

# PIC schemes for multi-scale plasma simulations

*Tobias Ott<sup>\*,†</sup> and Marcel Pfeiffer<sup>\*</sup>*

*\*Institute of Space Systems, Germany*

*Pfaffenwaldring 29, 70569 Stuttgart*

*ottt@irs.uni-stuttgart.de · mpfeiffer@irs.uni-stuttgart.de*

*<sup>†</sup>Corresponding author*

## Abstract

An energy conserving method for electrostatic PIC simulations is compared against an explicit scheme using either the continuous Galerkin method or the discontinuous Galerkin method to solve the field equation. The energy conserving method uses Ampère's Law to advance the electric field whereas the explicit methods solve the electrostatic Poisson equation. The three methods are compared using a one dimensional test case. It is shown that, in contrast to the explicit methods, the energy conserving method remains stable even for large time steps. Finally, a three dimensional test case is presented to show the beneficial properties of the energy conserving scheme.

## 1. Introduction

Plasmas play a major role in many modern applications, ranging from space-related topics such as electric propulsion<sup>1</sup> to plasma-based coating technology,<sup>2</sup> which plays a major role in micro and nano manufacturing. In the vast majority of these plasmas, non-equilibrium effects and charge separation effects play a significant role. The correct simulation of these plasmas with the effects mentioned is still extremely complex and only possible with extreme numerical effort. The macroscopic description using continuum-based magnetohydrodynamic equations (MHD) is insufficient in most cases, as the non-equilibrium effects cannot be represented in a physically correct way.<sup>3</sup> On the other hand, plasma kinetic models such as the Particle-In-Cell method can describe non-equilibrium effects correctly, but the numerical complexity of the simulation is high due to the spatial and temporal resolution required for stability.<sup>4</sup> In common electrostatic explicit PIC methods, the Debye length of the plasma usually has to be resolved spatially and the plasma frequency of the electrons has to be resolved temporally. However, this leads to extreme computing times, especially for full 3D simulations, so that many applications cannot currently be simulated for computational reasons. It is often the case that the fine resolution of the electron movement in the plasma is necessary for the stability of the calculation, but the exact movement of the electrons is not of primary importance for the application, e.g. electric propulsion systems, but rather the collective behaviour of the plasma and the behaviour of the significantly slower ions are of interest here.

A possible solution to this problem is hybrid methods in which the electrons are approximated within the PIC simulation using a fluid approach. The simplest variant of this is the Boltzmann fluid in which the electrons are assumed to be isothermal and with instantaneous propagation.<sup>5,6</sup> Although good results can already be achieved for various applications, this assumption fails for plasmas with an applied magnetic field or non-isothermal conditions such as in plasma expansions. Depending on the area of application, there are of course various much more complex fluid models which can also deliver very good results, but the possibility of covering non-equilibrium effects of higher momentum with them is only possible to a certain extent.<sup>7,8</sup> In addition, solving the fluid equations adds complexity to the solver by coupling the PIC method with a hyperbolic fluid solver to handle the electrons.

Another possible approach to the resolution problem in PIC simulations is to treat the electrons using implicit,<sup>9</sup> semi-implicit<sup>10</sup> or energy conserving methods,<sup>11</sup> which suppress the typical plasma heating that results from under-resolving the plasmas in explicit PIC methods. With coarser temporal and spatial discretisations in these methods, electron trajectories are not necessarily resolved exactly, emphasising the accuracy and convergence concerns of implicit methods. However, the advantage is that no additional solver is required to compute the fluid assumptions in addition to the PIC solver. The aim of this article is to investigate this second approach, focusing in particular on the connection of these methods with higher-order solution schemes in terms of accuracy and convergence. In recent years, various Discontinuous Galerkin (DG) methods have emerged as an alternative to conventional finite element methods for solving Poisson problems, offering advantages in terms of the number of degrees of freedom and matrix

conditioning, especially at higher orders.<sup>12,13</sup> However, the non-continuous solution space of DG methods poses challenges when combined with implicit particle solvers and energy-conserving PIC methods. On the other hand, higher order methods provide an opportunity to significantly increase the accuracy of the approach, allowing efficient and accurate solutions even with large time steps. This work presents three methods that have been implemented in the code PICLas<sup>14</sup> and comparative results as a reference on which future studies can build.

## 2. Physical Model

In the context of rarefied plasma, the evolution of the probability density function  $f_\alpha$  of each species  $\alpha$  is governed by the Vlasov equation

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_\alpha = 0. \quad (1)$$

Each physical particle of every species has the same mass  $m_\alpha$  and the same charge  $q_\alpha$ . The physical particles are accelerated due to the electric and magnetic fields generated by other charged particles or by applying external fields. These electric and magnetic fields are given by Maxwell's equations

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (3)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (4)$$

$$\nabla \times \mathbf{B} = \mu_0 \left( \mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right), \quad (5)$$

The charge density  $\rho$  and the charge current density  $\mathbf{J}$  are calculated as follows:

$$\rho = \sum_{\alpha} \int q_{\alpha} f_{\alpha} \, d\mathbf{v} \quad (6)$$

$$\mathbf{J} = \sum_{\alpha} \int q_{\alpha} \mathbf{v} f_{\alpha} \, d\mathbf{v}. \quad (7)$$

In many cases however, the current induced magnetic field can be neglected. Under these conditions, the electroquasistatic limit is used to calculate the electric field. This can be done by solving the Poisson equation

$$\Delta \varphi = -\frac{\rho}{\epsilon_0} \quad (8)$$

or by taking the derivative of Ampère's law (5) and solving the resulting equation:

$$\frac{\partial \Delta \varphi}{\partial t} + \frac{1}{\epsilon_0} \nabla \cdot \mathbf{J} = 0. \quad (9)$$

In the preceding equations, the electrostatic potential  $\varphi$  was introduced, which forces a rotation free electric Field. It is defined as  $\nabla \varphi + \mathbf{E} = 0$ . It is important to note that the modified Ampère's law (9) and the Poisson equation (8) are interchangeable due to charge conservation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0. \quad (10)$$

## 3. Numerical Approach

We approximate the probability density function of each species using  $N_p$  simulation particles

$$f(\mathbf{x}, \mathbf{v}, t) \approx \sum_{p=1}^{N_p} \omega_p \mathcal{S}(\mathbf{x} - \mathbf{x}_p(t)) \delta(\mathbf{v} - \mathbf{v}_p(t)). \quad (11)$$

Each computational particle has a time dependent position  $\mathbf{x}_p(t)$  and velocity  $\mathbf{v}_p(t)$ , as well as a constant weighting  $\omega_p$  that represents the number of physical particles per simulation particle. A shape function  $\mathcal{S}(\mathbf{r})$  may be used to smooth the solution in all spatial dimensions, as suggested by Jacobs et al.<sup>15</sup> However, in this work, we limit the particle shapes

to the Dirac delta,  $\mathcal{S}(\mathbf{x} - \mathbf{x}_p(t)) = \delta(\mathbf{x} - \mathbf{x}_p(t))$ . With the probability density function approximated by computational particles, we then solve the particle trajectories given by the characteristic equations

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \quad (12)$$

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_p}{m_p} \mathbf{E}_p. \quad (13)$$

We discretize these particle trajectories using the leapfrog scheme

$$\mathbf{x}_p^{n+1/2} = \mathbf{x}_p^{n-1/2} + \Delta t \mathbf{v}_p^n \quad (14)$$

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \frac{q_p}{m_p} \mathbf{E}_p^{n+1/2}(\mathbf{x}_p^{n+1/2}). \quad (15)$$

The approximated probability density function (11) can also be inserted into expressions (6) and (7), yielding

$$\rho = \sum_{\alpha} \sum_{p=1}^{N_{\alpha}} q_{\alpha} \omega_p \mathcal{S}(\mathbf{x} - \mathbf{x}_p(t)) \quad (16)$$

$$\mathbf{J} = \sum_{\alpha} \sum_{p=1}^{N_{\alpha}} q_{\alpha} \omega_p \mathbf{v}_p(t) \mathcal{S}(\mathbf{x} - \mathbf{x}_p(t)). \quad (17)$$

### 3.1 Field Solver

The task of the field solver is to calculate the electric field at the particle positions  $\mathbf{E}_p^{n+1/2}(\mathbf{x}_p^{n+1/2})$  depending on the corresponding charge density and boundary conditions. In the particle-in-cell method, the electric field is evaluated on a grid and then interpolated onto the particle positions. In this study, we compare three different methods for calculating the electric field: a discretization of the Poisson equation using the continuous Galerkin spectral element method (CG-SEM) or the hybridizable discontinuous Galerkin spectral element method (HDG-SEM), and a discretization of the modified Ampère's law using the CG-SEM, which results in the energy conserving method (ECSim) by Lapenta.<sup>16</sup> The CG-SEM and HDG-SEM are spectral element methods that can provide high accuracy and flexibility in handling complex geometries when used at a high order. Compared to the CG-SEM, the HDG-SEM may offer faster convergence due to a smaller number of globally coupled unknowns, especially for high order simulations.<sup>17</sup> This faster calculation of the electric field can significantly speed up the entire code, as the electric field needs to be calculated in each time step. On the other hand, the ECSim, which is based on the discretization of the modified Ampère's law, has the advantage of being energy conserving. This property is promising as it essentially eliminates the restrictive grid heating and is unconditionally stable for all time steps.<sup>16</sup> However, each method also has its potential drawbacks. For instance, the HDG-SEM, despite its potential for faster convergence, introduces discontinuity in the deposited charge density, which may lead to oscillations.

In the following subsections, we will briefly cover the derivation of each method. We refer to the book from Kopriva<sup>18</sup> for an in-depth introduction into the CG-SEM and to a series of papers by Cockburn et al.<sup>19–22</sup> for detailed derivation of the HDG method. The derivation of the ECSim with finite element methods is found in Parodi et al.<sup>11</sup>

#### 3.1.1 Continuous Galerkin discretization of the Poisson equation

In the CG-SEM, Poisson's equation (8) is multiplied by a test function  $\phi$  and integrated over the whole computational domain  $\Omega$

$$\int_{\Omega} \Delta \phi \phi \, d\mathbf{x} = - \int_{\Omega} \frac{1}{\varepsilon_0} \rho \phi \, d\mathbf{x}. \quad (18)$$

Applying partial integration, we obtain the weak form of the equation

$$\int_{\Omega} \nabla \phi \cdot \nabla \phi \, d\mathbf{x} = - \int_{\Omega} \frac{1}{\varepsilon_0} \rho \phi \, d\mathbf{x}. \quad (19)$$

In this work, we consider exclusively periodic test cases. Thus, the surface integral at the domain boundaries is omitted. What remains is to insert the expression for the charge density (16) and the approximation for the solution

$$\varphi \approx \varphi_h = \sum_j \hat{\varphi}_j \psi_j \quad (20)$$

## PIC SCHEMES FOR MULTI-SCALE PLASMA SIMULATIONS

with  $\psi_j$  being the continuous basis functions and  $\hat{\varphi}$  being the according weight. Also, we insert the Galerkin ansatz and choose the basis functions the test functions,  $\phi = \psi_i$ .

$$\int_{\Omega} \sum_j \hat{\varphi}_j \nabla \psi_j \cdot \nabla \psi_j \, d\mathbf{x} = -\frac{1}{\varepsilon_0} \sum_p \omega_p q_p \psi_i(\mathbf{x}_p). \quad (21)$$

The remaining integrals are approximated using Lobatto quadrature, which uses the same nodes as the basis functions. This results in a sparse symmetric linear system of equations, that can be solved either directly or by using an iterative method. In our work, we implemented the library PETSc<sup>23</sup> to solve large linear systems. For the small one-dimensional test cases, a direct solver, Cholesky decomposition, was used. When the system becomes too large, we instead opt for the conjugate gradient method with ILU preconditioning.

The simplest approximation are linear basis functions inside the elements, which leads to the standard finite element method. This second order scheme is identical to the widely used linear  $b_1$ -spline shape functions for the particles and the finite difference scheme to solve Poisson's equation.

### 3.1.2 Hybridizable discontinuous Galerkin discretization of the Poisson equation

In contrast to the CG-SEM, for the HDG-SEM we do not start with the Poisson equation (8), but with the divergence equation (2).<sup>24</sup> The system to solve is given by the following equations

$$\mathbf{E} + \nabla \varphi = 0 \quad (22)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \quad (23)$$

$$(\mathbf{E}^* \cdot \mathbf{n})^- + (\mathbf{E}^* \cdot \mathbf{n})^+ = 0 \quad (24)$$

The last equation enforces continuity of the electric field across interfaces between elements. It is needed because new variables are introduced at the interfaces and will couple the electric fields between elements as we will see in the following lines. First however, the equations are brought to the variational form by multiplying with the test functions and integrating over one element. Then, we integrate equation (22) by parts to obtain the weak form and integrate equation (23) twice to obtain the strong form. The resulting system reads

$$\int_{\mathcal{E}} \mathbf{E} \cdot \Phi \, d\mathbf{x} - \int_{\mathcal{E}} \varphi \nabla \cdot \Phi \, d\mathbf{x} + \int_{\partial \mathcal{E}} \varphi^* \Phi \cdot \mathbf{n} \, d\mathbf{x} = 0 \quad (25)$$

$$- \int_{\mathcal{E}} \nabla \cdot \mathbf{E} \phi \, d\mathbf{x} - \int_{\partial \mathcal{E}} (\mathbf{E}^* - \mathbf{E})^- \cdot \mathbf{n} \phi \, d\mathbf{x} = \int_{\mathcal{E}} \frac{\rho}{\varepsilon_0} \phi \, d\mathbf{x} \quad (26)$$

$$\int_{\Gamma} ((\mathbf{E}^* \cdot \mathbf{n})^- + (\mathbf{E}^* \cdot \mathbf{n})^+) \bar{\lambda} \, d\mathbf{x} = 0 \quad (27)$$

where  $u$ ,  $\Phi$  and  $\lambda$  are the test functions for the three equations. The first two equations are integrated over one element  $\mathcal{E}$  whereas the third equation is integrated over one interface  $\Gamma$  between elements. Now, numerical traces for  $\varphi^*$  and  $(\mathbf{E}^* \cdot \mathbf{n})$  have to be introduced. We follow the ansatz of Cockburn et al.<sup>22</sup> and choose

$$\varphi^* = \lambda \quad (28)$$

$$(\mathbf{E}^* \cdot \mathbf{n})^{+/-} = (\mathbf{E} \cdot \mathbf{n})^{+/-} + \tau(\varphi^{+/-} - \lambda) \quad (29)$$

with  $\tau > 0$  being the stabilization parameter. In the ansatz, we have introduced a new variable  $\lambda$  which describes the unique potential at the faces between elements. After also inserting the approximate solution

$$\mathbf{E} \approx \mathbf{E}_h = \sum_j \hat{E}_j \Phi_j(\mathbf{x}), \quad \varphi \approx \varphi_h = \sum_j \hat{\varphi}_j \phi_j(\mathbf{x}), \quad \lambda \approx \lambda_h = \sum_j \hat{\lambda}_j \bar{\lambda}_j(\mathbf{x}) \quad (30)$$

with the Galerkin ansatz, we arrive at the final set of equations to solve

$$\int_{\mathcal{E}} \mathbf{E}_h \cdot \Phi_i \, d\mathbf{x} - \int_{\mathcal{E}} \varphi_h \nabla \cdot \Phi_i \, d\mathbf{x} + \int_{\partial \mathcal{E}} \lambda_h \Phi_i \cdot \mathbf{n} \, d\mathbf{x} = 0 \quad (31)$$

$$- \int_{\mathcal{E}} \nabla \cdot \mathbf{E}_h \phi_i \, d\mathbf{x} - \int_{\partial \mathcal{E}} \tau(\varphi_h^- - \lambda_h) \cdot \mathbf{n} \phi_i \, d\mathbf{x} = \sum_p \frac{\omega_p}{\varepsilon_0} \phi_i(\mathbf{x}_p) \quad (32)$$

$$\int_{\Gamma} ((\mathbf{E}_h \cdot \mathbf{n}) + \tau(\varphi_h - \lambda_h))^- \bar{\lambda} \, d\mathbf{x} + \int_{\Gamma} ((\mathbf{E}_h \cdot \mathbf{n}) + \tau(\varphi_h - \lambda_h))^+ \bar{\lambda}_i \, d\mathbf{x} = 0 \quad (33)$$

While the system has more degrees of freedom, only the introduced variable  $\lambda$  must be calculated globally. The potential and the electric field can be calculated in an element-local post-processing step that is easily parallelized.

### 3.1.3 Continuous Galerkin discretization of Ampère's law

In order to calculate the electric field with the modified Ampère's law, we first discretize the time derivatives using the implicit midpoint method, as it is done in the ECSim<sup>16</sup>

$$\Delta\delta\varphi = -\frac{\Delta t}{\varepsilon_0} \nabla \cdot \mathbf{J}^{n+1/2} \quad \text{with} \quad \delta\varphi = \varphi^{n+1} - \varphi^n. \quad (34)$$

Since the electrostatic potential and thus the electric field is given at times  $n$  and  $n+1$ , the field must be interpolated to  $n+1/2$  for equation (15). This is done with a simple linear interpolation

$$\mathbf{E}^{n+1/2} = \frac{\mathbf{E}^n + \mathbf{E}^{n+1}}{2}. \quad (35)$$

Again, we multiply the equation by the test functions, integrate over the domain and integrate by parts to obtain the weak form of the equation

$$\int_{\Omega} \nabla\delta\varphi \cdot \nabla\phi \, d\mathbf{x} = -\frac{\Delta t}{\varepsilon_0} \int_{\Omega} \mathbf{J}^{n+1/2} \cdot \nabla\phi \, d\mathbf{x}. \quad (36)$$

Now, the expression for the charge current density must be inserted

$$\mathbf{J}^{n+1/2} = \sum_p \omega_p \frac{\mathbf{v}_p^{n+1} + \mathbf{v}_p^n}{2} \delta(\mathbf{x} - \mathbf{x}_p^{n+1/2}) \quad (37)$$

Inserting this leads to an implicit scheme, hence the name ECSim. However, inserting equation (15) and equation (35), together with  $\mathbf{E} = -\nabla\varphi$ , leads to the following expression

$$\mathbf{J}^{n+1/2} = \sum_p \omega_p \left( \mathbf{v}_p^n + \frac{\Delta t}{2} \frac{q_p}{m_p} \nabla\varphi_p^{n+1/2} \right) \delta(\mathbf{x} - \mathbf{x}_p^{n+1/2}) \quad (38)$$

$$= \sum_p \omega_p \left( \frac{\Delta t}{4} \frac{q_p}{m_p} \nabla\delta\varphi_p \right) \delta(\mathbf{x} - \mathbf{x}_p^{n+1/2}) + \sum_p \omega_p \left( \mathbf{v}_p^n + \frac{\Delta t}{2} \frac{q_p}{m_p} \nabla\varphi_p^n \right) \delta(\mathbf{x} - \mathbf{x}_p^{n+1/2}), \quad (39)$$

where  $\varphi_p = \varphi(\mathbf{x}_p^{n+1/2})$ . Inserting this expression into the weak form (36) as well as the approximate solution (20) and the Galerkin ansatz, the final equation reads

$$\sum_j \delta\hat{\varphi}_j \left( \int_{\Omega} \nabla\psi_j \cdot \nabla\psi_i \, d\mathbf{x} + \sum_p \frac{\Delta t^2 \omega_p q_p}{4\varepsilon_0 m_p} \nabla\psi_{j,p} \cdot \nabla\psi_{i,p} \right) = - \sum_p \left( \frac{\Delta t \omega_p}{\varepsilon_0} \mathbf{v}_p^n + \frac{\Delta t^2 \omega_p q_p}{2\varepsilon_0 m_p} \sum_j \hat{\varphi}_j^n \nabla\psi_{j,p} \right) \cdot \nabla\psi_{i,p}. \quad (40)$$

Again, the remaining integral is approximated by Lobatto quadrature and the resulting linear system is solved

## 4. Results

### 4.1 1D Plasma Expansion

In this section, we compare the performance of the explicit implementation using the HDG-SEM with the explicit and semi-implicit schemes using the CG-SEM. We simulate the 1D plasma expansion testcase presented by Parodi et al.<sup>11</sup> as our benchmark. The simulation is conducted on a periodic grid of size  $L = 2000\lambda_D$ , with half of the domain initially filled with a quasi-neutral plasma with number density  $n_e = n_i = 5 \times 10^{16}$ . The Debye length is calculated by with  $\lambda_D = \sqrt{\varepsilon_0 k_B T_e / (n_e q_e^2)}$ . The electrons ( $m_e \approx 9.109 \times 10^{-31}$  kg,  $q_e = -e$ ) are initialized with the temperature  $T_e k_B = 1$  eV. The ions ( $m_i \approx 1.673 \times 10^{-27}$  kg,  $q_i = e$ ) are initialized with the temperature  $T_i = 300$  K.

In Case 1, we ensure that both temporal and spatial scales are well resolved. The time step is set to  $\Delta t = 0.2\omega_e$ , with the electron plasma frequency  $\omega_e = \sqrt{n_e q_e^2 / (\varepsilon_0 m_e)}$ , which is considered the accuracy limit for explicit solvers.<sup>4</sup> The element length is set to  $\Delta x = 0.5\lambda_D$ , resolving the Debye length and thus preventing excessive grid heating. Figure 1 shows the simulation results for the this case. All three simulation methods are close agreement with each other and with the result reported by Parodi et al.<sup>11</sup> However, the ECSim exhibits slightly increased noise in the solution, which can be attributed to the sampling of the charge current density instead of the charge density. As the sampled current density for ECSim is the first moment of the distribution function and the sampled charge density for the other two methods is the zeroth moment of the distribution function, it remains more noisy for the same number of particles.

## PIC SCHEMES FOR MULTI-SCALE PLASMA SIMULATIONS

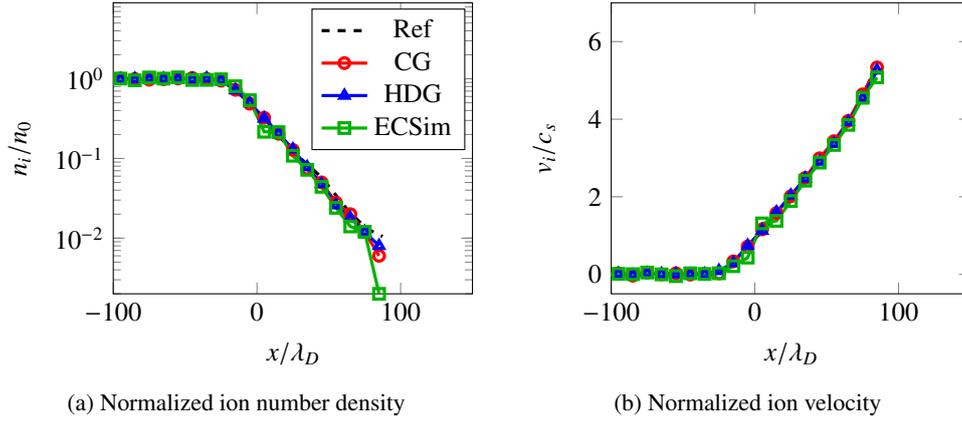


Figure 1: Ion densities and velocities for the fully resolved test case ( $\Delta x = 0.5\lambda_D$ ,  $\Delta t = 0.2\omega_e$ ) at time  $t = 22.08\omega_i$ . The black dashed line shows approximately the solution of the fully resolved case by Parodi et al.<sup>11</sup> Ion speed of sound  $c_s = \sqrt{kT_e/m_i}$  used for the nondimensionalization of the ion velocities. The plot shows the area of the plasma boundary.

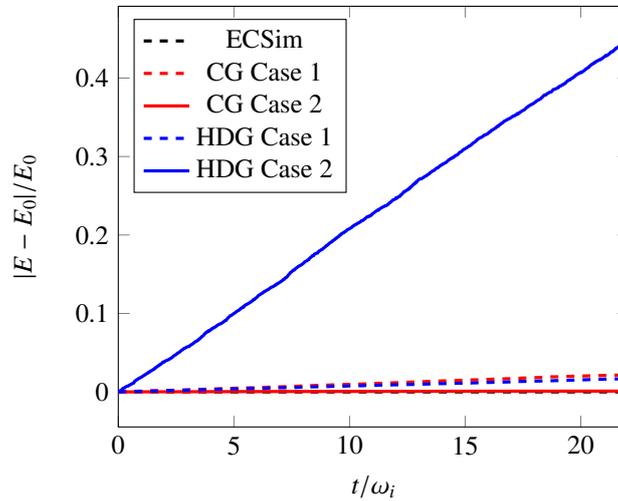


Figure 2: Total energy variation over time for the different methods. *Case 1* denotes the fully resolved test case with cell size  $\Delta x = 0.5\lambda_D$  and time step  $\Delta t = 0.2\omega_e$ . *Case 2* denotes the spatially underresolved test case with the same time step, but with cell size  $\Delta x = 10\lambda_D$ .

In Case 2, we increase the element size to  $\Delta x = 10\lambda_D$ , as done by Parodi et al.<sup>11</sup> The new mesh does not resolve the Debye length and grid heating is expected when an explicit time integration scheme is used. Figure 2 displays the total energy variation for both Case 1 and Case 2. As expected, the total energy remains constant when using the ECSim within the precision of the solver. Interestingly, the CG and HDG methods exhibit different behaviors in Case 2. While the HDG method experiences the expected grid heating, the CG method does not, and its energy remains nearly constant. This unexpected result requires further investigation. Looking at the ion number density and ion velocity plots in Figure 3, we can see that the result obtained from the CG method closely aligns with those from the ECSim. For both methods, the expansion lags slightly behind that of the reference simulation. The HDG method, on the other hand, exhibits noticeable grid heating. This leads to an inflated ion velocity, causing the expansion to occur at a faster rate than it does with the smaller cell size.

In the final test, Case 3, we also increase the time step size to  $\Delta t = 0.2\omega_i$ , with the ion plasma frequency  $\omega_i = \sqrt{n_i q_i^2 / (\epsilon_0 m_i)}$ . Now, only the trajectory of the ions is well resolved and plasma oscillations by the electrons cannot be captured. The explicit time integration methods become unstable at the given time step size, leading to a rapid increase in energy and to an unphysical behaviour of the particles, so that the solver cannot produce a result. Therefore, Figure 4 only shows the solution of the ECSim. Although the density deviation in the expansion region is relatively large, the ion velocity of the reference solution is very well matched. As this is an expansion into vacuum,

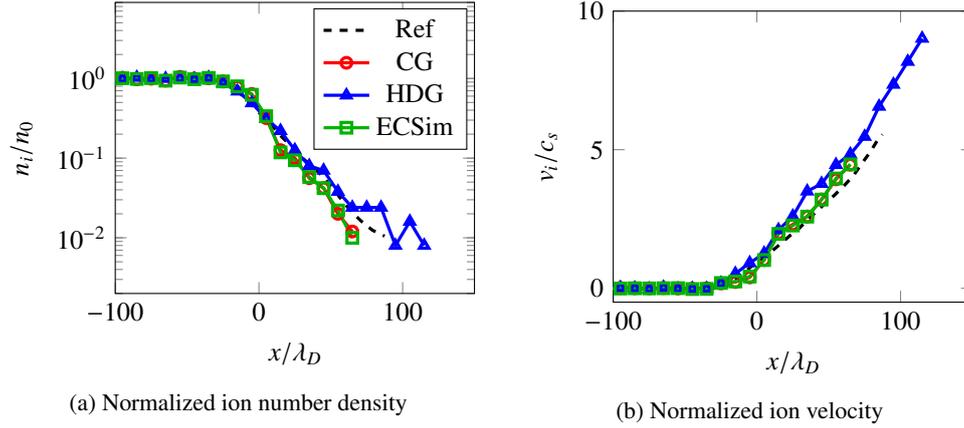


Figure 3: Ion densities and velocities for the spatially underresolved test case ( $\Delta x = 10\lambda_D$ ,  $\Delta t = 0.2\omega_e$ ) at time  $t = 22.08\omega_i$ . The black dashed line shows approximately the solution of the fully resolved case by Parodi et al.<sup>11</sup> Ion speed of sound  $c_s = \sqrt{kT_e/m_i}$  used for the nondimensionalization of the ion velocities. The plot shows the area of the plasma boundary.

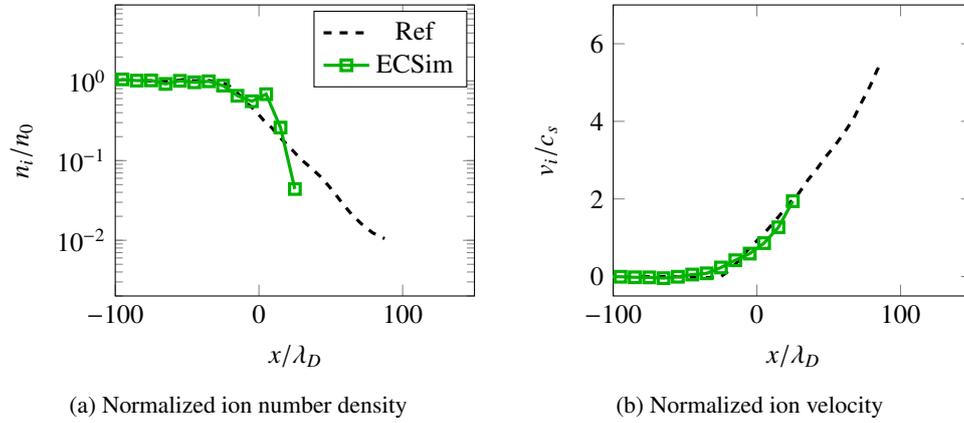


Figure 4: Ion densities and velocities for the underresolved test case ( $\Delta x = 10\lambda_D$ ,  $\Delta t = 0.2\omega_i$ ) at time  $t = 22.08\omega_i$ . The black dashed line shows approximately the solution of the fully resolved case by Parodi et al.<sup>11</sup> Ion speed of sound  $c_s = \sqrt{kT_e/m_i}$  used for the nondimensionalization of the ion velocities. The plot shows the area of the plasma boundary.

the density gradient here is very large, so the noise from the source of the field solver in the form of the current density may also play a greater role. If greater accuracy is required, electron sub-stepping could also reduce the integration errors of the fast electrons in these cases. In summary, however, the ECSim is a very promising way to perform plasma simulations much more effectively, since many restrictions on spatial and temporal resolution are removed without having to solve the particles iteratively as in fully implicit methods.

#### 4.2 3D Plasma Expansion

To demonstrate the capabilities of the ECSim in a three-dimensional test case, we simulate the expansion of an initially quasi-neutral and homogeneous plasma ball. The computational domain is a periodic cube with dimensions of  $80 \times 80 \times 80$  elements, each with length  $\Delta x = 10\lambda_D$ . The plasma sphere, with a radius of  $r_0 = 400\lambda_D/3$ , is initialized with 500000 particles per species. All other parameters remain identical to those in the one-dimensional test case.

Figure 5 shows the normalized density and the magnitude of the outward-pointing velocity vector as a function of the radius. The simulation was performed using the ECSim with a large time step of  $\Delta t = 0.2\omega_i$ . Again, the explicit methods were unable to produce a result due to the instability of the time integrator. As observed in the one-dimensional test case, the ions are accelerated outwards.

In future work, the accuracy of the unresolved ECSim method will be compared with a resolved reference for the 3D plasma expansion, and methods will be developed to improve this for ECSim. This is only a demonstration of the

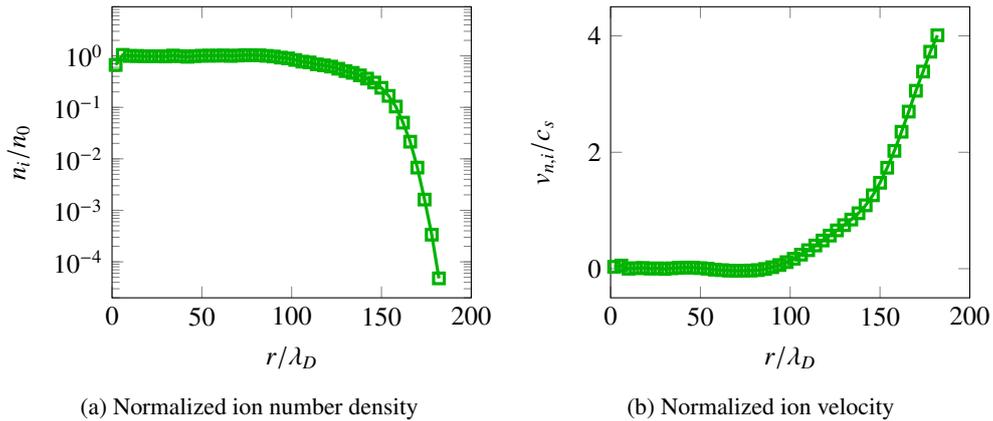


Figure 5: Ion densities and velocities for the three dimensional test case at time  $t = 22.08\omega_i$  calculated with the ECSim. Ion speed of sound  $c_s = \sqrt{kT_e/m_i}$  used for the nondimensionalization of the ion velocities.

3D capabilities of the method in principle, as these play an even greater role in 3D simulations. To resolve the Debye length,  $1600 \times 1600 \times 1600$  elements would have been needed, which is 8000 times more degrees of freedom than the simulation shown here, and 40 times more time steps would have to be calculated if the electron plasma frequency had to be resolved instead of the ion plasma frequency. Thus, ECSim clearly shifts the computation time for such simulations into ranges that are still computable without a large high-performance cluster.

## 5. Conclusion and Future Work

We have presented and compared three different methods for calculating the electric field in a PIC simulation: a discretization of the Poisson equation using the continuous Galerkin spectral element method (CG-SEM) or the hybridizable discontinuous Galerkin spectral element method (HDG-SEM), and a discretization of the modified Ampère’s law using the CG-SEM (ECSim).

While considerable grid heating was observed with the HDG-SEM in the spatially underresolved one-dimensional test case, the CG-SEM did not show this issue. When the time step was set to a much larger value than the electron plasma frequency, the two explicit methods became unstable. In this case, only the ECSim was able to produce reasonable simulation results. The favorable properties of the energy-conserving scheme were further demonstrated in a three-dimensional test case.

Looking ahead, an energy-conserving implementation of the HDG method may be even more beneficial as it reduces the number of globally coupled degrees of freedom, thereby enabling faster computation of the electric field. Also, the use of high-order basis functions allows for a further reduction of coupled unknowns.

In our future work, we plan to investigate the use of particle sub-stepping for electrons in order to reduce integration errors.

## 6. Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 899981 MEDUSA).

## References

- [1] Manuel Martinez-Sanchez and James E Pollard. Spacecraft electric propulsion-an overview. *Journal of propulsion and power*, 14(5):688–699, 1998.
- [2] Robert B Heimann. *Plasma-spray coating: principles and applications*. John Wiley & Sons, 2008.
- [3] Peter Alan Davidson. *An introduction to magnetohydrodynamics*, 2002.
- [4] Charles K. Birdsall and A. Bruce Langdon. *Plasma Physics via Computer Simulation*. McGraw-Hill, New York, 1985.

- [5] Yuan Hu and Joseph Wang. Electron properties in collisionless mesothermal plasma expansion: Fully kinetic simulations. *IEEE Transactions on Plasma Science*, 43(9):2832–2838, 2015.
- [6] Iain D Boyd. Review of hall thruster plume modeling. *Journal of Spacecraft and Rockets*, 38(3):381–387, 2001.
- [7] Eduardo Ahedo. Using electron fluid models to analyze plasma thruster discharges. *Journal of Electric Propulsion*, 2(1):2, 2023.
- [8] Ari Le, Adam Stanier, Lin Yin, Blake Wetherton, Brett Keenan, and Brian Albright. Hybrid-vpic: An open-source kinetic/fluid hybrid particle-in-cell code. *Physics of Plasmas*, 30(6), 2023.
- [9] Guangye Chen, Luis Chacón, and Daniel C Barnes. An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm. *Journal of Computational Physics*, 230(18):7018–7036, 2011.
- [10] Giovanni Lapenta, JU Brackbill, and Paolo Ricci. Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas. *Physics of plasmas*, 13(5), 2006.
- [11] P Parodi, G Lapenta, and T Magin. Simulation of plasmas for electric propulsion using the energy-conserving semi-implicit PIC scheme. In *Proceedings of the 9th European Conference for Aerospace Sciences*, Lille, France, 2022.
- [12] Bernardo Cockburn, Bo Dong, Johnny Guzmán, Marco Restelli, and Riccardo Sacco. A hybridizable discontinuous galerkin method for steady-state convection-diffusion-reaction problems. *SIAM Journal on Scientific Computing*, 31(5):3827–3846, 2009.
- [13] Douglas N Arnold, Franco Brezzi, Bernardo Cockburn, and L Donatella Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5):1749–1779, 2002.
- [14] Claus-Dieter Munz, Monika Auweter-Kurtz, Stefanos Fasoulas, Asim Mirza, Philip Ortwein, Marcel Pfeiffer, and Torsten Stindl. Coupled Particle-In-Cell and Direct Simulation Monte Carlo method for simulating reactive plasma flows. *Comptes Rendus Mécanique*, 342(10-11):662–670, October 2014.
- [15] G.B. Jacobs and J.S. Hesthaven. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 214(1):96–121, May 2006.
- [16] Giovanni Lapenta. Exactly Energy Conserving Semi-Implicit Particle in Cell Formulation. *Journal of Computational Physics*, 334:349–366, April 2017.
- [17] Bernardo Cockburn and Jayadeep Gopalakrishnan. The Derivation of Hybridizable Discontinuous Galerkin Methods for Stokes Flow. *SIAM Journal on Numerical Analysis*, 47(2):1092–1125, January 2009.
- [18] David A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Scientific Computation. Springer Netherlands, Dordrecht, 2009.
- [19] Bernardo Cockburn, Guido Kanschat, Iliaria Perugia, and Dominik Schötzau. Superconvergence of the Local Discontinuous Galerkin Method for Elliptic Problems on Cartesian Grids. *SIAM Journal on Numerical Analysis*, 39(1):264–285, January 2001.
- [20] Bernardo Cockburn, Bo Dong, and Johnny Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Mathematics of Computation*, 77(264):1887–1916, May 2008.
- [21] Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, January 2009.
- [22] Bernardo Cockburn, Bo Dong, Johnny Guzmán, Marco Restelli, and Riccardo Sacco. A Hybridizable Discontinuous Galerkin Method for Steady-State Convection-Diffusion-Reaction Problems. *SIAM Journal on Scientific Computing*, 31(5):3827–3846, January 2009.
- [23] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil M. Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, Jacob Faibussowitsch, William D. Gropp, Václav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. PETSc Web page, 2023.

PIC SCHEMES FOR MULTI-SCALE PLASMA SIMULATIONS

- [24] Marcel Pfeiffer, Florian Hindenlang, Timon Binder, Steven Coplestone, Claus-Dieter Munz, and Stefanos Fasoulas. A Particle-in-Cell solver based on a high-order hybridizable discontinuous Galerkin spectral element method on unstructured curved meshes. *Computer Methods in Applied Mechanics and Engineering*, 349:149–166, June 2019.