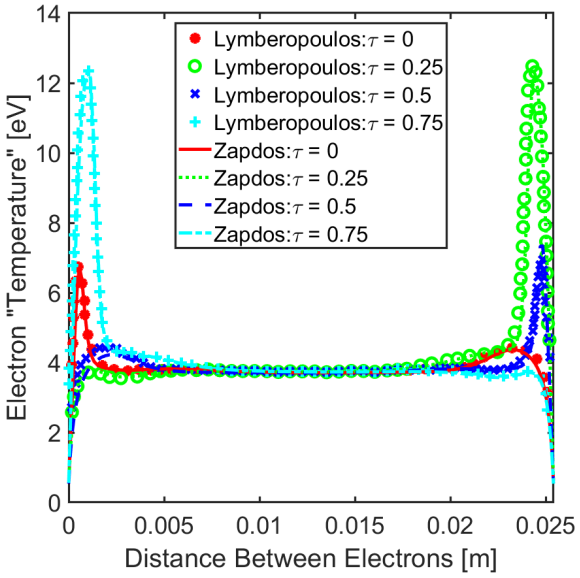


Figure 3: Spatial Average Growth With Accelerator



(a) Electron and Ion Density



(b) Electron "Temperature"

Figure 4: Case without Metastables

3.2.3. 2D Discharge Conditions

The 2D benchmarking case was a 1 Torr and 100 mTorr argon plasma in the GEC reference cell. Zapdos ran the simulation as an axisymmetric geometry found in figure 7 [3]. The mobility and diffusion coefficients for the electrons, ions, and metastables were the same as for the 1D demonstration case. The boundary conditions for the electrons, ions, and mean electron energy at all surfaces were the Sakiyama BC set with the same γ of 0.01 (Equations (11),(12), and (13)) and the metastables were again set as a Dirichlet BC equaling close to zero (in all cases, this was 100 m^{-3}). For the 1 Torr case, the potential at the electrodes was configured so that the peak to peak voltage was 60 V. This was done by setting the electrodes with the RF BC for $f = 2\pi(13.56 \times 10^6)\ s^{-1}$, $V_{amp} = 30V$ for the top electrode and $V_{amp} = -30V$ for the bottom electrode. For the 100 mTorr case, the potential was setup similarly except for a peak to peak voltage of 100 V. The insulator was set with the dielectric BC of Equation ((15)), where the permittivity and effective thickness of the insulator was 1.859382×10^{-11} Fm^{-1} and 0.0127 m, respectively.

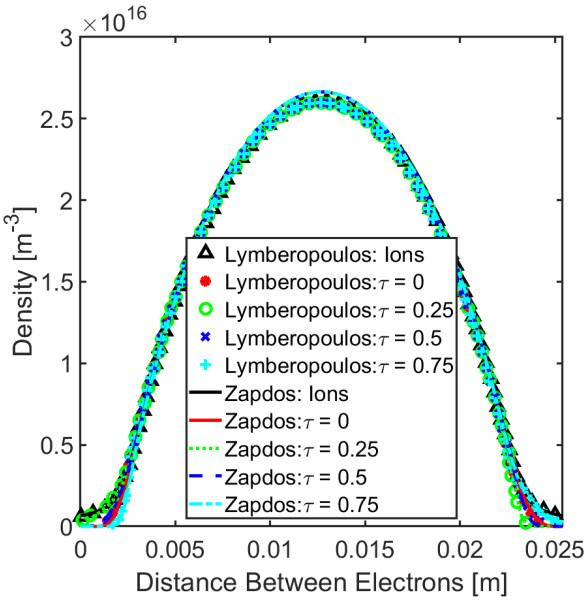
Along with the physics mentioned in Section 2.1.1, the ion advection was determined by effective electric field, that was defined by:

$$\frac{\partial \mathbf{E_{eff}}}{\partial t} = \nu_+(\mathbf{E} - \mathbf{E_{eff}}) \tag{27}$$

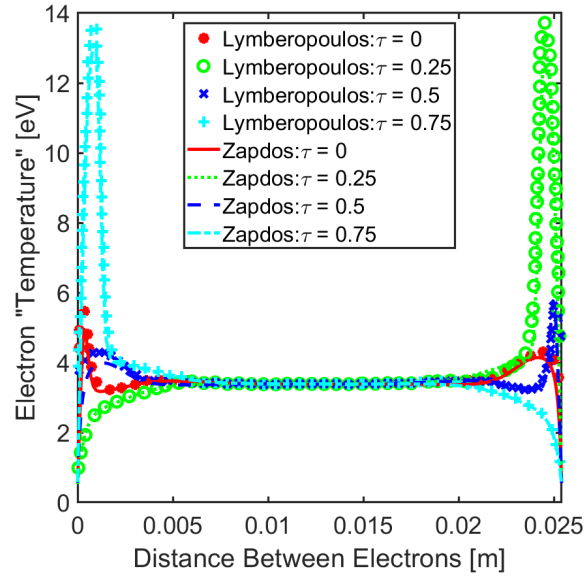
Where ν_+ is the ion-neutral collision frequency and all electric fields assume the electrostatic field approximation. The simulation used a 2D structured mesh that contained approximately 3000 nodes, with slight refinement between the electrode and insulator. Both simulations run for about 1000 RF cycles, about 96 hours on an HPC system, utilizing 45 cores. Results for the 1 Torr and 100 mTorr case are shown in figure 8 and figure 9, respectively. Zapdos results are displayed in color, while previous simulations results are white contour lines.

3.2.4. 2D Discharge: Results

For both density regimes, the results profiles are similar between Zapdos and previous results. There are slight discrepancies in the peak values, with Zapdos over-predicting approximately 20%. This is most likely due to the difference of the effectiveness of the accelerator between Zapdos and LSODI, as mentioned in section 3.2.1. Neither sets of simulations ran to the number of RF cycles needed to reach steady-state demonstrated by the 1D cases. Zapdos ran for 1000 RF cycles while the 1D case required 5000 cycles, and



(a) Electron and Ion Density



(b) Electron "Temperature"

Figure 5: Case with Metastables

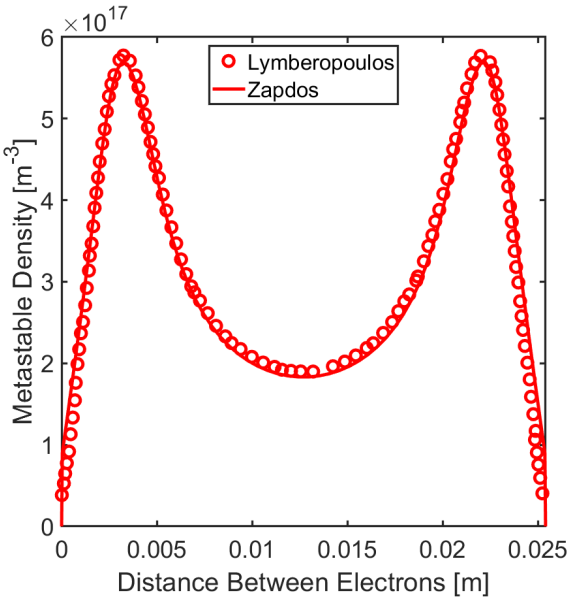


Figure 6: Metastable Density

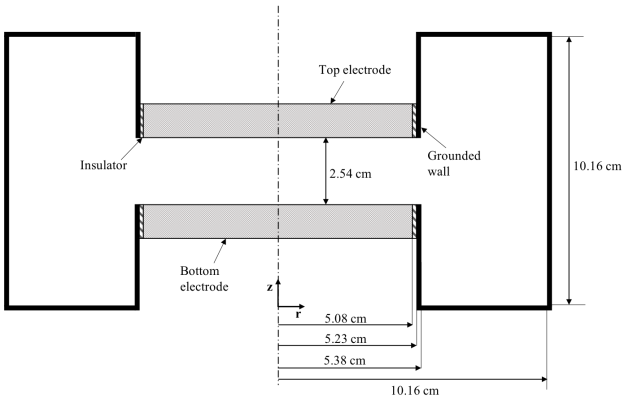


Figure 7: GEC Reference Cell Figure

the previous LSODI simulations ran for 500 RF while the 1D case required 1000. The reasoning for stopping Zapdos at 1000 cycles were due to wall time. A possible reason for the constant over-prediction in all sets of measurements is the hump in Figure 3. Zapdos results could still be in the region were the accelerator over predicts values, compared to the accelerator used in the benchmarking cases.

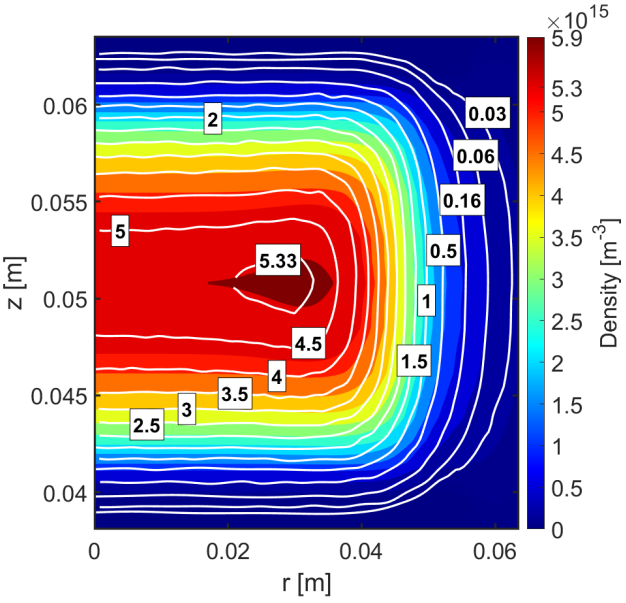
It should also be noted that during this validation effort, it was noticed that both the rate coefficients and electron flux coefficients had a large effect on the plasma potential. This was especially true when comparing the effects on plasma potential between constant electron flux coefficient and energy dependent coefficients.

3.3. Experimental Validation

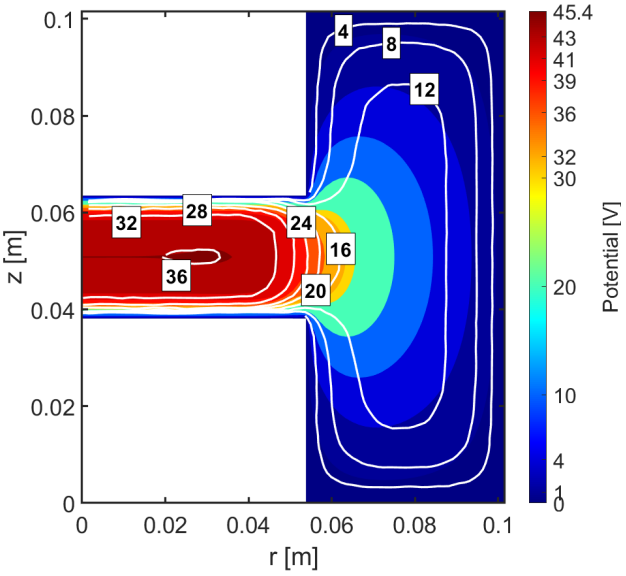
Two sets of previous experimental works were used for the validation process. Both sets involved a parametric studies for a range of applied voltages and pressure within the GEC reference cell. The main difference is the discharge configuration and method of diagnostics. The first set of experimental date were of electron density measured by microwave interferometry for a push-pull discharge in works by Greenberg [4]. The second set of experimental data were of relative metastable density measured by planar laser-induced fluorescence imaging for a single powered plate discharge in works by McMillin [5].

3.3.1. Experimental Validation: Results

Figure 10 shows the electron density between Zapdos (lines) and experiments (points) for a range of pressures and voltages. At higher pressures, Zapdos shows good agreement with the data set, and then Zapdos results starts to diverge at low pressures. This divergence has

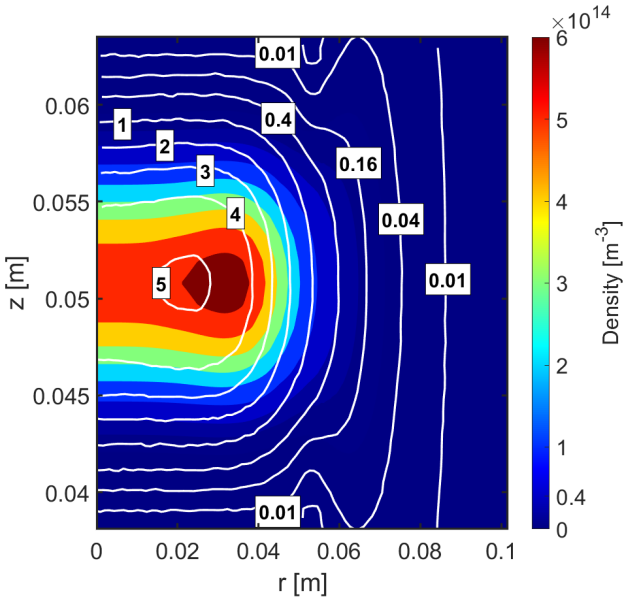


(a) Time Average Electron Density for 1 RF Cycle

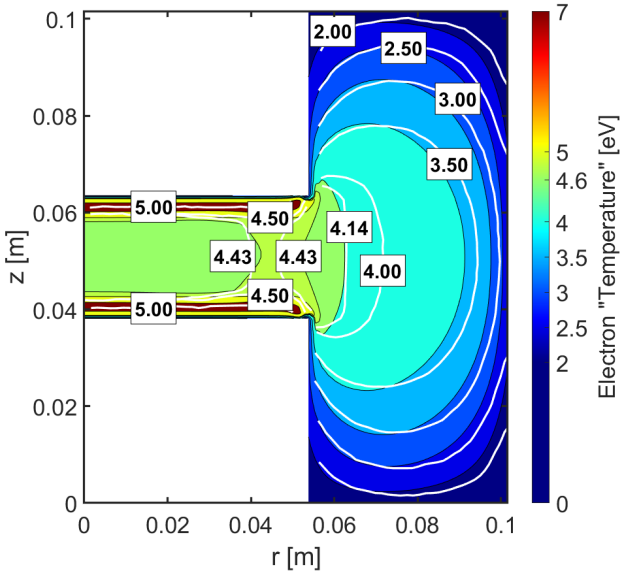


(b) Time Average Potential for 1 RF Cycle

Figure 8: 2D GEC Results at 1 Torr



(a) Time Average Electron Density for 1 RF Cycle



(b) Time Average Electron "Temperature" for 1 RF Cycle

Figure 9: 2D GEC Results at 100 mTorr

two probably causes, a break down of the fluid approximation or the need for more robust boundary condition. For the first case, the fluid approximation can start to break down at lower pressures due the lack of collisions between particles. The main assumption of the fluid approximation is that the collisional effects dominate over particle-to-particle interactions, but this is not necessarily true for low pressures. For second case, plasma bulk electron density is often proportional to pressure, resulting in lower bulk densities for lower pressures. With low bulk densities, the effects at the boundaries become more significant. These effects can include secondary electron emission (where ions have enough energy to pull electrons from surfaces), or electron being reflected off the electrodes back into the plasma, instead of completing the circuit.

Figure 11 and 12 shows the results for metastable densities for an applied voltage of 75 V_{pp} and 150 V_{pp} at 1 Torr and 100 mTorr, respectively. Zapdos results are displayed in color, while previous experimental results are white contour lines. Zapdos ran for approximation 250 - 500 RF cycles. For the 1 Torr case, Zapdos matches experimental trends with a slight discrepancy in the location of the peak density. In terms of errors in the modeling, is discrepancy could be do to the lack of a self bias voltage in the boundary condition. In a single powered electrode discharges, the plasma can create a self bias voltage at the

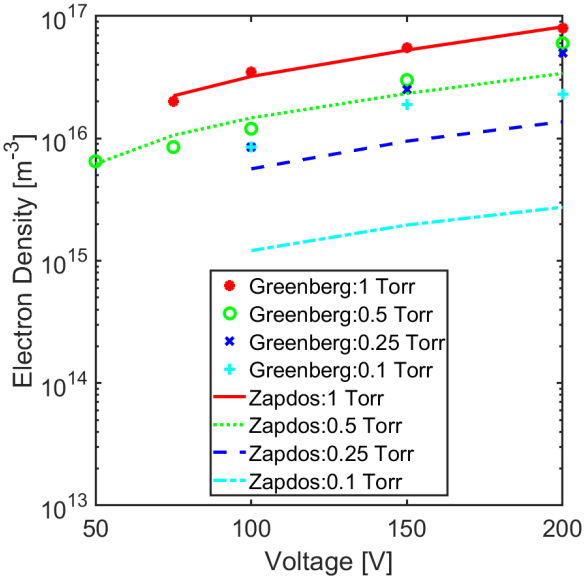
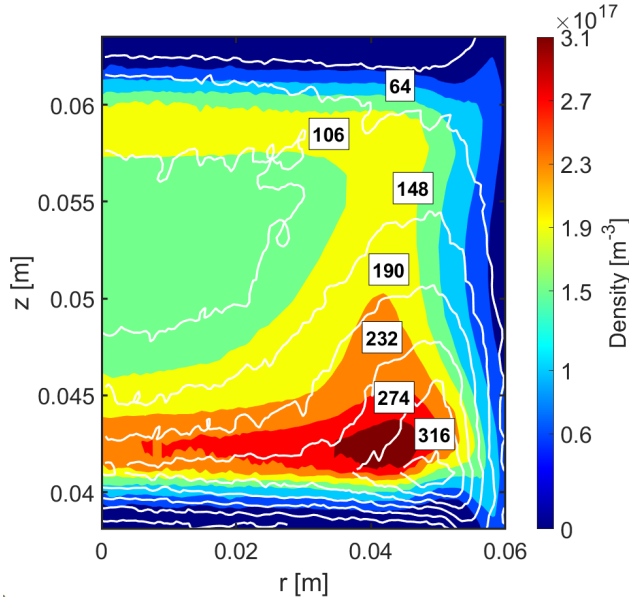
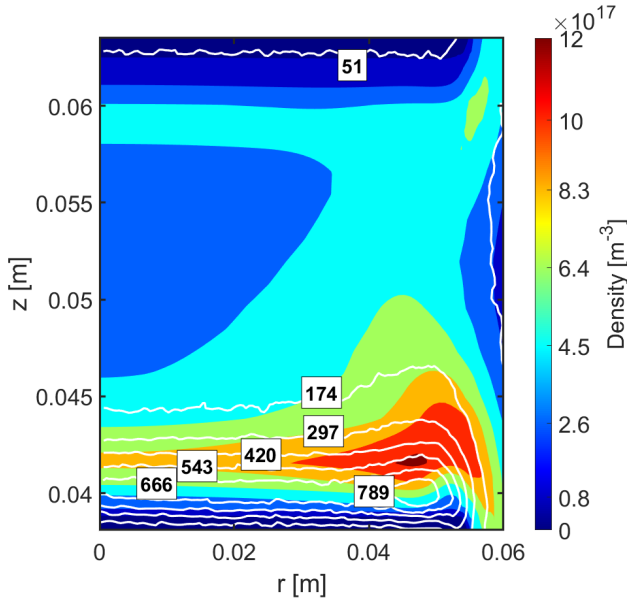


Figure 10: Electron Density



(a) Power Electrode at 75 V_{pp}



(b) Power Electrode at 150 V_{pp}

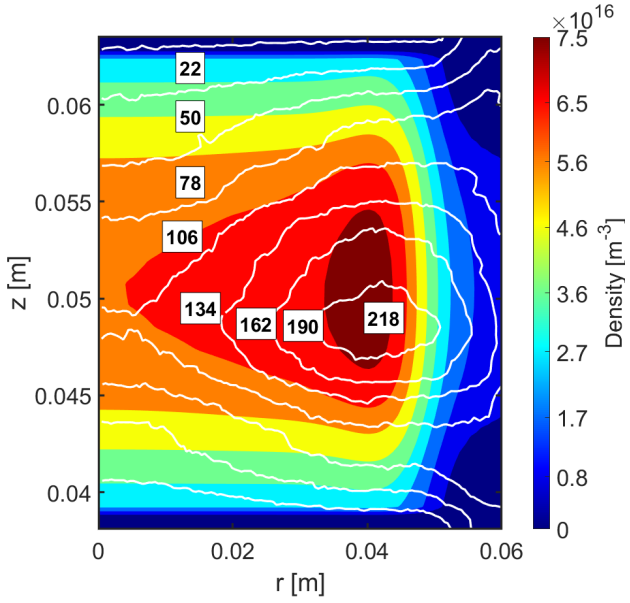
Figure 11: Metastable Density at 1 Torr

powered electrode. This results in a higher time average voltage at the boundary, which could affect densities values near that electrode. For the 100 mTorr case at 75 V_{pp}, the density profile at near the edges of the electrodes are closer together than the experimental data set. This error is most likely due to residual initial condition effects cause by the slower react of species at lower pressures and voltages. In terms of pressure, there are fewer collisions between particles, resulting in lower diffusion coefficients for species. In terms of voltage, the charges species' advection is driven by voltage, thus a lower voltage results in a lower advection term.

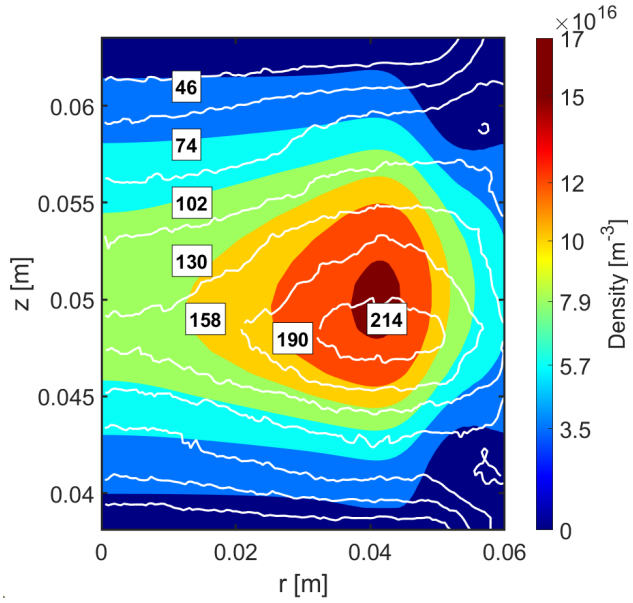
4. Conclusions and Future Work

The V&V process with benchmarking was carried out for the Zapdos, an open-source finite element plasma fluid solver. For verification, both code and calculation verification were performed with MMS with spatial and temporal error convergence studies. L_2 errors ranged from 10^{-2} to 10^{-4}, while the convergence's slope were in agreement with the predicted slopes for the variable order of one and time integration order for two. Benchmarking was done for both 1D and 2D simulations with results compared to previous work performed on LSODI. These works included 1D and 2D simulations for a range of plasma parameters (densities, temperatures, voltage, etc.). For the 1D cases, Zapdos and LSODI results were in very good agreement. In the 2D cases, variable behaviors matched, with slight discrepancies in peak values. Zapdos was validated using electron and metastable density for the GEC reference cell. At high pressures, Zapdos matches trend of the experimental data set and starts to diverge at low pressure. This diverge is not unexpected due to the limitations of the models being used.

For future work, development on Zapdos will focus on decrease the limitations of the models. This includes more robust boundary conditions, such as calculation a self biasing voltage, and extending the fluid approximation to better capture particle-to-particle effects. There are plans to increase the number of current MOOSE applications that are natively coupled within Zapdos, such as the MOOSE's Navier-Stokes modules for background gas flow [10]. To expand beyond the electrostatic approximation currently used in Zapdos, there is current work to couple an electromagnetic solver to Zapdos [41]. This will help open the way for more applications, such microwave plasmas and magnetized plasmas.



(a) Power Electrode at 75 V_{pp}



(b) Power Electrode at 150 V_{pp}

Figure 12: Metastable Density at 100 mTorr

5. Acknowledgements

The authors of this work would like to acknowledge funding provided by the Nuclear Energy University Program (NEUP) Fellowship and NSF program for Software Infrastructure for Sustained Innovation (NSF-SI^2) under award 1740300. This research also made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

References

[1] Dimitris P. Lymberopoulos and Demetre J. Economou, Fluid simulations of glow discharges: Effect of metastable atoms in argon, J. Appl. Phys. **73** (8) (1993). doi:10.1063/1.352926.

[2] Dimitris P. Lymberopoulos and Demetre J. Economou, Fluid simulations of radio frequency glow discharges: Two-dimensional argon discharge including metastables, Appl. Phys. Lett. **63** (18) (1993). doi:10.1063/1.110454.

[3] Dimitris P. Lymberopoulos and Demetre J. Economou, Modeling and simulation of glow discharge plasma reactors, J. Vac. Sci. Technol. A **12** (4) (1994). doi:10.1116/1.579300.

[4] K. E. Greenberg and G. A. Hebner, Electron and metastable densities in parallel-plate radio-frequency discharges, Journal of Applied Physics **73** (1993) 8126 (1993). doi:10.1063/1.353451.

[5] Brian K. McMillin and Michael R. Zachariah, Two-dimensional argon metastable density measurements in a radio frequency plasma reactor by planar laser-induced fluorescence imaging, Journal of Applied Physics **77** (1995) 5538 (1995). doi:10.1063/1.359193.

[6] P. C. *et al.*, PlasmaPy (2021). doi:10.5281/zenodo.4602818.
URL https://github.com/PlasmaPy

[7] QuickPIC-OpenSource.
URL https://github.com/UCLA-Plasma-Simulation-Group/QuickPIC-OpenSource

[8] A. Venkattraman and A. Verma, plasmaFoam: An OpenFOAM framework for computational plasma physics and chemistry, APS Gaseous Electronics Conference (abstract id. DT2.007) (2016).

[9] Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, and *et al.*, MOOSE: Enabling massively parallel multiphysics simulation, SoftwareX **11** (2020). doi:10.1016/j.softx.2020.100430.

[10] John W. Peterson, Alexander D. Lindsay, Fande Kong, Overview of the incompressible navier–stokes simulation capabilities in the MOOSE framework, Advances in Engineering Software **119** (2018) 68–92 (2018). doi:10.1016/j.advengsoft.2018.02.004.

[11] Derek R. Gaston, Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andrš, Yaqi Wang and *et al.*, Physics-based multiscale coupling for full core nuclear reactor simulation, Annals of Nuclear Energy 84 (2015) 45–54 (2015). doi:10.1016/j.anucene.2014.09.060.

[12] Richard L. Williamson, Jason D. Hales, Stephen R. Novascone, Giovanni Pastore, Kyle A. Gamble, Benjamin W. Spencer, and *et al.*, BISON: A flexible code for advanced simulation of the performance of multiple nuclear fuel formss, Nuclear Technology **207** (7) (2021) 954–980 (2021). doi:10.1080/00295450.2020.1836940.

[13] Shane Keniley and Davide Curreli, CRANE: A MOOSE-based open source tool for plasma chemistry applications, Plasma Physics (2019). doi:arXiv:1905.10004.

[14] National Academies of Sciences, Engineering, and Medicine, Plasma Science: Enabling Technology, Sustainability, Security, and Exploration, The National Academies Press, Washington, DC, 2021 (2021). doi:10.17226/25802.
URL https://www.nap.edu/catalog/25802/plasma-science-enabling-technology-sustainability-security-and-

[15] S. Baalrud, N. Ferraro, L. Garrison, N. Howard, C. Kuranz, J. Sarff, W. Solomon, A community plan for fusion energy and discovery plasma sciences, arXiv preprint arXiv:2011.04806 (2020).

[16] Alexander D Lindsay and David B Graves and Steven C Shannon, Fully coupled simulation of the plasma liquid interface and interfacial coefficient effects, J. Phys. D: Appl. Phys. **49** (4) (2016) 235204 (2016). doi:10.1088/0022-3727/49/23/235204.

[17] J. R. Haase, D. B. Go, Designing microscale gas discharges to enhance thermionic energy conversion, International Vacuum Nanoelectronics Conference (2017). doi:10.1109/IVNC.2017.8051595.

[18] B. D. Dudson, J. Madsen, J. Omotani, P. Hill, L. Easy, and M. Løiten, Verification of BOUT++ by the method of man-

ufactured solutions, Phys. Plasmas **23** (2016) 062303 (2016). doi:10.1063/1.4953429.

[19] Rémi Delaporte-Mathurin and Etienne A. Hodille and Jonathan Mougenot and Yann Charles and Christian Grisolia, Finite element analysis of hydrogen retention in ITER plasma facing components using FESTIM, Nuclear Materials and Energy **21** (2019) 100709 (2019). doi:10.1016/j.nme.2019.100709.

[20] Fish, Jacob and Belytschko, Ted, A first course in finite elements, Wiley, 2007 (2007).

[21] Miles M. Turner, Computer simulation in low-temperature plasma physics: Future challenges, Plasma Process Polym **14** (2017) 1600121 (2017). doi:10.1002/ppap.201600121.

[22] A. Fierro, E. Barnat, M. Hopkins, C. Moore, G. Radtke, and B. Yee, Challenges and opportunities in verification and validation of low temperature plasma simulations and experiments, Eur. Phys. J. D **75** (2021) 151 (2021). doi:10.1140/epjd/s10053-021-00088-6.

[23] Martin Greenwald, Verification and validation for magnetic fusion, Physics of Plasmas **17** (2010) 058101 (2010). doi:10.1063/1.3298884.

[24] L. E. Schwer, An overview of the PTC 60/V&V 10: guide for verification and validation in computational solid mechanics, Engineering with Computers **23** (2007) 245–252 (2007). doi:10.1007/s00366-007-0072-z.

[25] J. K. Olthoff and K. E. Greenberg, The gaseous electronics conference rf reference cell—an introduction, J. Res. Natl. Inst. Stand. Technol. **100** (1995) 327 (1995). doi:10.6028/jres.100.025.

[26] G J M Hagelaar and L C Pitchford, Solving the boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models, Plasma Sources Sci. Technol. **14** (722) (2005). doi:10.1088/0963-0252/14/4/011.

[27] Y Sakiyama and D B Graves, Corona-glow transition in the atmospheric pressure rf-excited plasma needle, J. Phys. D: Appl. Phys. **39** (2006) 3644 (2006). doi:10.1088/0022-3727/39/16/018.

[28] G J Hagelaar and F J De Hoog and G M Kroesen., Boundary conditions in fluid models of gas discharges, Phys. Rev. E **62** (1) (2000) 1452–4 (2000). doi:10.1103/PhysRevE.62.1452.

[29] Y Sakiyama and D B Graves., Nonthermal atmospheric rf plasma in one-dimensional spherical coordinates: asymmetric sheath structure and the discharge mechanism, J. Appl. Phys. **101** (2007) 073306 (2007). doi:10.1063/1.2715745.

[30] E. Gogolides, H. H. Sawin and R. A. Brown, Direct Calculation of Time-Periodic States of Continuum Models of Radio-Frequency Plasma, Chemical Engineering Science **47** (1992) 3839–3855 (1992). doi:10.1016/0009-2509(92)85133-V.

[31] Alexander Lindsay, Casey Icenhour, Corey DeChant, and Shane Keniley, Zapdos: An open source MOOSE framework application for the simulation of plasmas.
URL https://shannon-lab.github.io/zapdos/

[32] Dimitris P. Lymberopoulos, Multidimensional modeling and simulation of plasma processing reactors, Ph.D. thesis, University of Houston (1995).

[33] C M Ferreira and J Loureiro, Electron transport parameters and excitation rates in argon, Phys. D: Appl. Phys. **16** (1983) 1611 (1983). doi:10.1088/0022-3727/16/9/008.

[34] C M Ferreira, J Loureiro and A. Ricard, Populations in the metastable and the resonance levels of argon and stepwise ionization effects in a low-pressure argon positive column, J. Appl. Phys. **57** (82) (1985). doi:10.1063/1.335400.

[35] E V Karoulina and Yu A Lebedev, Computer simulation of microwave and DC plasmas: comparative characterisation of plasmas, Phys. D: Appl. Phys. **25** (401) (1992). doi:10.1088/0022-3727/25/3/010.

[36] A. L. Ward, Effect of space charge in cold-cathode gas discharges, Phys. Rev. **112** (1852) (1958). doi:10.1103/PhysRev.112.1852.

[37] Geoffrey R. Scheller, Richard A. Gottscho, and T. Intrator, Nonlinear excitation and dissociation kinetics in discharges through mixtures of rare and attaching gases, J. Appl. Phys. **64** (9) (1988). doi:10.1063/1.341287.

[38] I. Yu Baranov, V. I. Demidov, and N. B. Kolokolov, Temperature dependence of rate constants for metastable atomic-argon deactivation by slow electrons, Opt. Spectrosc. **51** (316) (1981).

[39] Peter L. G. Ventzek, Robert J. Hoekstra, and Mark J. Kushner, Two-dimensional modeling of high plasma density inductively coupled sources for materials processing, J. Vac. ScI. Technol. B **12** (461) (1994). doi:10.1116/1.587101.

[40] Mark J Kushner, Hybrid modelling of low temperature plasmas for fundamental investigations and equipment design, J. Phys. D: Appl. Phys. **42** (2009) 194013 (2009). doi:10.1088/0022-3727/42/19/194013.

[41] C. T. Icenhour, C. DeChant, A. D. Lindsay, R. C. Martineau, D. Green, S. Shannon, Validation studies using elk and the open-source moose framework application zapdos for electromagnetic coupled plasma simulation (10 2019).
URL https://www.osti.gov/biblio/1574011