LFR Study US-side Magnetic Nozzle Computational Modeling Report

Nathan Schilling

August 2021

Greetings colleagues! We have been hard at work updating and augmenting our computational model to verify the test case set out in Ref. [1]. We would like to provide an overview of the results so far. For our computational model, we began with the model in Ref. [2] and augmented it to run out tests cases for longer times (the model in [2] cannot run out to the 8 μs time required by Ref. [1]).

1 Our Computational Model

In contrast to the hybrid-Particle in Cell (PIC) model used in Ref. [1], we use a Smooth Particle Hydrodynamic (SPH) model augmented with Maxwell's equations to account for the electrodynamics plasma effects. Our model is called Smoothed Particle Fluid with MAXwell equation-solver (SPFMAX) and a broad overview of the method is given in Ref. [2]. However, we will go into some of the specifics of the model here, and will emphasize the differences between our model and the hybrid PIC model of Ref. [1].

Firstly, SPFMAX is a SPH code, meaning that it uses the SPH method to model hydrodynamic effects. SPH is a Lagrangian method, and so it calculates fluid properties (density, temperature, position) on a 3-dimensional grid that moves with the fluid [3]. Each grid point is called a particle in SPH parlance, because each grid points functions the same as a macro-particle of fluid, with it's own mass, temperature, density, etc [3]. This represents one of the major differences between our method and hybrid PIC; instead of calculating fluid properties by averaging macro-particles over set volumes, we calculate fluid properties directly at each grid point. Also, instead of calculating electromagnetic fields ($\vec{E}$, $\vec{B}$, etc.), at specified grid points and interpolating between particles and grid points, we calculate electromagnetic fields directly at the particle locations themselves. We use these calculations to directly calculate the forces between macro-particles to integrate the equations of motion. Also, in SPH we assume the particles are the result of uniform random sampling of the fluid [3]. We use these samples to construct a kernel function, which describes the fluid properties throughout space and time [3]. Mathematically, a fluid property (A) at grid point a is calculated according to Eq. (1).

$$A_a(r) = \int A(r')W(r - r', h)dr' \tag{1}$$

Here, r is a point a distance away from point a in 3D space, W is the kernel function, h is the compact support distance (see Eq. (3) and surrounding discussion), and r' is the displacement vector. Instead of integrating over the entire domain, we integrate over a b number of nearest neighbors.

$$A_a = \sum_b A_b V_b W_{ab}(r - r', h_{ab}) \tag{2}$$

V_b is the volume of neighboring b number of particles. For the kernel function, we assume a cubic-spline function (see Eq. (3))

$$W_{ab}(q) = \begin{cases} \frac{1}{4\pi h_{ab}^3}[(2-q)^3 - 4(1-q)^3] & \text{for } 0 \le q \le 1 \\ \frac{1}{4\pi h_{ab}^3}(2-q)^3 & \text{for } 1 \le q \le 2 \\ 0 & \text{for } q \ge 2 \end{cases} \tag{3}$$

h_{ab} is the compact support distance (the radius out from a SPH particle where particles inside the radius are considered for constructing the kernel function). h_{ab} must be chosen to satisfy Eq. (4)-(5), making h_{ab} selection very important.

$$\sum_b V_b W_{ab} = 1 \tag{4}$$

$$\sum_b V_b \nabla W_{ab} = 0 \tag{5}$$

We have found that choosing an h_{ab} so that b=60 (only consider the 60 nearest neighbors for a particle) works best [2].

One of the advantages of the SPH method is that by using a kernel function, gradients in the equations of motion can be replaced by the gradient of the kernel function. And since the kernel function is known at all grid points, and the location of each grid point is know, calculating the gradient is easy. In fact, the value of the kernel function, as well as the value of the gradient, is stored as a property of the SPH particle. This allows of easy vectorization of the code, and reduced computational time.

The equations of motion, solved at each time-step by SPFMax are Eq. (6), the continuity equation, Eq. (7), the momentum equation, Eq. (8), the energy equation.

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \vec{u}) \tag{6}$$

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \tau + \frac{1}{\rho}(\vec{j} \times \vec{B}) \tag{7}$$

$$\frac{De}{Dt} = -\frac{p}{\rho}\nabla \cdot \vec{u} + \frac{1}{\rho}\left(\overleftrightarrow{\tau} : \nabla\vec{u}\right) - \frac{1}{\rho}\nabla \cdot (k\nabla T) - 4\sigma_{sb}T^4\chi_{Planck} + \frac{1}{\rho\sigma}j^2 \tag{8}$$

In Eq. (6)-(8), ρ is the mass density, $\vec{u}$ is the velocity vector, $\vec{j}$ is the current density vector, $\vec{B}$ is the magnetic field, t is time, p is the static pressure, $\overleftrightarrow{\tau}$ is the deviatoric viscous stress tensor, k is the thermal conductivity, σ_{sb} is the Stefan-Boltzmann constant, T is temperature, χ_{Planck} is the single group Planck emission opacity, and σ is the plasma conductivity [2]. $\vec{j}$ and $\vec{B}$ are calculated from Ohm's law and Biot-Savart's law.

$$\vec{B} = \frac{\mu_0}{4\pi}\int\int\int_{V_{ab}} \frac{\vec{j}dV_{ab} \times \vec{r'}}{r'} \tag{9}$$

$$\vec{j} = \sigma(\vec{u} \times \vec{B}) \tag{10}$$

Lastly, SPFMax is implemented in MATLAB with the parallel computing toolbox.

2 Test Case

Having discussed the model, we will now discuss our test case. We attempted to replicate the results from Ref. [1]; the input parameters from Ref. [1] are given in the following table. Because of differences in how our simulation works (see

Parameter	Value
Plasma energy	4 MJ
Plasma mass	110 mg
Plasma molecular weight	197 amu
Plasma initial radius	0.3 m
Plasma composition	Gold (Au)
Electron temperature	0 eV
SCM radius	1.0 m
SCM current	3.57 MA
SCM axial position	z=-1.0 m
Calculation domain	6.0 m x 6.0 m x 7.0 m
Mesh number	60 x 60 x 70
Number of macro-particles	100,000

Table 1: Plasma Input Parameters from Ref. [1]

previous section) we must make several changes to the input parameters in Table 1. Firstly, we must change the plasma energy to a temperature (SPH method takes fluid properties as input conditions, like temperature, not particle properties like initial kinetic energy). For this we use an in-house developed equation of state for Gold which yielded an initial temperature of 100 million Kelvin.

Next, we must change the mass to an initial density. Using the initial plasma radius and mass, we found an initial density of 973 mg/m^3 matches with the input conditions. Also, SPFMAx does have the ability to consider a two-temperature plasma, for the temperature we are at, we thought it reasonable to just assume the electrons and ions are at the same temperature. Additionally, SPFMAX only has a capability to track and graph the ion motion - electron motion is considered secondary.

Lastly, while our group does have limited access to supercomputer time, for expediency we ran it on our desktop platform (where runs usually only take an hour or two). So, we had to reduce the number of particles from 100,000 to anywhere between 1,000 to 36,000. We tested a couple of cases within that range. Also, we ran out all cases to the required 8 μs.

In summary, our input parameters are given in Table 2 Instead of running our cases on SX-4, we ran them on a

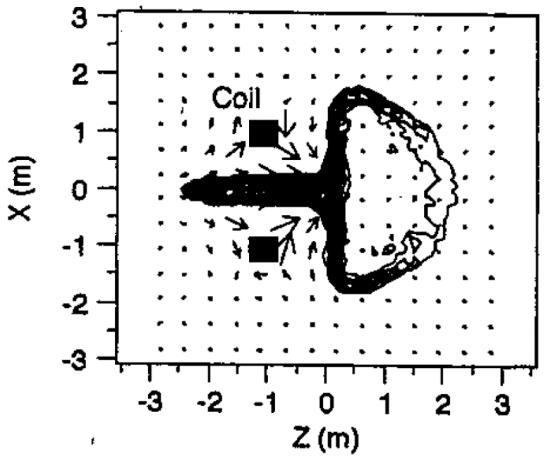
Parameter	**Value**
Plasma temperature	100 $\times 10^6$ K
Plasma initial density	973 mg/m^3
Plasma molecular weight	197 amu
Plasma initial radius	0.3 m
Plasma composition	Gold (Au)
Electron temperature	100 $\times 10^6$ K
SCM radius	1.0 m
SCM current	3.57 MA
SCM axial position	z=-1.0 m
Number of macro-particles	1,000-36,000

Table 2: Our Plasma Input Parameters

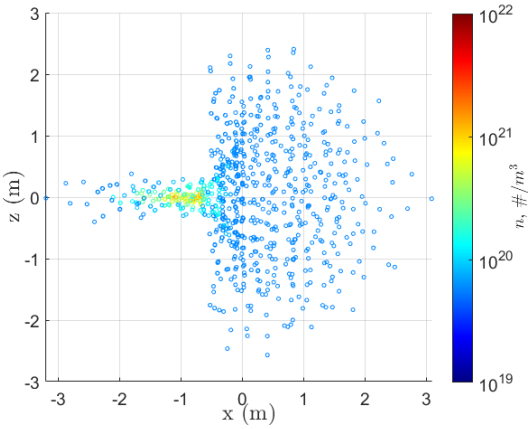
Windows 10 Enterprise machine with an Intel(R)Xenon(R) E5-1630 v4 CPU running at 3.70 GHz, and an NVIDA Quatro M5000 graphics card. We used MATLAB R2020b [4].

3 Preliminary Results

So far, preliminary results have been somewhat of a mixed bag. We tested three different particle resolutions: 1,000 particles, 8,000 particles, and 36,000 particles. Starting with the 1,000 particle case gave us encouraging results.



(a) Reference Case Results. *Reproduction of Fig. 3b t=8μs from Ref. [1]*



(b) SPFMAX Results 1,000 particle resolution

Figure 1: Plots of Reference Case and SPFMAX Output Case

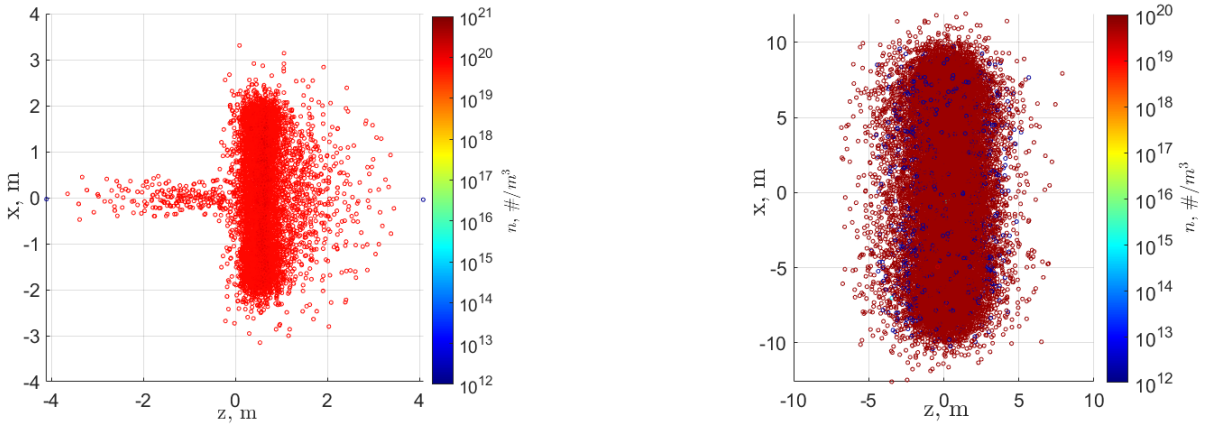
The plasma cloud positions match up fairly well both radially and axially. Both also have similar mushroom cloud shapes. However, the plasma density is concentrated in the tail for the SPFMAX results, which we're not exactly sure is corroborated in Fig. 1a. When we calculate the nozzle efficiency, defined in Eq. (11)in Ref. [1]

$$\eta = \frac{\Sigma M v_z}{\Sigma M |v_0|} \tag{11}$$

we have to, again, tweak it slightly for our SPH method. In our SPH method, fluid macro-particles do not start with an initial velocity; they start with an initial temperature. This temperature induces them to diffuse away from each other. Assuming the fluid has a Maxwellian velocity profile initially, we find the initial speed of each gas particle using Eq. (12)

$$v_0 = \sqrt{3R < T_0 >} \tag{12}$$

where R is the specific gas constant for gold, and $< T_0 >$ is the average initial temperature (100 million Kelvin). Using Eq. (12) to substitute in v_0 in Eq. (11) allows us to calculate nozzle efficiency for our SPFMAX cases. As shown in Table 3, the nozzle efficiency of this case (0.54) is fairly close to the nozzle efficiency in Ref. [1] (0.65). We would expect that, with increasing particle resolution, the our results approach the high-fidelity Ref. [1] results. However, this is not the case. For higher particle resolutions, the plasma cloud seems to get larger and expand more - see Fig. 2.



(a) SPFMAX Results 8,000 particle resolution

(b) SPFMAX Results 36,000 particle resolution

Figure 2: Plots of SPFMAX Output at Two Different resolutions

For the 8,000 particle case, the tail is elongated but the majority of the plasma cloud stays within the limits of the 1,000 particle case. However, for the 36,000 particle case, the plasma expands drastically outside the domain. As shown in Table 3, figures of merit also change drastically for higher particle resolutions. The efficiency drastically increases, then decreases as particle resolution is increased. The final momentum in the z-direction also shows similar trends. However, the final energy in the z-direction seems to increase significantly for higher particle resolutions - which is strange.

Case	η	**% particles escaped**	$\frac{1}{2}\Sigma MV_z^2$ **(MJ)**	ΣMv_z **N/s**
Ref. [1]	0.65	5	-	-
SPFMAX (1,000 particles)	0.54	9.2	1.4	9
SPFMAX (8,000 particles)	1.73	2.1	3.4	26
SPFMAX (36,000 particles)	0.38	13.6	7.0	5

Table 3: Comparison of SPFMAX results with Ref. [1] results

In summary, our current results seem to vary significantly depending on the particle resolution. We are investigating these results further and trying to determine what in the code is dependent on particle resolution, and is giving us these trends. We preliminary believe that the issue has something to do with the time step - the time step is set to the same constant value for all cases, but for the higher resolution cases should probably be lower. We are currently trying to determine how to set the time step dynamically based on the resolution.

References

[1] Y. Nagamine and H. Nakashima, "Analysis of plasma behavior in a magnetic thrust chamber of a laser fusion rocket," *Fusion Technology*, vol. 35, no. 1, pp. 62–70, 1999.

[2] J. Cassibry, K. Schillo, and B. Winterling, "Pulsed magnetic nozzle for fusion propulsion," *Journal of the British Interplanetary Society*, vol. 71, pp. 119–125, 2018.

[3] J. J. Monaghan, "Smoothed particle hydrodynamics," *Reports on progress in physics*, vol. 68, no. 8, p. 1703, 2005.

[4] The Mathworks, Inc., Natick, Massachusetts, *MATLAB version 9.9.0.1524771 (R2020b) Update 2*, 2020.