



Numerical Simulation of Plasma Interfaces Using the Starfish Plasma Simulation Code

James Senig¹ and Xiaowen Wang²

Department of Aerospace Engineering and Mechanics
The University of Alabama, Tuscaloosa, AL 35487

A particular challenge for low temperature plasma (LTP) research is the diversity of parameter space and conditions. For plasma systems where the densities are not low, the kinetic theory is time-consuming and becomes unrealistic. In this regime, the particle-in-Cell (PIC) method is appropriate where the evolution of a particle system at every time step consists of an Eulerian stage and a Lagrangian stage. The PIC method can deal with complex geometries and large distortions in the field. The PIC solver, Starfish, is a two-dimensional plasma and gas simulation code operating on structured 2D/axisymmetric Cartesian or body fitted stretched meshes. The purpose of this study is to use the Starfish Plasma Simulation Code for numerical simulations of plasma interfaces. Specifically, two applications are considered: 1) the modeling of a large area (30 cm x 30 cm) microwave plasma chemical vapor deposition system, and 2) the understanding of LTP treatment on surface modification of polycaprolactone pellets and thermal properties of extruded filaments. With the exact geometries and experimental results being provided, numerical simulations of these two applications are ongoing.

I. Introduction

A particular challenge for low temperature plasma (LTP) research is the diversity of parameter space and conditions. For low-density plasma, the kinetic theory is a more refined treatment [1-2]. Distribution functions give the complete solutions when each energy mode (translation/rotation/vibration/electron) is represented by a unique distribution function. One does not need models of mass, momentum and energy transports. However, it is very difficult to solve all collision terms. For plasma systems where the densities are not low, the kinetic theory is time-consuming and becomes unrealistic.

Fortunately, continuum models used in fluid mechanics can explain most of real plasma phenomena and achieve high-order accuracy. Kundrapu et al. [3] developed a multi-physics framework for plasma characterization in the reentry flows and analysis of blackout mitigation devices. The framework was based on multi-fluid MHD equations. Franck et al. [4] explored the effectiveness of plasma actuator for the controls of lift and trailing edge separation on wind turbine blades. In their decoupled simulations, the plasma actuator was modeled as a body force. Croft & Moeller [5] implemented a finite element solver for a two-fluid plasma utilizing the purely Maxwell's equations. However, the coupling of fluid dynamics and electromagnetic dynamics was only for inviscid flow and perfect gas. Chen et al. [6] studied a two-dimensional plasma based on fluid modeling. Poisson equation was used for electrostatic potential. Their calculations of electrons, ions, and neutral species employed significant simplifications. In summary, previous calculations are short of self-consistent thermochemical models for multi-species plasma. On the other hand, numerical methods for ionized hypersonic flows generally have complete thermochemical models of mass, momentum and energy transports. Wang & Zhong [7] developed a high-order solver for hypersonic non-equilibrium flows. To capture the multi-physics feature, the solver employed a two-temperature model. However, the solver could not directly apply to plasma simulations because it assumes no charge separation happens in the field, an apparently unreasonable assumption applied in most

¹ Graduate Student, AIAA Student Member, Email: jdsenig@crimson.ua.edu

² Assistant Professor, AIAA Associate Fellow, Email: xwang@eng.ua.edu

hypersonic flow solvers for simplification. Recently, Blanco and Josyula [8] developed a numerical model of weakly ionized hypersonic flows. They coupled the Poisson equation to the Navier-Stokes equations for electric field. Nevertheless, it neglected magnetic field generated by moving charges.

In this regime, the PIC method is appropriate where the evolution of a particle system at every time step consists of two stages [9]. At the first stage, under fixed states of particles their interactions and joint effects on the medium are calculated (Eulerian stage). At the next stage, states of particles are updated for the next time step (Lagrangian stage). The PIC method can deal with complex geometries and large distortions in the field. The PIC solver, Starfish, is a two-dimensional plasma and gas simulation code operating on structured 2D/axisymmetric Cartesian or body fitted stretched meshes [10]. Surface geometry is included via linear or cubic splines. The current open-source version mainly implements functionality needed for simulations of low-density plasmas using the PIC method with MCC/DSMC collisions. It supports four types of inter-material interactions: material and a surface boundary, two kinetic materials, a kinetic source and a fluid target, and two fluid materials. Poisson's equation is used to obtain plasma potential, which is in turn used to compute the electric field. The PIC method makes Starfish convenient to simulate plasma interfaces, although modeling and implementation of new material properties, interface profiles, and electromagnetic dynamics are still needed.

In this paper, numerical simulations of plasma interfaces are conducted using the Starfish plasma simulation code. Specifically, two applications will be considered: 1) the modeling of a large area (30 cm x 30 cm) microwave plasma chemical vapor deposition system, and 2) the understanding of LTP treatment on surface modification of polycaprolactone pellets and thermal properties of extruded filaments.

II. Numerical Methods and Governing Equations

When solving any fluid flow problem, there are two different viewpoints one can use; the Eulerian and Lagrangian viewpoints. These viewpoints are used to construct the partial differential equations that govern the flow. The Eulerian viewpoint describes the flow at a certain point in space and time and the coordinate system is fixed relative to the observer. This means that every flow characteristic in the flow field is determined by its location and time. For example, the velocity and acceleration are denoted as below.

$$\vec{u} = \vec{u}(x, y, z, t) \quad (1)$$

$$\vec{a} = \vec{a}(x, y, z, t) \quad (2)$$

The change in the velocity with respect to time is given below as a substantial derivative. Two different types of acceleration are present in equation 3, the local acceleration and convective terms. The first term is the local acceleration and shows the dependence on time as described in the Eulerian viewpoint. The remaining three terms are the convective terms and they describe the change of the motion of the particles from areas of lower or higher velocity. Equation 3 can also be written in vector notation as given in equation 4. In the Lagrangian viewpoint, the coordinate system is fixed to each fluid particle moving in the flow field. There are some advantages to using the Eulerian frame of reference over the Lagrangian frame of reference. The Eulerian view can deal with distortions in the fluid as well as an imperfect no-slip condition. There are also some disadvantages such as false diffusion when the fluid interacts with a material in the flow field. The mesh created when using the Eulerian view is quite different from that of the Lagrangian view. Each mesh is comprised of cells which retain constant values such as density, velocity, etc. with the fluid moving through the mesh.

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \quad (3)$$

As mentioned previously, the Lagrangian viewpoint is not equipped to deal with large distortions in the field. The mesh for the Lagrangian viewpoint is comprised of cells that deform according to the deformation of the fluid.

The PIC method in essence is a combination of the Eulerian and Lagrangian viewpoints. This combination eliminates some of the disadvantages the previously mentioned methods have and retain some of their advantages. The disadvantage of not being able to deal with large distortions in the fluid flow that

the Lagrangian method gives is gone. The inclusion of the Lagrangian method does however allow the user to track the fluid properties in each cell, which will be shown in the next section. Since the PIC method is a combination of both of these methods, it will contain an Eulerian and Lagrangian computing mesh. The Eulerian mesh is the mesh that the fluid will flow through and is fixed relative to a laboratory setting. The fluid itself is grouped into particles that contain enough actual particles of the fluid to still be considered a continuum. This grouping of particles creates the Lagrangian mesh and these particles will move throughout the aforementioned Eulerian mesh. According to Harlow [9], in the Lagrangian mesh the masses and positions are tracked and in the Eulerian mesh the velocity, masses of each material, and internal energy are recorded.

Starfish uses the PIC method to solve for plasma flow fields as well as the interaction of ions with its surrounding environment. Starfish utilizes several input files to run each simulation. These files contain items such as mesh types, particle information, material interactions, and boundaries to name a few. Each of these files are described in detail in the following subsections.

A. starfish.xml

Every simulation will contain the “starfish.xml” file. This file contains every command that the program will follow in any given simulation. The tagline **<simulation>** is set as the parent class within the JAVA program and every command below it reaches into other xml files to set up each simulation. Aside from these other files, the type of solver used and the time parameters to be used are also set in this file. After the simulation is over, this file sends all the results to corresponding TecPlot files which can then be viewed in TecPlot 360.

B. domain.xml

The first input file loaded in starfish.xml sets up the flow domain for the simulation being run. Starfish is equipped to set up elliptic meshes as well as rectilinear meshes. The size of the mesh and the number of nodes is set in this file. The final requirement for any simulation is the boundary condition and their type (Dirichlet, Neumann, etc.).

C. materials.xml

This file contains the description for every type of material present in a given simulation. These materials range from the types of surfaces that can be encountered and the types of ions in the simulation. Each ion in the simulation is given a type, the main one being “kinetic”, which means its presence in the flow field is calculated using the PIC method. The molecular weight and charge among other aspects of each ion is also given in this file. As mentioned previously, the materials the ions can potentially interact with are also described.

D. boundaries.xml

In each flow field, Starfish allows the user to input different types of boundaries. These boundaries can vary from solid objects that the plasma must flow around to injection sites for different types of particles. These solid objects are represented by either linear or cubic Bezier splines. These splines are a collection of points that can accurately define the object. Along with these points the type is also assigned based on what the object represents in the flow field. This could be an inlet for the injection of particles or an actual solid body. When constructing a solid body, each point is considered a node, and the order for these nodes matters to the code. The inlets can be described by many fewer points and they are denoted as “virtual” under the type category and is not taken into account when constructing the mesh.

E. interactions.xml

In more complex simulations, collisions between kinetic, solid, and fluid materials can be considered. Interactions are only considered between two items at a time, one being regarded as the “surface_hit source” and the other the “target.” The result of such an interaction gives a product which is also an input in the interactions file. Starfish supports several types of interaction models. Ions are given three choices when

they encounter a solid surface. They can transform into a neutral atom and reflect at the angle that they came in at while maintaining the same momentum. They can also be re-emitted from the solid surface following the cosine law. The cosine law treats the particles as if they land on the solid surface and then are re-emitted after a short time. Finally, the particle can be “absorbed” by the solid surface, taking it out of the flow field. Each of these interactions can be implemented at the same time in one file given likelihoods for occurrence. Ions can also collide with neutral particles and switch charges. Chemical reactions such as ionization can also be considered in Starfish.

Equations 5-8 are the partial differential equations used in computational fluid dynamics problems. Equation 5, the continuity equation, is written down in the conservative form allowing the density to vary throughout the flow field. Equations 6 and 7 are the Navier-Stokes equations ignoring the viscous terms, the gravitational force, and any external body forces. Following equations 5-8 are the steps taken to compute values such as the velocity, mass, positions and internal energy of each material present in the simulation as given by Harlow [9].

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \quad (5)$$

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} \quad (6)$$

$$\rho \frac{\partial v}{\partial t} + \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} \quad (7)$$

$$\rho \frac{\partial e}{\partial t} + \rho u \frac{\partial e}{\partial x} + \rho v \frac{\partial e}{\partial y} = -P \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \quad (8)$$

Before proceeding to solving for the pressures, velocities, and densities of each cell, the notation for these values must be discussed. The meshes used in this project so far have remained rectilinear. Each cell is labeled (i^j) and each value such as pressure will be given as p_i^j . The following formulation will have

some values listed as $(i + \frac{1}{2})^j$, representing the average value between cells (i^j) and $(i + 1)^j$. Equations 6 and 7 are written in a finite difference form to help solve for pressure first. At this point in the calculation, the fluid is considered stationary so the second and third terms, known as the transport terms, are dropped. This gives equations 9 and 10 which can be reduced on the right-hand side to be the average of the pressures on each side of the cell being examined. Δx and Δy are the distances between each cell.

$$\rho_i^j \left(\frac{\partial u}{\partial t} \right)_i^j = -\frac{1}{\Delta x} \left(P_{i+\frac{1}{2}}^j - P_{i-\frac{1}{2}}^j \right) \quad (9)$$

$$\rho_i^j \left(\frac{\partial v}{\partial t} \right)_i^j = -\frac{1}{\Delta y} \left(P_{i+\frac{1}{2}}^{j+\frac{1}{2}} - P_{i-\frac{1}{2}}^{j-\frac{1}{2}} \right) \quad (10)$$

The same procedure can be done with the energy equation given in equation 11. A new term, q , was introduced into the energy equation as the artificial viscosity which helps with stability in the solution.

$$\rho_i^j \left(\frac{\partial e}{\partial t} \right)_i^j = -P_i^j \left(\frac{u_{i+\frac{1}{2}}^j - u_{i-\frac{1}{2}}^j}{\Delta x} - \frac{v_i^{j+\frac{1}{2}} - v_i^{j-\frac{1}{2}}}{\Delta y} \right) - \frac{(qu)_{i+\frac{1}{2}}^j - (qu)_{i-\frac{1}{2}}^j}{\Delta x} - \frac{(qv)_{i+\frac{1}{2}}^{j+\frac{1}{2}} - (qv)_{i-\frac{1}{2}}^{j-\frac{1}{2}}}{\Delta y} + u_i^j \left(\frac{q_{i+\frac{1}{2}}^j - q_{i-\frac{1}{2}}^j}{\Delta x} \right) + v_i^j \left(\frac{q_{i+\frac{1}{2}}^{j+\frac{1}{2}} - q_{i-\frac{1}{2}}^{j-\frac{1}{2}}}{\Delta y} \right) \quad (11)$$

After the charge density, equation 12, and pressure are computed, Starfish uses Poisson's equation, equation 13, to solve for the plasma potential. With the plasma potential, the electric field can be found by taking the negative Laplace of the solved potential as seen in equation 14. Finally, the particle velocities can be found using the Lorentz force as seen in equation 15. Equations 16-19, as shown by Brieda [11],

show how the Boris scheme is derived to eliminate the electric field from the Lorentz equation, allowing the velocity to be solved for.

$$\rho = \sum_s q_s n_s \quad (12)$$

$$\epsilon_0 \nabla^2 \phi = -\rho \quad (13)$$

$$\vec{E} = -\nabla \phi \quad (14)$$

$$\vec{F} = m \frac{d\vec{v}}{dt} = q(\vec{E} + \vec{v} \times \vec{B}) \quad (15)$$

$$\frac{\vec{v}^{n+\frac{1}{2}} - \vec{v}^{n-\frac{1}{2}}}{\Delta t} = \frac{q}{m} \left(\vec{E} + \frac{\vec{v}^{n+\frac{1}{2}} + \vec{v}^{n-\frac{1}{2}}}{2} \times \vec{B} \right) \quad (16)$$

$$\vec{v}^{n-\frac{1}{2}} = \vec{v}^- - \frac{q\vec{E}}{m} \frac{\Delta t}{2} \quad (17)$$

$$\vec{v}^{n+\frac{1}{2}} = \vec{v}^+ + \frac{q\vec{E}}{m} \frac{\Delta t}{2} \quad (18)$$

$$\frac{\vec{v}^+ - \vec{v}^-}{\Delta t} = \frac{q}{2m} (\vec{v}^+ + \vec{v}^-) \times \vec{B} \quad (19)$$

III. Results of Code Test

To test the open source Starfish code and get familiar with it, plasma interface over an ellipse has been simulated. Details of the simulation are as follows.

A. Boundaries and Domain

Our goal was to establish proper implementation of Starfish to describe the fluid flow around a cylinder with an ellipsoidal face. The first task in this simulation was to construct the solid object that the flow would go around. This was done in Mathematica by creating a function to calculate the corresponding y-values for each x-value in the ellipse domain. As mentioned previously, the code Starfish runs on is concerned with the order of the points. The list of points generated by Mathematica was generated in clockwise order starting at (0.1, 0). Through trial and error with the uploaded boundary file, it was found that near the end points on the x-axis there needed to be more points. When boundaries with an even distribution of points were uploaded, the simulation produced irregularities along the boundary. This was due to the sharp slopes of the splines from the end points to the next point. To mediate this problem without causing the solver to take too much longer to converge, the frequency of the points generated around the ellipse was based on their respective locations. The formula used for the ellipse can be seen below along with the boundary loaded in Tecplot. An inlet spanning the left side of the domain was also inserted in the boundaries file.

$$\frac{x^2}{0.1} + \frac{y^2}{0.025^2} = 1$$

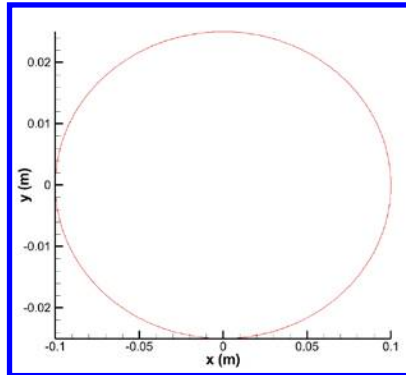


Figure 1: Boundary in flow field

The mesh for this simulation was started from an origin at $(-0.15, 0)$ and was given a uniform space of 0.005 for 81, 41 nodes.

B. Materials, Interactions, and Sources

Five different materials were present in this simulation. They were the solid boundary described above, O^+ , O , e^- , and the inserted O^+ . These materials, listed in the materials.xml file, are given their respective molecular weights, charges, and types. The inlet created for this simulation was set to up to introduce O^+ ions into the flow field. Several complex interactions were implemented for this simulation. The first interaction considered was the collision of the inserted O^+ ions and the solid steel boundary in the fluid flow. The product of this interaction was a neutral O atom and it was given the cosine model to follow along with a 90% likelihood of occurring. The other 10% is taking the O^+ ions out of the simulation. The ionization of the O atoms is also considered, the product being an O^+ ion and an electron. Finally, the collision between O^+ and O atoms was treated as them switching their charges.

C. Results

Figure 2 shows the plasma potential throughout the flow field as solved for by Poisson's equation. It follows the general trend expected from a uniform flow of ions over a structure. The potential was set to 0 V at the boundary and 100 V on the solid surface, and the figure agrees to these conditions. Figure 3 shows the plasma density throughout the fluid flow, which is consistent with known flows.

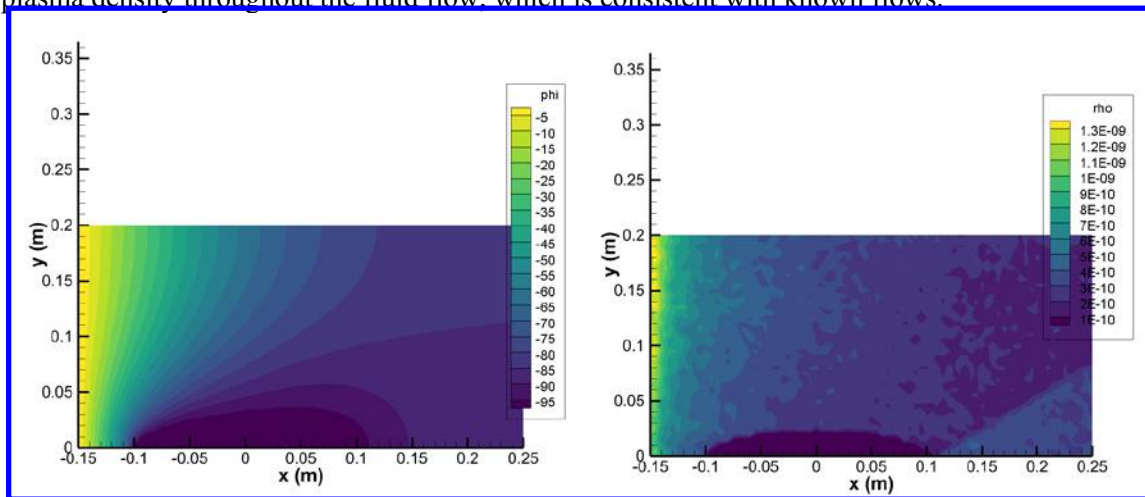


Figure 2: Plasma Potential

Figure 3: Density of plasma

Figures 4, 5, and 6 show the average number densities of each type of particle in the fluid flow. As expected, the highest concentration of the inserted O^+ particles are at the inlet and the concentration dissipates as some of those particles are converted to neutral O particles or absorbed by the solid cylinder. The number density for the neutral O particles in figure 5 resulted as expected. The O atoms were produced by two different mechanisms: charge switches between O^+ ions and O particles and the impact between an O^+ ion and the solid cylinder. Considering this, the density being higher near the boundary is correct. Figure 6 shows the number density for the O^+ particles resulting from the ionization of the neutral O particles, which were more prevalent near the side of the boundary closest to the inlet.

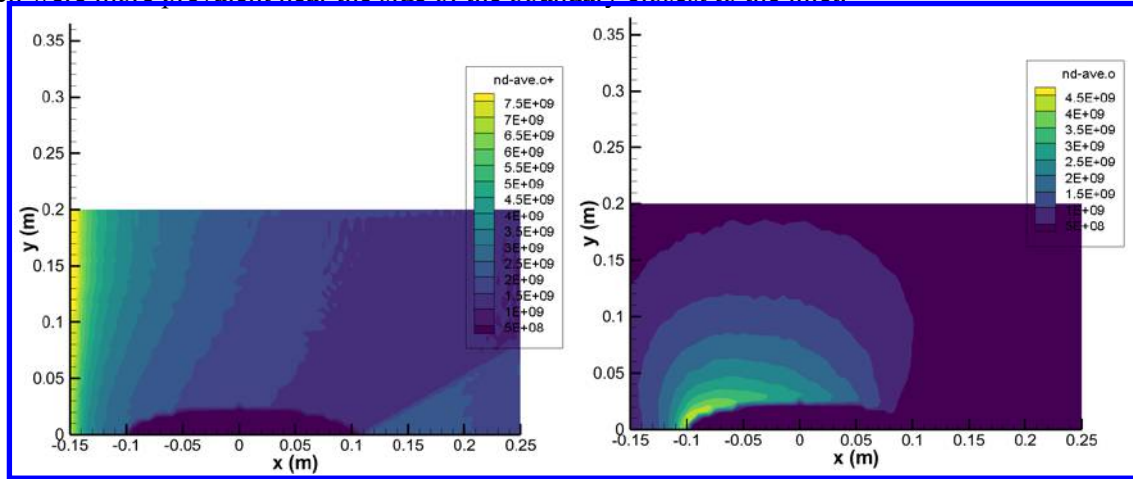


Figure 4: Number density of inserted O^+

Figure 5: Number density of neutral O

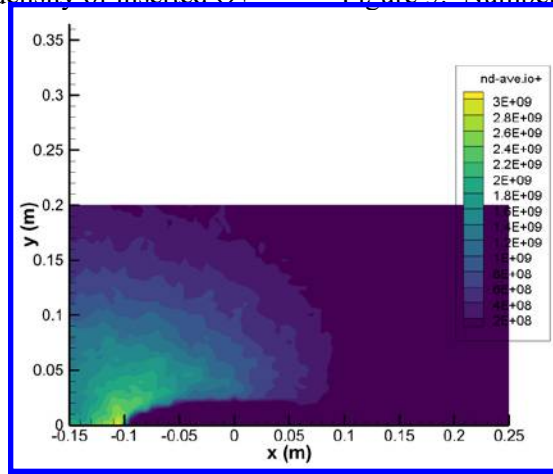


Figure 6: Number density of ionized O

IV. Simulation of A Large Area Microwave Plasma Chemical Vapor Deposition System

Many current hypersonic challenges are due to the limitations of materials degradation under extreme conditions (i.e. thermal, electrical, magnetic, acoustic, shear, or pressure fields in hypersonic conditions). A UAB research group focuses on the synthesis of high temperature super-hard coatings for applications in hypersonic vehicle. For example, Baker et al. [12] synthesized a novel series of boron-carbon materials using a microwave plasma chemical vapor deposition technique employing hydrogen/methane/diborane gas-phase chemistry. Depending on the growth conditions, the hardness of these composite materials can be tailored from 8 GPa to as high as 62 GPa. Hence, the modeling of the microwave plasma chemical vapor deposition system is of critical importance to tailor coatings for specific applications. In addition, hypersonic cruise vehicles are normally designed with predominantly two-dimensional shapes and

relatively sharp leading edges. When a hypersonic flow passes such a configuration, surface porous coating is considered one of the most promising transition control techniques. It has a minimal effect on the mean flow but can greatly stabilize the boundary layer. Figure 7 shows schematics of surface porous coating. The pore size is generally on micron scale so that it is expected to have a minimal effect on the properties of boron-carbon materials.

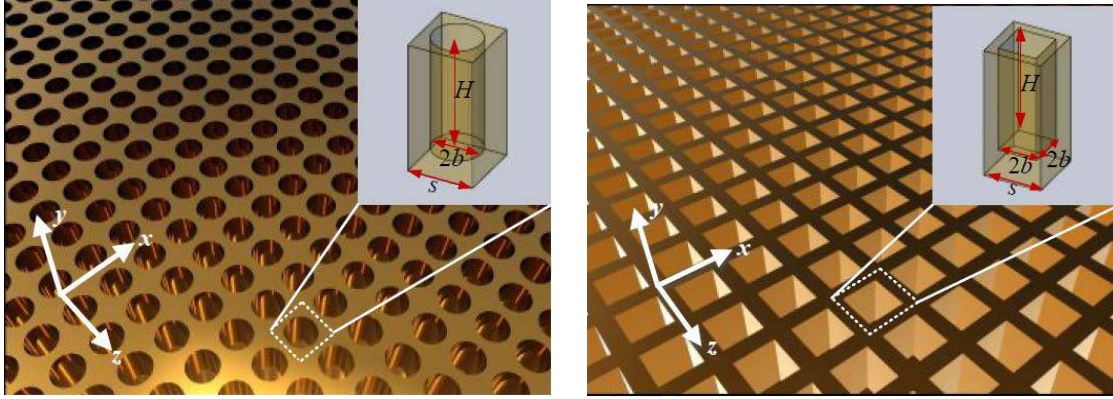


Figure 7: Schematics of surface porous coating for hypersonic vehicle applications (Pore size on micro scale).

The design and synthesis of high temperature super-hard porous coatings can not only protect the hypersonic vehicles but also control boundary-layer transition that can lead to less drag, less surface heating, and less fuel consumption. For this problem, the first step is to implement microwave plasma conditions and hydrogen/methane/diborane gas-phase chemistry into the PIC solver, Starfish. Once these implementations are verified, the second step is to conduct parametric studies on different synthesis conditions and porous coating efficiency in delay boundary-layer transition.

Two different solvers were used to simulate the plasma flow field. The first solver used is the “QN” solver. The QN solver uses the quasi-neutral Boltzmann relationship to calculate the plasma potential. This can be seen below in equation 20, where ϕ_0 , $T_{e,0}$, and n_0 are the reference values for potential, temperature, and number density respectively. These values are provided in the starfish.xml and materials.xml files. This form of the Boltzmann relation relates the local potential ϕ to the local electron density n .

$$\phi = \phi_0 + kT_{e,0} \ln\left(\frac{n}{n_0}\right) \quad (20)$$

The second solver used in the simulation is the non-linear Poisson solver. This solver solves the Poisson equation, as shown in equation 21, where the charge density has been replaced by the ion and electron densities. The Boltzmann relation is inverted to a non-linear mode to solve for the simulation electrons. This can be seen in equation 22. Equation 22 is then substituted into equation 21 to create the non-linear Poisson equation which is discretized using a central differencing method.

$$\epsilon_0 \nabla^2 \phi = -e(Z_i n_i - n_e) \quad (21)$$

$$n_e = n_0 \exp\left(\frac{\phi - \phi_0}{kT_{e,0}}\right) \quad (22)$$

$$\nabla^2 \phi = -\frac{e}{\epsilon_0} \left(Z_i n_i - n_0 \exp\left(\frac{\phi - \phi_0}{kT_{e,0}}\right) \right) \quad (23)$$

A. Boundaries and Domain

The boundaries for this system can be seen below. The “large area” in this system is modeled using a stainless-steel plate. The steel plate was placed 0.2 m into the flow field. The steel plate was constructed

using two different types of boundaries, a reactive boundary and a non-reactive boundary. This was done to prevent any gases from reacting with the other side of the plate. The other boundary created for this simulation is the injection site for the incoming particles and runs along the left-hand side of Figure 8. A Dirichlet boundary condition was used for the boundary of the plasma flow field and the surface of the stainless-steel plate, where the plasma potential was set to 0 V.

As described previously, the domain file is responsible for setting the size and type of the simulation area and is where the mesh is prescribed. The domain for this simulation was set to 0.8 m x 0.4 m and is set to the XY domain. Three separate meshes were created across the domain to be used during computation. A finer mesh was created surrounding the plate to account for the increased number of particle interactions. These interactions will be described later. The mesh size in the x-direction for this finer mesh is 0.00125 m and 0.005 m for the y-direction. The other two meshes surrounding the finer mesh are the same size and have a cell size of 0.005 m x 0.005 m. This mesh can be seen in Figure 9.

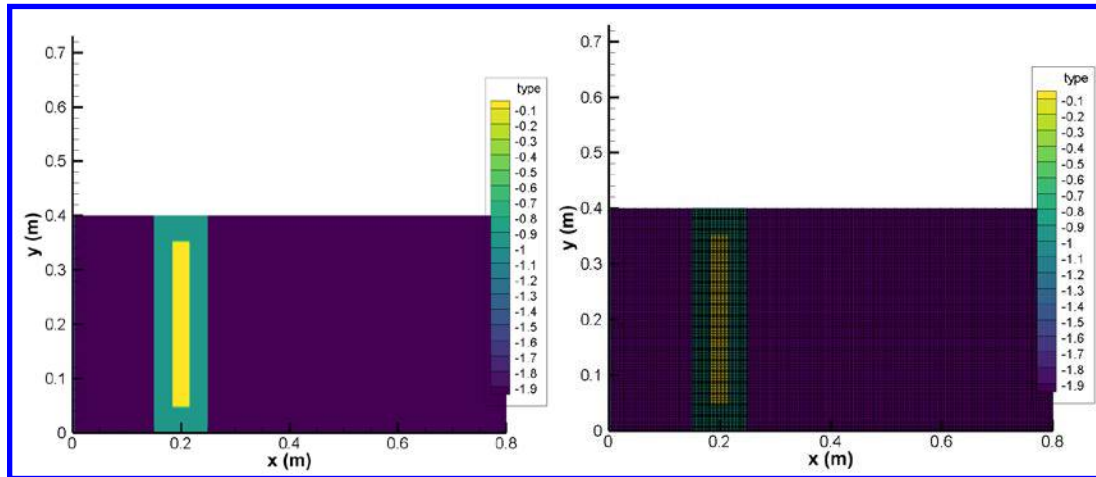


Figure 8: Boundaries

Figure 9: Mesh for Simulation

B. Materials, Interactions, and Sources

The first two materials implemented in this simulation are of the “solid” type. Solid materials are used to construct obstacles in the flow field. The obstacle in this simulation is the stainless-steel plate with the substrate placed on it. This plate was constructed using two different types of materials. As mentioned previously, only one side of the plate was designed to react with incoming particles. Both materials are given the same density and molecular weight.

The injected particles for this simulation were chosen to be O^+ ions. The UAB project is using other reacting gases but for preliminary work only the O^+ particles are being considered. The reacting gases are classified as the “kinetic” type. Brieda [11] models kinetic materials with Direct Simulation Monte Carlo or the PIC method. General information on the O^+ ion such as molecular weight (16) and charge (1) are specified here. Some other information such as molecular diameter, viscosity index, and reference temperature were pulled from Bird [14]. Lastly, each simulation particle was assigned several real particles, denoted as the specific weight. In this simulation, each simulation particle has 2500 actual particles. This value was chosen based on simulation examples given by Brieda [11].

Two more kinetic materials are included in this preliminary work, neutral Oxygen atoms and ionized Oxygen atoms. The neutral Oxygen atoms have the same molecular weight as the O^+ ions, however the charge is 0 and the specific weight was set to 5000 real particles per simulation particle. The difference in specific weights means that for every O^+ particle that reacts in the flow field, two neutral Oxygen atoms are created. The ionized Oxygen atom differs from the injected O^+ particles because they are created in the flow field due to high temperatures. Their molecular weight and charge are the same as the O^+ ions, but the specific weight was set to 50 real particles per simulation particle. The last material included in the simulation is the Boltzmann electron.

Several interactions were used in the development of this simulation. The first interaction considered is a surface interaction between the injected O^+ ions and the stainless-steel plate. According to Brieda [11], a smooth surface will still contain several irregularities at the atomic scale, which can in turn alter the deflection path for particles that encounter it. This was considered using two separate interaction models inside Starfish: Lambert's Cosine model and the specular model. Each were given a probability of 0.3 of happening. When the O^+ ion reaches the plate, it will create two neutral Oxygen atoms, which follow the trajectories assigned to them. The remaining particles are absorbed by the plate and not reemitted.

As mentioned previously, the plate was constructed so that only the frontal portion reacts with the O^+ ions. That is taken care of in this file. The O^+ ions are only permitted to create neutral atoms with the frontal portion of the plate, denoted in the materials file as SS (solid steel). The back portion of the plate is denoted as NR (non-reactive). The only interactions between the O/O^+ atoms and this side of the plate are diffuse reflections, where the corresponding atom remains unchanged.

The final interaction included in this simulation is the ionization of the neutral Oxygen atoms produced in the previous interactions. This ionization will typically occur when the temperature of the plasma reaches approximately 3000 K. Two materials are involved in this interaction: the Boltzmann electron and the neutral Oxygen atoms. The following equation shows the chemical reaction taking place. The associated rates for this chemical reaction are included in the file as well.



The injected O^+ ions were sent into the flow field with a mass flow rate according to equation 25. In equation 25, m is the mass of the particle, n is the number density, U is the velocity of the particle, and A is the area of the injection site. The particles being injected, the O^+ ions, have a mass of 16 amu, or 2.66×10^{-26} kg. The number density chosen per Brieda [11] was $1e12 \text{ m}^{-3}$, representing the average plasma density at low earth orbit. The velocity chosen for this simulation was set to 50 m/s and the area is 0.4 m^2 assuming unit length depth. These values are used to confirm that Starfish is capable of handling large area plasma flow fields. Substituting these values in equation 25, the mass flow rate is $5.3 \times 10^{-13} \text{ kg/s}$.

$$\dot{m} = mnUA \quad (25)$$

C. QN Solver Results

The following results were obtained using the QN solver. Figure 10 shows the potential in the flow field found using this solver. The largest potential occurs immediately behind the substrate plate. This is consistent with figure 12, which shows where the density of O^+ is smallest at the same location. This will create a larger potential value. As with the test case discussed previously, the density of each particle present in the flow field matches what it expected. Figure 12 shows the density of the inserted O^+ ions. As expected, these ions are more highly concentrated near the insertion area and on the edges of the plate. The density increases near the edges of the plate due to the reduced particles velocities in this area.

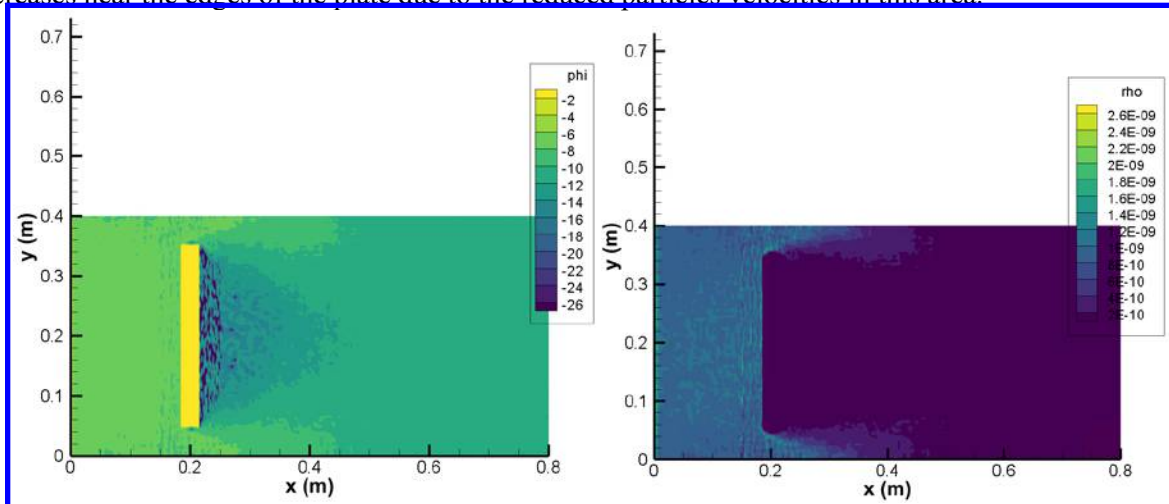
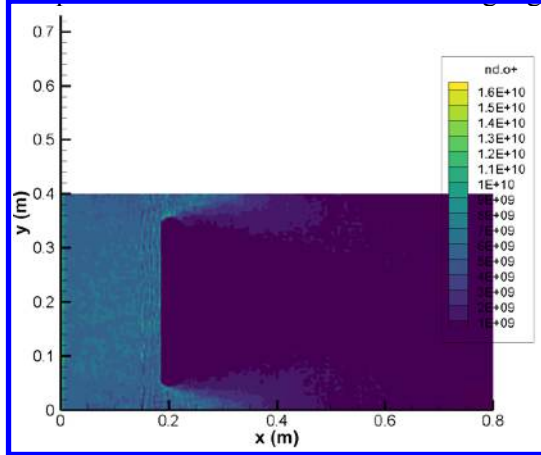
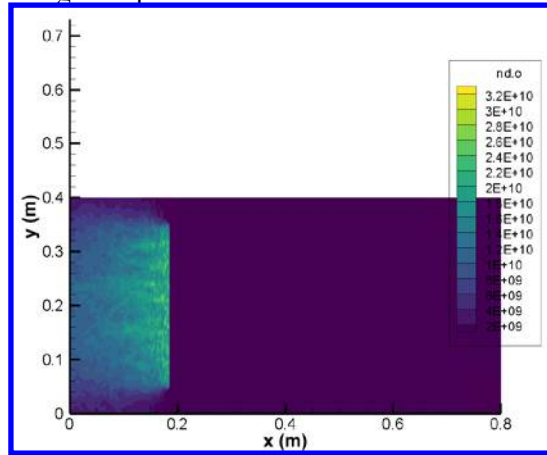
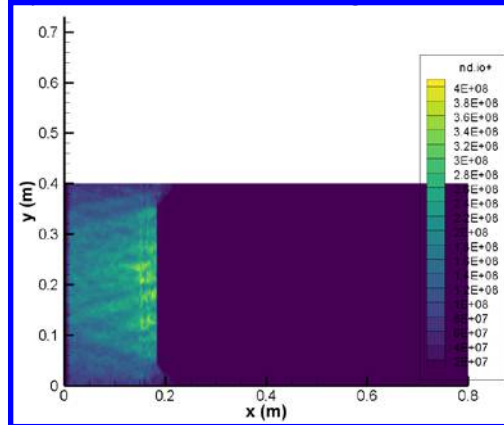
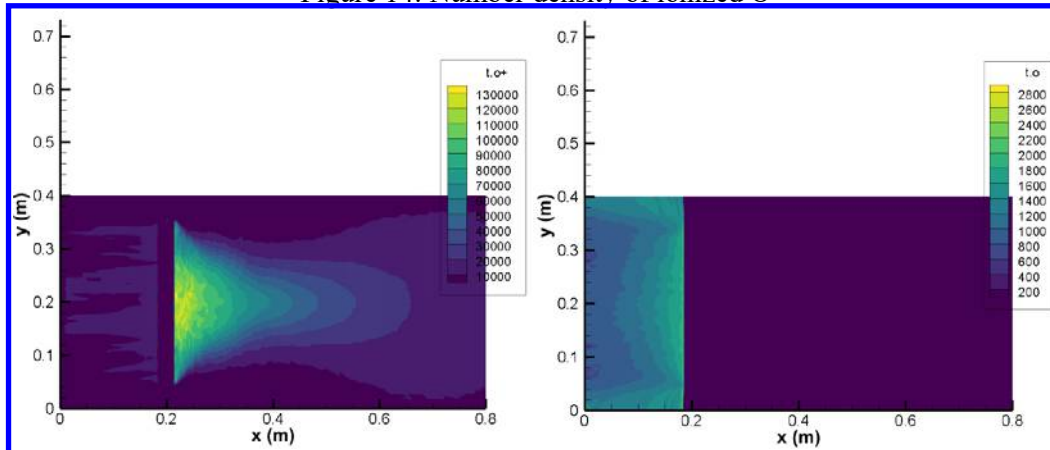
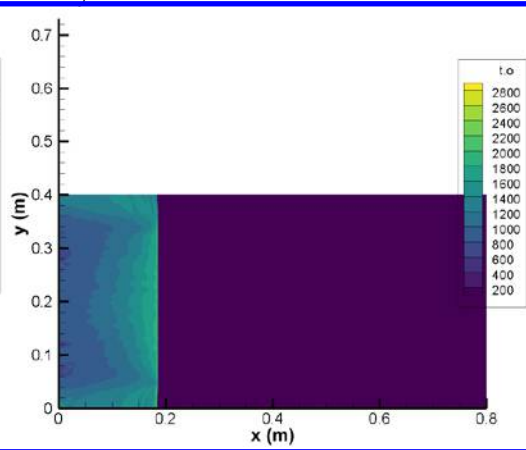


Figure 10: Plasma Potential

Figure 11: Density of Plasma

The density of the neutral Oxygen atoms shown in figure 13, as with the test case, is highest near the plate and decreases towards the insertion point. This is due to the interactions implemented in the simulation. Any neutral Oxygen atoms produced in the flow field are a result of an inserted O^+ ion encountering the plate. The models implemented do not allow them to curl around the plate and there is no implementation for particle-particle interactions since the densities are so low. Figure 14 shows the density of the ionized Oxygen particles. As with the neutral Oxygen particles, the ionized particles could physically only appear in front of the plate due to the ionization interaction. It should also be noted that the density of the ionized Oxygen is much lower than the other particles present in the simulation. This can be attributed to the temperature in the flow field not reaching high enough temperatures until further into the simulation.

Figure 12: Number density of inserted O^+ Figure 13: Number density of neutral O Figure 14: Number density of ionized O Figure 15: Temperature of O^+ Figure 16: Temperature of neutral O

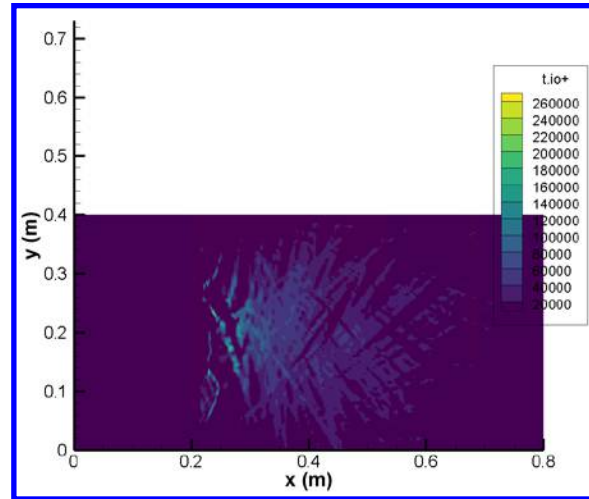


Figure 17: Temperature of ionized O

Figures 15, 16, and 17 show the temperatures of each particle throughout the flow field. The validity of these plots is not as intuitive, primarily figure 15 and 17. These figures show the corresponding particles spiking in temperature where their density is the lowest. This is where the QN solver will be compared to physical testing once the results are available from the UAB project.

Starfish also provides many other interesting values, one of which is the flux of different particles on the surfaces present in the flow field. The average flux of injected O⁺ particles on the plate is 3.54×10^{13} particles per square meter per second. This value is compared to the average flux for the Poisson solver.

D. Poisson Solver Results

The following results were obtained using the Poisson solver. Like the test case results, figure 18 shows the potential as smooth levels in the flow field. Both the insertion point for the O⁺ ions and the plate are held at 0 V due to the imposed boundary conditions. Using these boundary conditions, the non-linear differential equation in equation 23 can be used to obtain the rest of the flow field. It is solved using Newton's method for systems with a standard finite difference. Brieda [11] performed this operation on a 1D system bounded on two ends by a fixed boundary condition and obtained similar results as figure 18. This is consistent with the current problem, although this simulation is in one more spatial dimension.

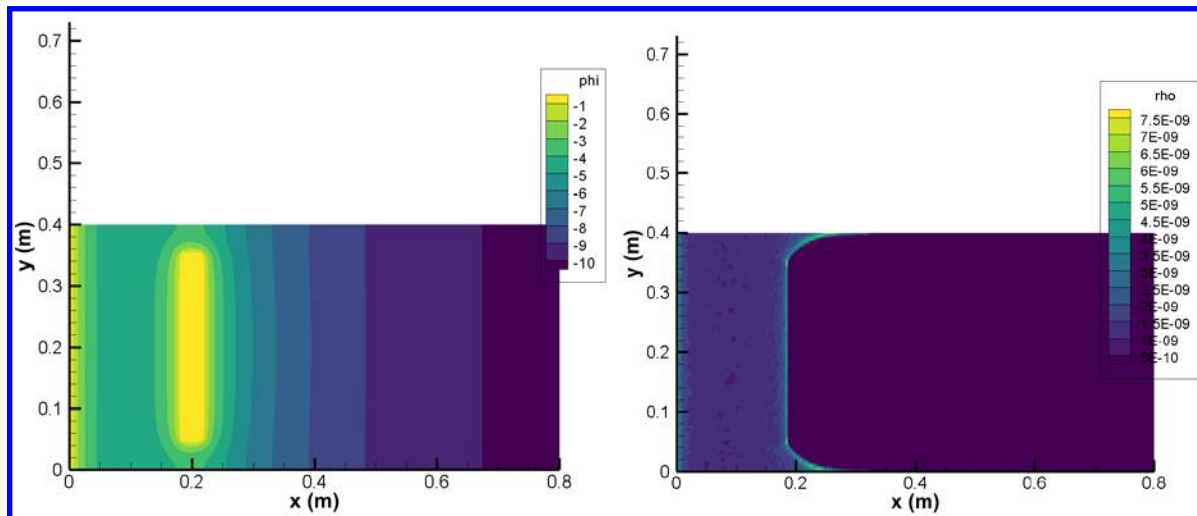


Figure 18: Plasma Potential

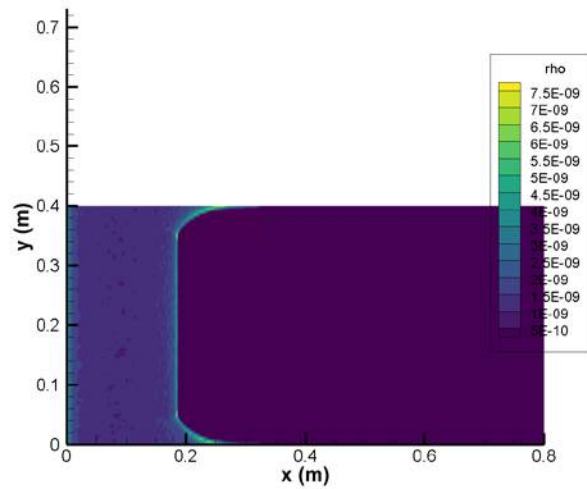


Figure 19: Density of Plasma

Figure 19 shows the density of the plasma and is somewhat like the results from the previous solver. The maximum value of the plasma density using the Poisson solver is around the surface of the plate just as the QN solver shows. The magnitude of this density varies by less than one order-of-magnitude compared with the QN solver, indicating that these two solvers are consistent. The same conclusions can be made from figures 12 and 20 for the inserted O^+ ions.

Figure 21 shows the true nature of the non-linear Poisson solver. The density of the neutral Oxygen atoms does not propagate out towards the insertion point like figure 13 shows for the QN solver. Figure 21 also shows that these particles follow a strict trajectory from their creation at the plate. This is unlike the QN solver which shows the Oxygen clouding around the plate. This is most likely a more realistic representation of what naturally happens. The most common results between the two solvers used is the density of the ionized Oxygen. The densities are nearly identical but the maximum density of the ionized Oxygen for the QN solver occurs closest to the plate, while it is also at a maximum near the insertion point for the non-linear Poisson solver.

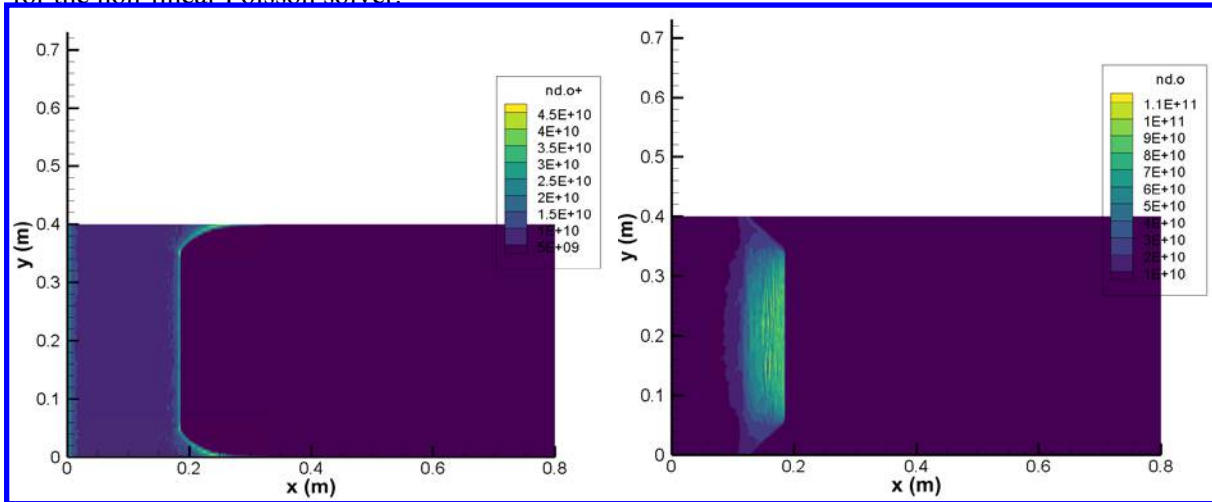


Figure 20: Number density of inserted O^+

Figure 21: Number density of neutral O

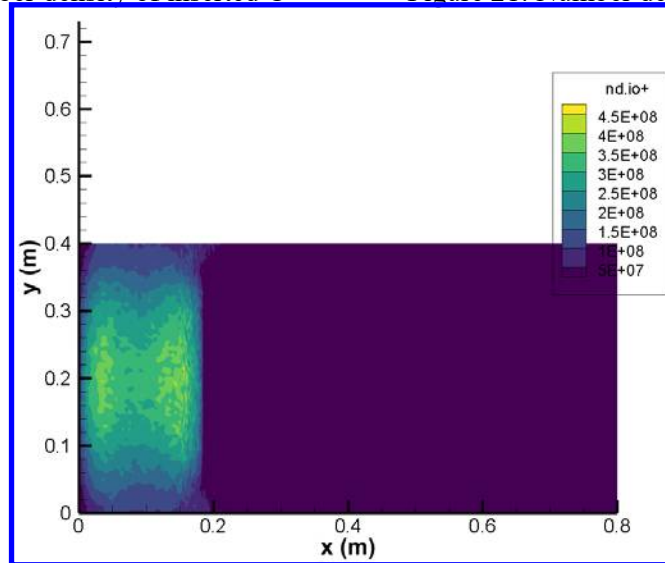


Figure 22: Number density of ionized O

The temperature patterns found using the non-linear Poisson solver differ greatly from the quasi-neutral Boltzmann solver. The previous solver showed that the O^+ temperature was much higher behind the plate and plumed out to lower temperatures. Figure 23 shows the temperature increasing around the

edges of the plate which is more consistent with the density of the O^+ particles in that area. Figures 24 and 25 are consistent with the previous solver aside from the magnitude of the temperatures involved. This is also apparent in the O^+ temperature plot. The temperatures found using this solver are more realistic compared with the previous solver.

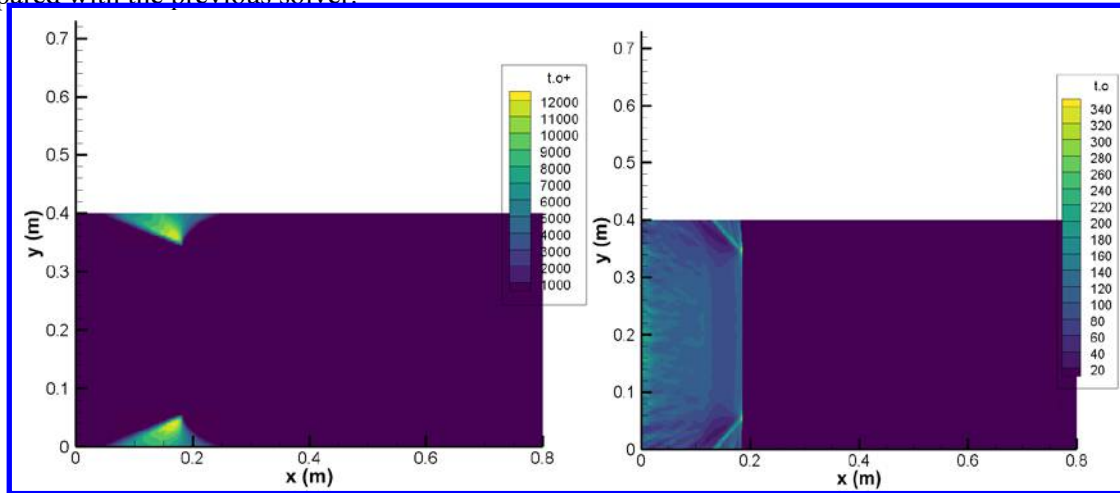


Figure 23: Temperature of O^+

Figure 24: Temperature of neutral O

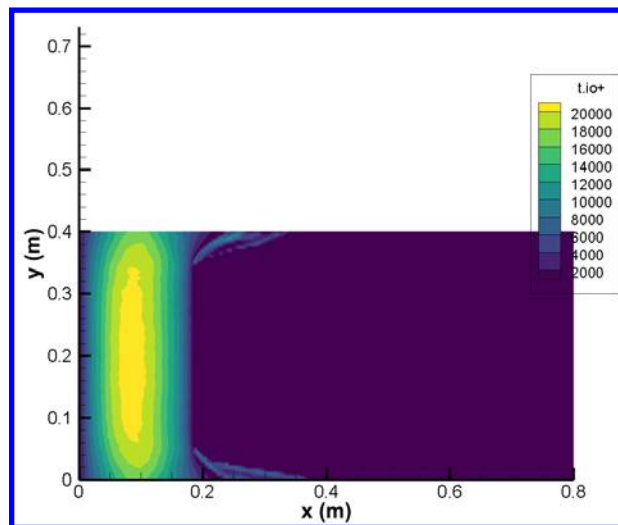


Figure 25: Temperature of ionized O

The average flux of injected O^+ particles on the plate is 3.16×10^{13} particles per square meter per second. This value is comparable to the average flux found from the QN solver.

V. Simulation of LTP Treatment on Surface Modification of Polycaprolactone Pellets

The second application is to improve the understanding of LTP processes and interactions with carbon and hydroxyapatite derived from natural waste materials. For example, Apalangya et al. [13] studied a rapid microwave synthesis of needle-like hydroxyapatite nanoparticles via template directing ball-milled spindle-shaped eggshell particles as shown in Fig. 26. The milled eggshell samples were irradiated using a CEU microwave for 7.5, 15 and 30 min while reagent concentration, the temperature (50°C), and argon purge pressure of 60 psi were kept constant. Muhammed [15] also employs gamma radiation for material processing. They can produce various shapes of particles by plasma treatment and help study microscopic and spectral characterizations.

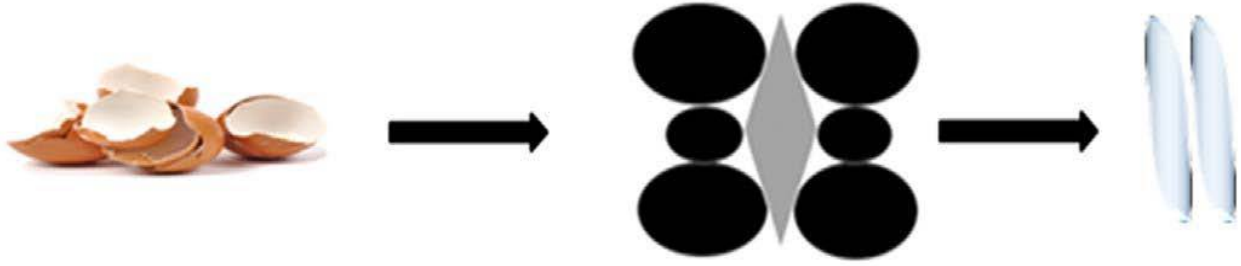


Figure 26: The milling of eggshells into spindle shape particles using a set of stainless-steel balls.

The interaction between Oxygen plasma and the interface in this set up was studied. The chamber used for this process is cylindrical so the “zr” domain was implemented in Starfish. This can be seen in figure 27 along with the size of the domain. A simple mesh was chosen for this simulation and can be seen in figure 28. The materials present in this simulation are the same as the previous application. The interactions implemented are also the same. The inserted Oxygen ions for this simulation were given a velocity of 200 m/s. With the entrance area set to 0.2 m^2 , equation 25 gives a mass flow rate of $1.064 \times 10^{-12} \text{ kg/s}$.

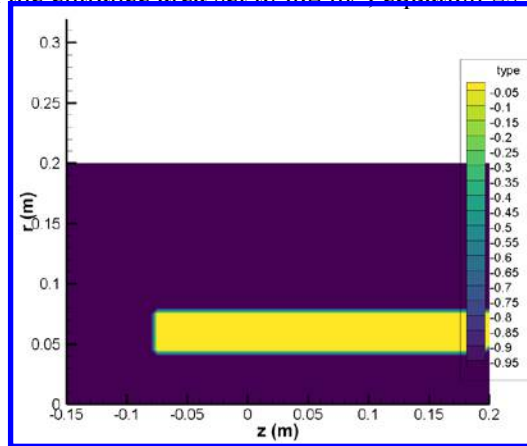


Figure 27: Boundaries

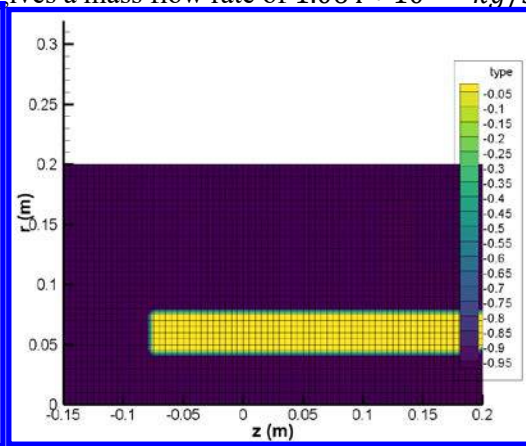


Figure 28: Mesh

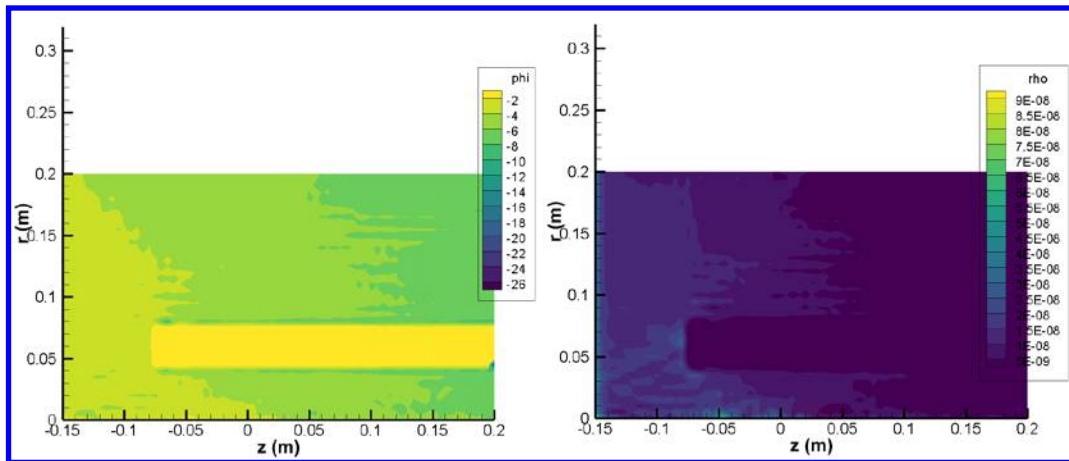


Figure 29: Plasma Potential

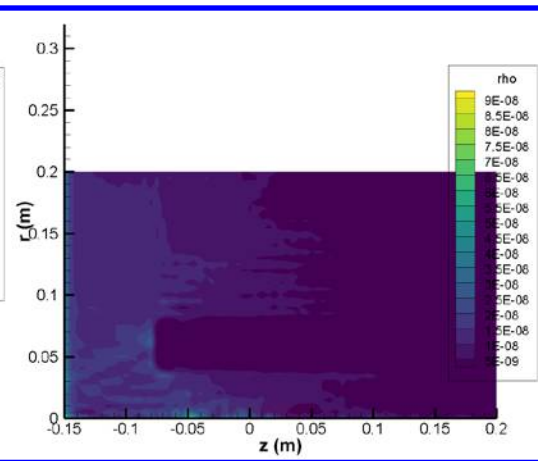


Figure 30: Plasma density

For this application the quasi-neutral Boltzmann, or QN, solver was used. Figure 29 shows the potential throughout the flow field. It is consistent with figure 31 which shows the density of the O^+ in the flow. As with the previous application, the boundary conditions applied were 0 V at the insertion point and

the surface in the flow field. Figure 31 shows how the density of the inserted Oxygen ions has propagated downstream of the insertion point and reaches a maximum at the frontal portion of the plate where their velocity has decreased. Figure 32, like figure 13, shows how the neutral Oxygen particles are only present at the frontal portion of the plate where they are created from the O^+ - stainless-steel interaction.

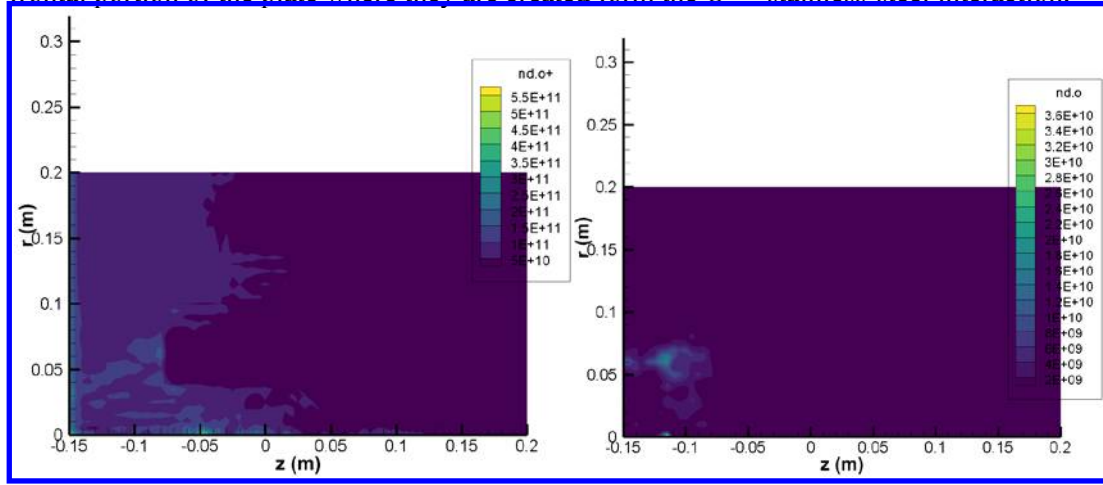


Figure 31: Number density of inserted O^+ Figure 32: Number density of neutral O

VI. Summary and Future Work

For plasma systems where the densities are not low, the kinetic theory is time-consuming and becomes unrealistic. In this regime, the particle-in-cell method is appropriate where the evolution of a particle system at every time step consists of an Eulerian stage and a Lagrangian stage. The PIC method can deal with complex geometries and large distortions in the field. The purpose of this study is to use the Starfish Plasma Simulation Code for numerical simulations of different plasma interfaces. Specifically, two applications were considered: 1) the modeling of a large area (30 cm x 30 cm) microwave plasma chemical vapor deposition system, and 2) the understanding of LTP treatment on surface modification of polycaprolactone pellets and thermal properties of extruded filaments.

To test the open source Starfish code and get familiar with it, the plasma interface over an ellipse has been simulated. The interaction between the plasma and the interfaces present in the two applications presented was simulated. In the first application, the large area interface was tested using both the QN and Poisson solver and their results were compared. The densities calculated by both methods were quite consistent. The only difference between the two solvers were the locations of maximum density and the magnitude of the temperatures in the flow field. In the second application, the quasi-neutral Boltzmann, or QN, solver was used. The interaction between Oxygen plasma and the interface in this set up was similar to that in the first application.

Several changes are going to be made for future simulations for both applications. An exact geometry for the large area microwave plasma chemical vapor deposition system has recently been provided and will be implemented. Future simulations will contain Hydrogen as the carrier gas and diborane for the reactive gas. The substrate temperature will be set to 750°C and the mass flow rate of both gases will also be implemented. The carrier gas and reactive gas will have a flow rate of 500 SCCM and 1 SCCM respectively. New interactions will also be added that result in the creation of $B_{50}C_2$ on the silicon substrate present on the stainless-steel plate. These additions will greatly influence the flow field present in the system. The schematics for the PE-100 chamber from Plasma Etch has been provided and will be implemented in a new simulation as the geometry for the second application. The PCL pellets used in Muhammed's [15] experiment will also be incorporated in the next simulation for this application. Once this simulation has been updated, the resulting flow field will be compared to the experimental results. The effects of transport coefficients for both applications and the rotation of system for the second application will be considered.

Acknowledgement

This material is based upon work supported by the NSF EPSCoR RII-Track-1 Cooperative Agreement OIA-1655280. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors thank our collaborators, Dr. Yogesh Vohra and Dr. Paul Baker at UAB and Dr. Vijaya Rangari at Tuskegee University, for the exact system geometries and experimental results.

Reference

1. Zank, G. P., "Faltering Steps Into the Galaxy: The Boundary Regions of the Heliosphere," *Annual Review of Astronomy and Astrophysics*, 2015. **53**: p. 449-500.
2. Kolobov, V., Arslanbekov, R., Tang, B., and Zank, G. P., "Plasma Simulations with Adaptive Cartesian Mesh," *Basilisk/Gerris Users' Meeting 2017*, Princeton, NJ.
3. Kundrapu, M., Averkin, S., Stoltz, P., and Keidar, M., "Software for plasma device simulations: Arc plasma sources," *AIAA Paper 2018-2940*, 2018.
4. Franck, J. A., Cooney, J. A., and Fine, N. E., "Simulations of Plasma Flow Control Strategies for Trailing Edge Separation," *AIAA Paper 2018-3524*, 2018.
5. Croft, K. A., and Moeller, T. M., "Implementation of a SUPG Stabilized Finite Element Method for a Two Fluid Plasma Model," *AIAA Paper 2018-2942*, 2018.
6. Chen, K-L., Tseng, M-F., Wu, J-S., Sharma, S., Cheng, G. C., and Branam, R., "Development of plasma fluid modeling code with immersed boundary method," *AIAA Paper 2018-2941*, 2018.
7. Wang, X., and Zhong, X., "Development and validation of a high-order shock-fitting non-equilibrium flow solver," *AIAA Paper 2011-365*, 2011.
8. Blanco, A., and Josyula, E., "Numerical modeling of hypersonic weakly ionized external flowfields with Poisson's equation," *AIAA Paper 2019-3354*, 2019.
9. Harlow, F. H., "The particle-in-cell method for numerical solution of problems in fluid dynamics," *Technical Report #LADC-5288*, 1962.
10. Brieda, L., and Keidar, M., "Development of the Starfish Plasma Simulation Code and Update on Multiscale Modeling of Hall Thrusters," *AIAA Paper 2012-4015*, 2012.
11. Brieda, L., "Multiscale Modeling of Hall Thrusters," *Dissertation Paper*, May, 2012
12. Baker, P. A., Catledge, S. A., Harris, S. B., Ham, K. J., Chen, W-C., Chen, C-C., and Vohra, Y. K., "Computational Predictions and Microwave Plasma Synthesis of Superhard Boron-Carbon Materials," *Materials*, 2018. **11**: 1279.
13. Apalangya, V., Rangari, V., Jeelani, S., Dankyi, E., Yaya, A., and Darko, S., "Rapid microwave synthesis of needle-like hydroxyapatite nanoparticles via template directing ball-milled spindle-shaped eggshell particles," *Ceramics International*, 2018. **44**: p. 7165-71.
14. G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Science Publications, 1994.
15. Mohammed, Z., Jeelani, S., and Rangari, V., "Effect of Low-Temperature Plasma Treatment on Surface Modification of Polycaprolactone Pellets and Thermal Properties of Extruded Filaments," *The Minerals, Metals & Materials Society*, Vol. 72, No. 4, 2020