

Enhanced biDimensional pIc: an electrostatic/magnetostatic particle-in-cell code for plasma based systems

G. Gallina^{1,2}, M. Magarotto^{3,4,†}, M. Manente⁵ and Daniele Pavarin^{3,4,5}

¹Department of Physics and Astronomy, University of British Columbia, 6224 Agricultural Road,
Vancouver BC V6T 1Z1, CA

²TRIUMF, 4004 Wesbrook Mall, Vancouver BC V6T 2A3, CA

³Department of Industrial Engineering, University of Padova, Via Gradenigo 6/a 35131 Padova, IT

⁴Centro di Ateneo di Studi e Attività Spaziali ‘Giuseppe Colombo’ – CISAS, University of Padova,
Via Venezia 15 35131 Padova, IT

⁵Technology for Propulsion and Innovation S.r.l., Via della Croce Rossa 112 35129 Padova, IT

(Received 9 October 2018; revised 19 February 2019; accepted 20 February 2019)

EDI (enhanced biDimensional pIc) is a two-dimensional (2-D) electrostatic/magnetostatic particle-in-cell (PIC) code designed to optimize plasma based systems. The code is built on an unstructured mesh of triangles, allowing for arbitrary geometries. The PIC core is comprised of a Boris leapfrog scheme that can manage multiple species. Particle tracking locates particles in the mesh, using a fast and simple priority-sorting algorithm. A magnetic field with an arbitrary topology can be imposed to study the magnetized particle dynamics. The electrostatic fields are then computed by solving Poisson’s equation with a finite element method solver. The latter is an external solver that has been properly modified in order to be integrated into EDI. The major advantage of using an external solver directly incorporated into the EDI structure is its strong flexibility, in fact it is possible to couple together different physical problems (electrostatic, magnetostatic, etc.). EDI is written in C, which allows the rapid development of new modules. A big effort in the development of the code has been made in optimization of the linking efficiency, in order to minimize computational time. Finally, EDI is a multiplatform (Linux, Mac OS X) software.

Key words: electric discharges, plasma applications, plasma simulation

1. Introduction

Particle-in-cell (PIC) codes are an effective tool for modelling collisionless plasmas. For collisional plasmas, the most widely used approach is the fluid one (Bukowski, Graves & Vitello 1996; Magarotto *et al.* 2016). Over the past few decades, PIC codes have proven to be a very reliable and successful method for kinetic plasma simulations. In PIC simulations individual particles are tracked in a Lagrangian frame in a continuous phase space, whereas moments of the distribution such as densities

† Email address for correspondence: magimir91@gmail.com

and currents are computed simultaneously on Eulerian (stationary) mesh points. The roots of the PIC method go back to the work performed by Buneman (1959). He simulated an electrostatic plasma in one dimension and showed that particles codes could be used to study the linear, nonlinear and saturation phases of instabilities. The PIC scheme was formalized and codified during the 1980s by Birdsall (Birdsall & Langdon 2004) and Hockney (Hockney & Eastwood 1988). Their texts remain prominent to the present. Today the latest generation of PIC algorithms consist of large versatile codes. We can quote solvers which can treat problems in one dimension (1-D) (Dawson 1962), in two dimensions (2-D) (Lapenta, Iinoya & Brackbill 1995; Verboncoeur, Langdon & Gladd 1995; Jacobs & Hesthaven 2006, 2009) and in three dimensions (3-D) (Fonseca *et al.* 2002; Spitkovsky 2005; Melzani *et al.* 2013). In addition, the versatile code VORPAL can treat 1-D, 2-D and 3D problems depending on the accuracy required (Nieter & Cary 2004). Different numerical approaches have been adopted in previous works (Verboncoeur 2005). Charge weighting on a grid has been done with: the nearest grid point method (NGP), the cloud in cell method (CIC) or high-order charge-conserving algorithms (Moon, Teixeira & Omelchenko 2015). The grid itself can also be structured or unstructured (Jacobs, Kopriva & Mashayek 2001; Spirkin 2006). Particles are usually updated using the Boris leapfrog scheme. Depending on the type of field equation that is solved, it is possible to divide PIC codes in two groups: electrostatic and electromagnetic. While in the first group Poisson's equation is usually solved (with different charge-conserving schemes as reported in Villasenor & Buneman 1992; Umeda *et al.* 2003; Carlsson, Manente & Pavarin 2009), in the second group the full set of Maxwell's equations including both plasma source terms (charge density and plasma currents) and external source terms (e.g. a polarized electrode) are solved. These equations can be analysed using different numerical schemes such as: finite difference methods (FDM), Fourier transform (FT) methods or finite element methods (FEM) (Jacobs & Hesthaven 2006). In order to use PIC simulation for modelling collisional plasmas, special parts of the numerical scheme can also differ (Ren *et al.* 2015) and usually an interaction stage between charged and neutral particles (Vahedi & Surendra 1995) is added to the PIC loop. Nowadays PIC codes are used in different research areas, for example in laser-plasma simulations, in plasma acceleration and also in space applications (Lapenta *et al.* 1995). In particular, during the European HPH.COM program, at the University of Padova different codes were developed (Pavarin *et al.* 2011; Manzolaro *et al.* 2012; Cardinali *et al.* 2014; Fabris *et al.* 2015), for the detailed design and optimization of plasma thrusters of the helicon type (Manente *et al.* 2019). In order to obtain a unique and global tool able to simulate the whole plasma discharge we have developed a new two-dimensional (2-D) electrostatic PIC code named EDI (enhanced biDimensional pIc). EDI has been developed following the issues of: (i) flexibility, (ii) extensibility and (iii) efficiency that should be addressed with any PIC code (Verboncoeur *et al.* 1995). EDI is a 2-D PIC code that assumes an isotropic distribution of the plasma in the direction perpendicular to the simulation domain. The new code is able to reproduce accurately the physics of the phenomena of interest, and its C structure allows for a strong extensibility and reusability in order to add new models or to modify the existing ones. If compared with other PIC codes, like the one previously quoted, EDI incorporates also an external versatile open source solver called GetDP (<http://getdp.info>) (Dular & Geuzaine 2016), which has been modified in order to be properly integrated into EDI. The major advantage of using an external solver like GetDP directly incorporated into the EDI structure is its strong flexibility, in fact it is possible to couple together different physical problems (electrostatic, magnetostatic,

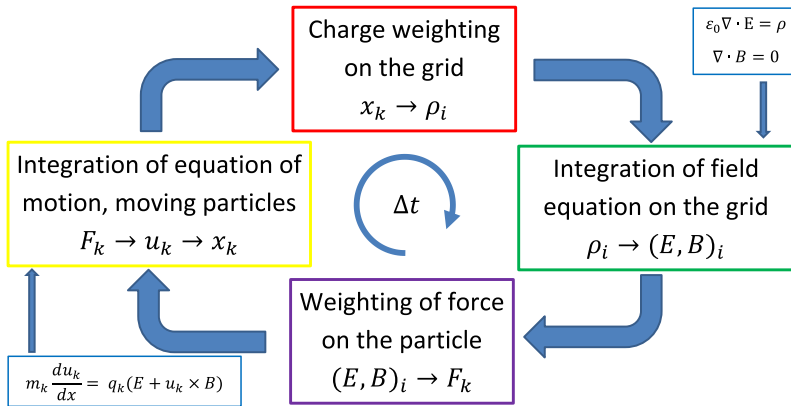


FIGURE 1. Scheme of EDI loop. Force F_k , velocity u_k and position x_k are referred to the particle k . Electric field E_i , magnetic field B_i and charge density ρ_i are evaluated at node i .

etc.) as well as to use different numerical methods (FEM, integral methods, etc.). In addition it is also possible to study problems with different formulations like static, transient or harmonic. In this paper we will focus on the electrostatic–magnetostatic version of EDI: a future work will be done to introduce in EDI a fully electromagnetic implementation. In § 2 we present the new code and its structure. In § 3, we analyse a set of test problems to prove how EDI is able to handle different physical situations and to reproduce precise analytical results. The first test (§ 3.1) is devoted to checking the correct integration of GetDP in EDI. The magnetostatic field computed by EDI (more precisely by GetDP integrated into EDI) has been compared against the results of an external solver, namely the open source code FEMM (finite element method magnetics) (<http://www.femm.info/wiki/HomePage>) (Meeker 2014). The second test (§ 3.2) is conceived to study the fluctuation spectra of a cold plasma. The last test (§ 3.3) consists in analysing the formation of the two stream instability, in particular we have compared the expected linear growth rate of the instability with the simulated one. Finally, in § 4 conclusions are drawn.

2. Physical model and numerical implementation

This section presents the numerical scheme implemented in EDI. EDI is an electrostatic–magnetostatic PIC code that uses an unstructured mesh of triangles. This allows us to manage arbitrary geometries. In the PIC scheme particles are defined in continuum space, while fields are defined at discrete locations in space. However, both fields and particles are defined at discrete times. In figure 1 we have proposed the general flow of the EDI code. Charge densities are calculated at grid points using a charge-conserving weighting scheme. Fields are computed using a FEM solver and are then interpolated onto particle positions with the same weighting scheme used to compute charge densities. Finally, particles are updated using a Boris leapfrog scheme and a large set of information (like particles velocities and positions) is saved.

2.1. The PIC method

The code simulates the evolution of charged particles due to their interaction with electromagnetic fields. The real plasma systems often contain an extremely large

number of particles. In numerical simulations super-particles are used to reduce the computational cost. A super-particle (or macroparticle) is a computational particle representing many real particles, it may be millions of electrons or ions in the case of a plasma simulation (Fehske, Schneider & Weiße 2007). We will indicate with p the number of real particles associated with a simulated super-particle. In such a way the mass and charge of the super-particle (labelled with sp) is $m_{sp} = p \cdot m$ and $q_{sp} = p \cdot q$ where q and m are the charge and mass of each real particle in the system. To avoid complicated equations in EDI the number p is fixed for all super-particles of a given species at all simulation times. The equations governing the super-particle dynamics are the equation of motion with the Lorentz force

$$m_{sp} \frac{d\mathbf{v}}{dt} = q_{sp}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.1)$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \quad (2.2)$$

and Maxwell's equations

$$\frac{d\mathbf{E}}{dt} = c^2 \nabla \times \mathbf{B} - \frac{\mathbf{J}}{\epsilon_0} \quad \frac{d\mathbf{B}}{dt} = -\nabla \times \mathbf{E} \quad (2.3a,b)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \nabla \cdot \mathbf{B} = 0, \quad (2.4a,b)$$

where \mathbf{E} and \mathbf{B} are the electric and magnetic fields. EDI is an electrostatic–magnetostatic code, for this reason we solve only (2.4) using a FEM formulation. For more details on the FEM implementation see § 2.5.

2.2. Spatial and temporal resolution

The mesh size is determined by the mean edge of the triangular mesh a . In what follows we will indicate with n_e^0 the plasma electron reference density and with the superscript 0 all the quantities based on it. For example $d_e^0 = c/\omega_{pe}^0$ is the reference electron skin depth ($\omega_{pe}^0 = \sqrt{n_e^0 q_e^2 / \epsilon_0 m_e}$ is the reference plasma pulsation) and $\lambda_D^0 = \sqrt{\epsilon_0 k_B T_e / n_e^0 q_e^2}$ is the reference Debye length. Using these two quantities, it is possible to introduce two indicators to measure the spatial resolution of the code (Melzani *et al.* 2013): $n_a = d_e^0/a$ and $n_\lambda = \lambda_D^0/a$. A similar analysis can be performed also for the temporal scale of the code that is instead defined starting from the global simulation time step Δt . In fact, if we define with $T_{pe}^0 = 2\pi/\omega_{pe}^0$ the reference electron plasma period, then the temporal resolution of the code is given by the quantity $n_t = T_{pe}^0/\Delta t$.

2.3. The mesh management

The code uses an unstructured mesh of triangles. The Delaunay mesh is built using an open-source software called Gmsh (<http://gmsh.info>) (Geuzaine & Remacle 2009). Gmsh is a free 2-D/3-D finite element grid generator with a build-in computer-aided drafting (CAD) engine and post-processor. The final *.msh* ASCII output file contains the mesh nodes, the elements, the region names and so on. In EDI we have included a parser able to import the *.msh* file and to construct the connectivity table of the mesh. This table is constructed by progressively adding mesh elements starting at the boundaries. This iteration results in a propagation of a front (internal boundary) between the meshed and the unmeshed region. For each node the parser computes also

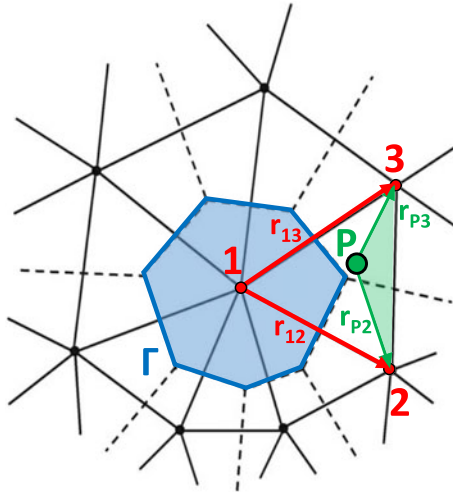


FIGURE 2. The parameter Γ is the area of the Voronoy cell associated with node 1. The particle P is located inside the triangular mesh element of vertices 1, 2 and 3. The vectors \mathbf{r}_{12} and \mathbf{r}_{13} are employed to calculate the surface of the triangular mesh element Ω_{123} , while the vectors \mathbf{r}_{P2} and \mathbf{r}_{P3} are used for the surface Ω_{P23} (depicted in green), see (2.8). Figure from Spirkin (2006).

a list of the barycentres of the triangles that include the node. This list is then used to compute the area Γ of the Voronoy cell associated with the node (see figure 2). Each area is computed using an open source software called Qhull (<http://www.qhull.org>) (Barber *et al.* 1996).

2.4. Weighting scheme (charge deposition)

In §2.1 we have described the concept of super-particle used in PIC simulations. In order to compute the nodal charge density it is necessary to introduce a function designated as the particle shape or a weighting function. We have followed the approach proposed by Hockney & Eastwood (1988), with particular attention paid to the analysis performed by Spirkin (2006), who describes this process by using a function that will be denoted in the following with W . The area of overlap between the particle shape and the grid cell determines the charge assigned to the grid point. We will call $W_I(\mathbf{r}_P)$ the fraction of a charge from a particle P located at $\mathbf{r}_P = (x_p, y_p)$ assigned to the grid point I located at $\mathbf{r}_I = (x_I, y_I)$. Charge conservation requires that

$$\sum_{I=1}^M W_I(\mathbf{r}_P) = 1, \tag{2.5}$$

where the sum is taken over M nodes used to distribute the charge close to the particle position. If we consider a 2-D domain discretized by an unstructured Delauney grid, each triangular cell is formed by three nodes. Consider now a particle P located inside the element identified by nodes 1, 2 and 3 whose positions are respectively $\mathbf{r}_1 = (x_1, y_1)$, $\mathbf{r}_2 = (x_2, y_2)$ and $\mathbf{r}_3 = (x_3, y_3)$ (see figure 2). We assign the charge of the

particle P to these three nodes. In this case, charge conservation requires that

$$\sum_{l=1}^3 W_l(\mathbf{r}_p) = 1, \tag{2.6}$$

where W_l is defined as

$$\left. \begin{aligned} W_1 &= \frac{\Omega_{P23}}{\Omega_{123}} \\ W_2 &= \frac{\Omega_{P31}}{\Omega_{123}} \\ W_3 &= \frac{\Omega_{P12}}{\Omega_{123}} \end{aligned} \right\} \tag{2.7}$$

In particular

$$\left. \begin{aligned} \Omega_{123} &= \frac{1}{2} |\mathbf{r}_{13} \times \mathbf{r}_{23}| \\ \Omega_{P23} &= \frac{1}{2} |\mathbf{r}_{P2} \times \mathbf{r}_{P3}| \end{aligned} \right\} \tag{2.8}$$

where $\mathbf{r}_{P2} = \mathbf{r}_2 - \mathbf{r}_P$, $\mathbf{r}_{P3} = \mathbf{r}_3 - \mathbf{r}_P$, $\mathbf{r}_{12} = \mathbf{r}_2 - \mathbf{r}_1$ and $\mathbf{r}_{13} = \mathbf{r}_3 - \mathbf{r}_1$. It is worth recalling that: (i) Ω_{123} is equal to the surface of the triangular mesh element, and (ii) Ω_{P23} to the surface delimited by the particle P and the nodes 2 and 3 (depicted in green in figure 2). The quantities Ω_{P31} and Ω_{P12} are computed in a similar manner. The number density and charge density due to N_s particles of species s (with charge q_s) at a fixed node (e.g. node 1) with coordinates $\mathbf{r}_1 = (x_1, y_1)$ can be evaluated as

$$n_s(x_1, y_1) = \frac{1}{\Gamma} \sum_{p=1}^{N_s} W_1(x_p, y_p) \tag{2.9}$$

$$\rho_s(x_1, y_1) = \frac{q_s}{\Gamma} \sum_{p=1}^{N_s} W_1(x_p, y_p), \tag{2.10}$$

where Γ is the area of the Voronoy cell where the particle is located (see figure 2).

Finally, it is worth recalling that the weighting scheme adopted does not require particular adjustments if the mesh is non-uniform. In fact each particle deposits charge only on the nodes of the element in which it is located. This approach is physically meaningful if the size of the mesh element is of the order of the Debye length in all of the domain; namely, the finer mesh where the plasma is denser (Birdsall & Langdon 2004).

2.5. The field FEM formulation

In EDI the fields are computed, at each time step, using a FEM formulation. EDI is an electrostatic–magnetostatic code, namely only equations (2.4) are solved among Maxwell’s set. In order to define a FEM approximation of equations (2.4), a weak formulation has been adopted (Bossavit 1998; Polycarpou 2005). We assume that the problem is defined in a region Ω which is an open, polygonal and simply connected set while $\partial\Omega$ denotes its frontier (Pinto *et al.* 2008):

$$\left. \begin{aligned} H_1(\Omega) &= \{v \in L^2(\Omega) : \nabla v \in L^2(\Omega, \mathbb{R}^n)\} \\ H_0^1(\Omega) &= \{v \in H_1(\Omega) : v|_{\partial\Omega} = 0\} \end{aligned} \right\} \tag{2.11}$$

for more details, see Girault & Raviart (2012). The electrostatic problem has been formulated in terms of Poisson’s equation

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \rightarrow \nabla \cdot (\epsilon_0 \nabla \phi) = -\rho, \tag{2.12}$$

where ϕ is the unknown electrostatic potential. The weak formulation reads (Bossavit 1998; Pinto *et al.* 2008)

$$\int_{\Omega} \epsilon_0 \nabla \phi \cdot \nabla \phi' \, d\Omega = \int_{\Omega} \rho \phi' \, d\Omega \quad \forall \phi' \in H_0^1(\Omega), \tag{2.13}$$

where ϕ' is a scalar test function. For the magnetostatic problem, we have assumed that the static magnetic fields is generated by magnets. Therefore we have solved the following problem using a \mathbf{H} conforming formulation (Dular *et al.* 1997)

$$\nabla \cdot \mathbf{B} = 0. \tag{2.14}$$

The weak formulation reads (Dular & Geuzaine 2016)

$$\int_{\Omega} \mu(\mathbf{H}_s - \nabla \zeta) \cdot \nabla \phi' \, d\Omega = 0 \quad \forall \phi' \in H_0^1(\Omega), \tag{2.15}$$

where \mathbf{H}_s is the source field and ζ is the scalar unknown potential for which $\mathbf{H} = -\nabla \zeta$. If we consider a permanent magnet we can assume that the magnetic field has the same value in all the internal points of the magnet so we can write

$$\mathbf{H}_s = \mathbf{H}_c(\mathbf{x}) \quad \forall \mathbf{x} \in \text{Magnets}, \tag{2.16}$$

where \mathbf{H}_c is the coercive magnetic field of the magnet assumed fixed and measured in A m^{-1} . Equations (2.13)–(2.15) are the weak formulation of (2.4) and can be solved with any FEM solver once appropriate boundary conditions are defined (Dirichlet in our case). In the current implementation of EDI, we managed to use the open source software GetDP (a ‘General environment for the treatment of Discrete Problems’) (Geuzaine 2007). GetDP is a well-established scientific software environment for the numerical solution of integro-differential equations. In particular, its capability in solving electro-magnetic (EM) problems has been proven both in the industrial (Trophime *et al.* 2002; Sabariego *et al.* 2004) and the plasma physics (Damyanova, Sabchevski & Zhelyazkov 2010) fields. In the current implementation, GetDP has been properly modified in order to accept EDI input parameters (e.g. nodal plasma density and electron temperature) and to solve the electrostatic/magnetostatic problem. The final GetDP outputs are the \mathbf{E} and \mathbf{B} fields at EDI nodes. From a numerical standpoint, the stationary equations (2.13)–(2.15) are solved separately at each time step in order to grasp respectively the ϕ and ζ potentials. On the triangular elements of the mesh, the potential field can be represented as a third-order polynomial of the form (Dular & Geuzaine 2016)

$$\phi(\mathbf{x}) = \sum_{j=1}^3 \phi_j L_j(\mathbf{x}), \tag{2.17}$$

where \mathbf{x} is a generic position, ϕ_j is the scalar potential in the node j of a specific triangular element and L_j is the basis function (specifically, a Lagrange polynomial)

associated with the same node. An equation similar to (2.17), holds true also for the ζ potential. In particular, in the GetDP environment, equation (2.17) is implemented setting the FunctionSpace as 0-form type and the basis function as BF-Node (Dular & Geuzaine 2016). Once the basis functions are defined (see (2.17)), the governing equations expressed in the weak form (see (2.13)–(2.15)) can be reduced to algebraic linear systems thanks to the Galerkin method (Quarteroni, Sacco & Saleri 2010). The latter is implemented in GetDP setting the problem Formulation as FemEquation (Dular & Geuzaine 2016). Finally, the two resulting linear systems are solved with the lower-upper (LU) decomposition algorithm implemented in the PETSc library (Balay 2001).

2.6. Particle initialization

In EDI, particles of each species can be initially loaded into the whole simulation domain or into a fixed region. The region in which particles are loaded will be called in the following the ‘loading region’. If the mesh and the plasma are uniform, each particle to be loaded is attributed to a randomly chosen triangle. Otherwise, the same procedure is repeated associating different probabilities to each mesh element depending on: (i) the density imposed at its barycentre, (ii) its surface. Anyway, let us assume that a particle has to be loaded on the triangle whose vertices are $\mathbf{A} = (x_A, y_A)$, $\mathbf{B} = (x_B, y_B)$ and $\mathbf{C} = (x_C, y_C)$. We position the particle at a point $\mathbf{P} = (x_P, y_P)$ on the triangle surface by generating two random numbers r_1 and r_2 between 0 and 1 and solving the following equation (Osada *et al.* 2002)

$$\mathbf{P} = (1 - \sqrt{r_1})\mathbf{A} + \sqrt{r_1}(1 - r_2)\mathbf{B} + \sqrt{r_1}r_2\mathbf{C}. \quad (2.18)$$

Equation (2.18) gives a uniform random point distribution with respect to the surface of the chosen triangle.

In addition, several momentum and energy distributions are available that can be used to initialize the energy and the momentum of the particles in the loading region. In this paper we have used two types of loading distribution. A Maxwell–Boltzmann distribution in energy

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-E)^2/2\sigma^2}, \quad (2.19)$$

with expectation value E and variance σ^2 . A convolution of two Gaussian distributions (Hu 2016) along the v_x velocity component for the two stream instability test (§ 3.3)

$$f(v_x, v_y) = \frac{1}{2} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-(v_x+v_D)^2/2\sigma^2} + \frac{1}{\sqrt{2\pi}\sigma} e^{-(v_x-v_D)^2/2\sigma^2} \right) \delta(v_y), \quad (2.20)$$

with expectation value for each Gaussian function equal to $\pm v_D$ and variance σ^2 ; $\delta(x)$ is the Dirac delta function, defined by $\delta(x) = \infty$ for $x=0$ and $\int_{-\infty}^{\infty} \delta(x) dx = 1$. Usually σ matches with v_{th} , the thermal velocity of each species. For the two stream instability test $\sigma \rightarrow 0$. From this it follows that (2.20) can be rewritten as

$$f(v_x, v_y) = \frac{1}{2} [\delta(v_x + v_D) + \delta(v_x - v_D)] \delta(v_y). \quad (2.21)$$

In the present code, in order to generate particles according to the distribution functions described in (2.19) and (2.20) we used the method called inversion of cumulative distribution functions. See D’Agostini (2003) for more details.

2.7. Particle updating

The particle motion is described in (2.1) and (2.2). Equation (2.2) can be discretized in three dimensions after Umeda (2003) and Umeda *et al.* (2003) as

$$x^{t+\Delta t} = x^t + v_x^{t+(\Delta t/2)} \Delta t \tag{2.22}$$

$$y^{t+\Delta t} = y^t + v_y^{t+(\Delta t/2)} \Delta t \tag{2.23}$$

$$z^{t+\Delta t} = z^t + v_z^{t+(\Delta t/2)} \Delta t \tag{2.24}$$

and (2.1) as

$$\mathbf{v}^{t+(\Delta t/2)} = \mathbf{v}^{t-(\Delta t/2)} + \Delta t \frac{q_{sp}}{m_{sp}} (\mathbf{E}^t + \mathbf{v}^t \times \mathbf{B}^t). \tag{2.25}$$

Since \mathbf{v}^t is not known *a priori*, equation (2.25) can be rewritten as

$$\mathbf{v}^{t+(\Delta t/2)} = \mathbf{v}^{t-(\Delta t/2)} + \Delta t \frac{q_{sp}}{m_{sp}} \left(\mathbf{E}^t + \frac{\mathbf{v}^{t+(\Delta t/2)} + \mathbf{v}^{t-(\Delta t/2)}}{2} \times \mathbf{B}^t \right). \tag{2.26}$$

Using (2.26) and following the approach proposed in Birdsall & Langdon (2004) and Hockney & Eastwood (1988), it is possible to compute $\mathbf{v}^{t+(\Delta t/2)}$ using a four step computation usually known as the ‘Buneman–Boris method’. At the first time step the velocity is pushed back to the time $t - \Delta t/2$. In addition to the just exposed algorithm, in EDI we have introduced other integration schemes, like the relativistic Vay algorithm (Vay 2008). However in the simulation used to benchmark the code we have used only the just exposed integrator, so that we will not discuss the other integration schemes.

2.8. Particle tracking

After each time step, particles are tracked in the unstructured triangle grid using an efficient and robust particle-localization algorithm. Given the previous particle position and the cell containing that position, the algorithm determines the cell which contains a nearby position. The algorithm is based on tracking a particle along its trajectory by computing the intersections of the trajectory and the cell faces and it is a modified version of the one proposed in Haselbacher, Najjar & Ferry (2007).

2.9. Fields and particle boundaries

In EDI we have introduced different fields and particles boundary conditions. For the particles we have introduced conductive and reflective boundaries. A secondary boundary electron emission subroutine is now being developed using the approach proposed in Seiler (1983) and Taccogna, Longo & Capitelli (2004) and it will be added to the particle boundary management. Also, fields have different types of boundaries. These could be periodic (field quantities at one boundary are equal to those at the other boundary), Dirichlet or Neumann boundary conditions depending on the material types and on the chosen FEM implementation (Bossavit 1998).

2.10. The main loop

In EDI we have defined a full integer time step $n\Delta t$ and a half-integer time step $(n + 1/2)\Delta t$. As already stated, fields are computed using a FEM formulation, for this reason these are known at integer time steps i.e. \mathbf{E}^t and $\mathbf{B}^t \quad \forall t$. The particle position \mathbf{r} is known at the full integer time step, and velocities \mathbf{v} at the half-integer time step. As initial condition, we provide $\mathbf{r}^t, \mathbf{v}^{t-(\Delta t/2)}, \mathbf{E}^t, \mathbf{B}^t$. The structure of the main loop is the following:

- (i) update particle velocity from $t - \Delta t/2$ to $t + \Delta t/2$: $\mathbf{v}^{t+(\Delta t/2)}$;
- (ii) update the position \mathbf{r} from t to $t + \Delta t$ using $\mathbf{v}^{t+(\Delta t/2)}$;
- (iii) particle tracking;
- (iv) boundary management for particles;
- (v) test exit condition;
- (vi) particle source;
- (vii) weighting scheme (charge deposition);
- (viii) advance electric and magnetic fields from t to $t + \Delta t$ with the FEM solver.

2.11. Particle source

At the end of each main loop of the code (steps 1–6 in the list of § 2.10), if needed, a particle source inserts a fixed number of particles into the system. These particles can be introduced into a fixed simulation region or into the whole simulation domain according to the requirements. The particle source inserts the particles using the same spatial and energy distribution functions already presented in § 2.6.

2.12. Time loop termination

The main time loop is terminated once a user-specified exit condition is satisfied. EDI supports two conditions (Brieda 2005):

- (i) maximum number of time steps;
- (ii) steady state.

A simulation is assumed to be at the steady state if the number of particles leaving through the external boundaries matches the number of new particles introduced into the system by the source. This condition shall be met for a sufficiently high number of time steps (m) in order to avoid a ‘false-positive’ induced by PIC noise. In particular, the steady-state condition is implemented by requiring that:

$$\frac{1}{m+1} \sum_{k=h-m}^h \frac{|N_{sp}^k - N_{sp}^{k-1}|}{N_{sp}^k} < \nu, \quad (2.27)$$

where ν is a user defined number that represents a steady-state condition (usually $\nu \sim 10^{-4}$), N_{sp}^k is the number of super-particles in the system at the time step k , and h is the current time step. As a rule of thumb, we can assume $m \sim 50$.

2.13. Output and data analysis

At the end of each time step, simulation results are saved in different file formats. EDI provides two types of output file: *.dat* files, with particle positions and velocities, and/or *.pos* files (Gmsh format). The latter is useful to visualize, using Gmsh, the evolution of quantities of interest like charge density. A DUMP file is also provided. This DUMP file produces a binary file that is used to save, at a user predefined time step, the configuration of the electromagnetic fields and/or particles velocities and positions. A parser is also provided to import the DUMP file and to load it as initial condition. These data can be used to start the simulation with a prefixed configuration i.e. a predefined configuration of fields and particles.

3. Code validation

In this section, we will discuss three tests performed to benchmark the new EDI PIC code. The first test (§ 3.1) is devoted to checking the correct integration of GetDP on EDI. The magnetostatic field computed by EDI (more precisely by GetDP integrated on EDI) has been compared against the results of an external solver, namely the open source code FEMM (Meeker 2014). It is worth highlighting that the aim of this test is not to evaluate the numerical model implemented on GetDP, but rather it is intended to validate the information exchange (namely, values of the EM fields at nodes) between GetDP and EDI. Subsequently we have analysed some classical plasma models to prove how EDI is able to handle different physical problems and to reproduce precise analytical results. The second test (§ 3.2) consists of studying the oscillation modes of a cold plasma by means of a FT analysis. The third and last tests (§ 3.3) consist in analysing the two stream instability formation, paying particular attention to the linear stage of this instability.

3.1. GetDP integration on EDI

As previously stated, in EDI the EM fields are calculated by the external solver GetDP. The latter has been properly modified in respect of classical formulations (e.g. Trophime *et al.* 2002; Sabariego *et al.* 2004) in order to provide to EDI the EM fields at the mesh nodes. In order to check (i) the implementation of (2.13)–(2.15) on GetDP, and (ii) the correct integration of GetDP on EDI, the results of GetDP-EDI have been benchmarked against FEMM (Meeker 2014). In particular, the two solvers have been exploited to calculate the magnetostatic field generated by two permanent magnets made of SmCo immersed in a vacuum region (magnetic permeability outside the magnets equal to the vacuum permittivity μ_0) and in the absence of plasma. The benchmark magnetic field computed with FEMM is reproduced in figure 3(a). We have defined a line along which we have compared the magnetic field calculated with the two solvers (see the red line in figure 3a at $y = -2.5$ mm). The comparison is reproduced in figure 3(b–d). The greater noise of the EDI curve in the position closer to the magnets is mainly due to the interpolation that was used to compute the magnetic field at the same mesh points used by FEMM. Nonetheless, the agreement between the two codes is satisfactory and we can conclude that the implementation of the equations in GetDP, and in particular their integration in EDI, are correct.

It is worth underlining that the FEM formulation implemented on GetDP has been evaluated with other tests where: (i) the capability of EDI in reproducing analytical results has been tested, and in turn the accuracy of the EM solver can be checked (see §§ 3.2 and 3.3), and (ii) the computational time required by each operation of EDI, along with how it scales with the size of the mesh and time step, has been investigated in a convergence study (see § 3.3).

3.2. Oscillation mode analysis

The second test consists of simulating the oscillation modes of a cold plasma. In particular we have analysed only the evolution of electrons in a fixed background of ions. Generally speaking, the wave spectrum can be analysed by means of the FT of the velocity field, in particular

$$F_x(\mathbf{k}, \omega) = \iint v_x(\mathbf{x}, t) e^{-i(\mathbf{k}\cdot\mathbf{x} + \omega t)} dV dt, \quad (3.1)$$

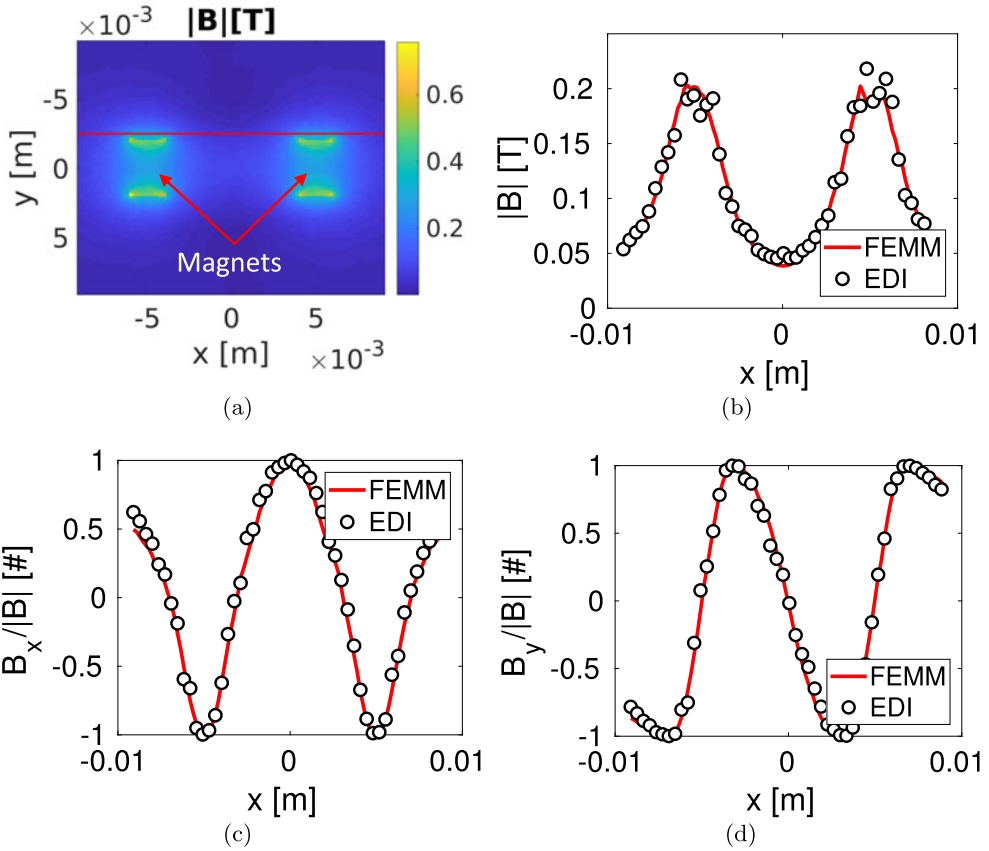


FIGURE 3. Comparison between the magnetic fields \mathbf{B} computed by EDI and FEMM: (a) intensity of the reference field computed with FEMM; (b) comparison between $|\mathbf{B}|$; (c) comparison between $B_x/|\mathbf{B}|$; (d) comparison between $B_y/|\mathbf{B}|$. The comparisons are performed along $y = -2.5$ mm (see red line in a).

where F_x is the FT of the component along the x axis of velocity (v_x), \mathbf{k} is the wavevector (Inan & Gołkowski 2010), \mathbf{x} is a generic position vector in the domain, ω is the frequency, t is the time and i is an imaginary constant. Similarly, F_y is defined as the FT of the component along the y axis of the velocity (v_y). To keep the analysis lucid, we have studied only the frequency modes of zero wavevector, $\mathbf{k} = 0$ (Melzani *et al.* 2013). In such a way, equation (3.1) reduces to

$$F_x(\mathbf{0}, \omega) = \int \left(\int v_x(\mathbf{x}, t) dV \right) e^{-i\omega t} dt = \int q_x(t) e^{-i\omega t} dt, \quad (3.2)$$

where q_x is the volume integral of v_x . Similarly, q_y can be defined with respect to v_y . Therefore, equation (3.2) can be discretized by means of a fast Fourier transform (FFT) as (Melzani *et al.* 2013)

$$F_x(\omega) \approx \sum_{k=1}^{N_t} \left(\sum_{i=1}^{N_{sp}} v_{ix}(t_k) \right) e^{-i\omega t_k}, \quad (3.3)$$

| | | |
|-------|-------|-------------|
| n_t | n_a | n_λ |
| 200 | 5499 | 3 |

TABLE 1. Simulation parameters (see § 2.2) adopted during the oscillation mode analysis.

where N_t is the total number of integration time steps, t_k is the time at the time step k , N_{sp} is the number of electron super-particles and v_{ix} is the component along the x axis of the velocity of the super-particle i . In particular, also F_y can be discretized in the same manner as described in (3.3). It is worth highlighting that in (3.3), the q_x term has been approximated as (Melzani *et al.* 2013)

$$q_x \approx \sum_{i=1}^{N_{sp}} v_{ix} \tag{3.4}$$

provided that: (i) with the weighting scheme adopted (see § 2.4), each super-particle is assumed to have the size of the triangular mesh element inside which it is located (Spirkin 2006), and (ii) for this test, we have adopted a uniform mesh. Simulation parameters (see § 2.2) have been chosen in accordance with table 1. The simulation domain is a square box of side $L = 2$ cm aligned with the x and y axes of a Cartesian reference system. Particles are inserted into the domain with the distribution function described in (2.19). Reflecting boundaries were chosen for particles.

We have analysed the electrostatic electron fluctuations with and without the presence of an external background magnetic field \mathbf{B}_0 perpendicular to the simulation domain, namely aligned along the z axis. In the absence of a background magnetic field (see § 3.2.1), electrons perform Langmuir oscillations with a characteristic frequency equal to the plasma frequency ω_{pe} . Provided that these oscillations involve the creation of electric fields by local charge imbalance, this first analysis investigates the capability of the code to reproduce the field propagation, as well as particle motion (Melzani *et al.* 2013). When the magnetic field is present (see § 3.2.2) and when this field is high enough, the Langmuir motion is fully covered by the Larmor motion in a plane perpendicular to \mathbf{B}_0 with a characteristic frequency equal to $\omega_c = q_e |\mathbf{B}_0| / m_e$. This example then probes the accuracy of the particle motion integrator (Melzani *et al.* 2013). Moreover, in an intermediate regime, electrons oscillate with a characteristic frequency equal to the upper hybrid frequency $\omega_h^2 = \omega_{pe}^2 + \omega_c^2$.

3.2.1. Absence of a background magnetic field

We have simulated a plasma initially at rest, in the absence of an external magnetic or electric field. The results of the frequency analysis are reported in figure 4. The evolution of the quantities q_x and q_y (see (3.4)) at time t has been depicted in figure 4(a). Instead, in figure 4(b), F_x and F_y (namely the FT of respectively q_x and q_y) are reported as a function of the frequency ω . The global simulation time T is 20 times a plasma oscillation period, so the frequency resolution is $\Delta\omega = 2\pi/T = 0.05\omega_{pe}$. The FT (see figure 4b) shows that the main contribution to the spectrum is given by a single frequency close to the plasma frequency; nonetheless the results along the x direction are worse with respect to the y axis. In addition, a satisfactory good agreement can be found comparing the value of the frequency for which FT is maximum (respectively ω_x^{SIM} and ω_y^{SIM}) against the expected value ω_{pe} (error of the

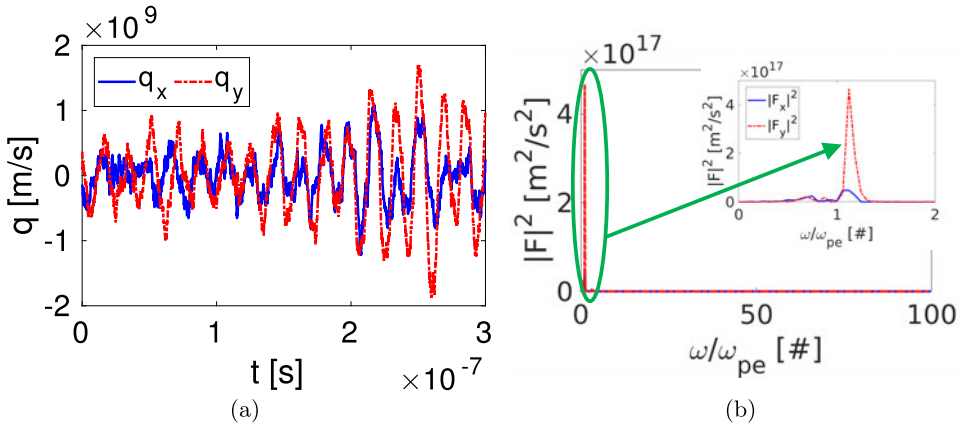


FIGURE 4. In the absence of a magnetic field: (a) q_x and q_y (see (3.4)) versus the simulation time t ; (b) square moduli of the F_x and F_y quantities (see (3.3)), namely the FT of q_x and q_y , as a function of the frequency ω (normalized in respect to the plasma frequency ω_p).

| $\omega_x^{\text{SIM}}/\omega_{pe}$ | $\omega_y^{\text{SIM}}/\omega_{pe}$ |
|-------------------------------------|-------------------------------------|
| 1.059 | 1.122 |

TABLE 2. Frequency analysis in the absence of a background magnetic field. Ratio between the plasma frequency (ω_{pe}) and the principal oscillation frequencies calculated from the FT of the q_x and q_y quantities (see (3.4)), respectively ω_x^{SIM} and ω_y^{SIM} .

order of 10% as reported in table 2). In conclusion, the mild difference between the simulated frequencies and the theoretical one, along with the non-negligible noise in F_x , are mainly due to the finite simulation domain. In fact, the theoretical derivation of ω_{pe} assumes that we are working with a plasma that is infinite in extent (Chen 1984), while we have limited our analysis to a relatively small box ($L = 2$ cm) (Melzani *et al.* 2013). Therefore, we can conclude that both the field propagation, and particle motion are satisfactorily well described by the EDI code.

3.2.2. Presence of a background magnetic field

In presence of a uniform magnetic field perpendicular to the simulation domain, particles oscillate with a frequency equal to ω_h . However, if the magnetic field is high enough such that $\omega_c \gg \omega_{pe}$ then $\omega_h \sim \omega_c$. In our case, assuming $B_0 = 1.0$ T is enough. Working in these conditions, we have performed the same analysis as that in § 3.2.1 computing the frequency modes of the system. In figure 5(a) the quantities q_x and q_y (see (3.4)) are depicted as a function of the time t . In figure 5(b) the FTs of both q_x and q_y (namely F_x and F_y) are reported as a function of the frequency ω . The spectral resolution of this analysis is $\Delta\omega = 0.001\omega_c$. In this case the agreement between numerical and theoretical data is very good: the spectrum (see figure 5b) is peaked, along both x and y directions, at a frequency very close to the expected ω_c (see table 3). This is a confirmation of the capability of the code in reproducing the particle motion.

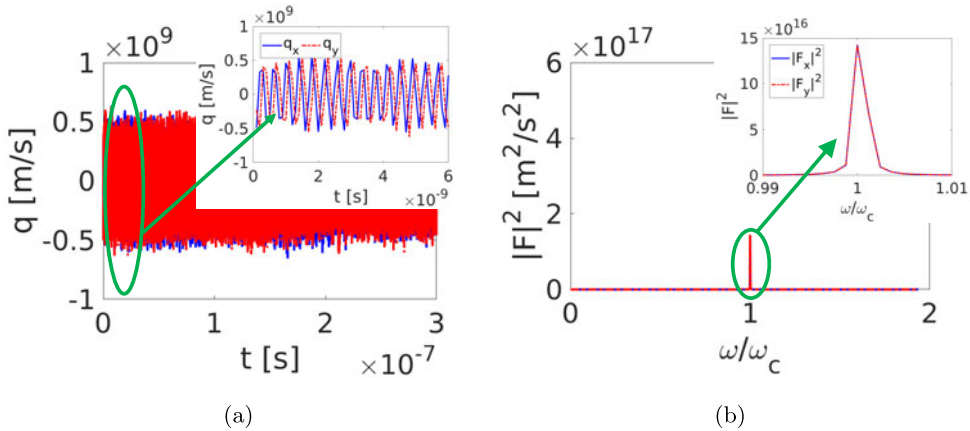


FIGURE 5. In the presence of magnetic field: (a) q_x and q_y (see (3.4)) versus the simulation time t ; (b) square moduli of the F_x and F_y quantities (see (3.3)) namely the FT of q_x and q_y , as a function of the frequency ω (normalized in respect to the cyclotron frequency ω_c).

| $\omega_x^{\text{SIM}}/\omega_c$ | $\omega_y^{\text{SIM}}/\omega_c$ |
|----------------------------------|----------------------------------|
| 1.00 | 1.00 |

TABLE 3. Frequency analysis in the presence of a background magnetic field. Ratio between the cyclotron frequency (ω_c) and the principal oscillation frequencies calculated from the FT of the q_x and q_y quantities (see (3.4)), respectively ω_x^{SIM} and ω_y^{SIM} .

3.3. Two stream instability test

In this section we explore the EDI results for the generation of electrostatic waves driven by two stream instability. These waves typically arise from the interaction between two symmetric counterstreaming particle beams, an energetic particle stream injected into a plasma or by setting a current along the plasma where different species can have different drift velocities. These phenomena are discussed in Chen (1984), Birdsall & Langdon (2004) and Ghorbanalilu, Abdollahzadeh & Rahbari (2014). A review on this subject is given in Anderson, Fedele & Lisak (2001).

3.3.1. Two stream instability formation

We have considered two counterstreaming electron beams with a constant background of ions. The system is initially described by an electron distribution function as reported in (2.21). The first stream travels in the x direction with a drift velocity $\mathbf{v}_D = v_D \hat{x}$ while the second one moves in the opposite direction with drift velocity $\mathbf{v}_D = -v_D \hat{x}$. If we assume $v_{th} \rightarrow 0$ for the loading electrons, i.e. $\sigma \rightarrow 0$ (see § 2.6), then each particle has exactly the stream velocity. If we look at periodic harmonic solutions for electric and magnetic fields with pulsation ω and wavevector \mathbf{k} (e.g. $\mathbf{E}(\mathbf{r}, t) = \mathbf{E}e^{i\mathbf{k}\cdot\mathbf{r} - i\omega t}$), the dispersion relation for longitudinal plasma waves propagating in the x direction ($\mathbf{k} = k\hat{x}$) in an electron gas described by the Vlasov

equation is (Lindman 1970; Bittencourt 2013)

$$1 = \frac{\omega_{pe}^2}{k^2} \int_v \frac{f_0(v)}{\left(v_x - \frac{\omega}{k}\right)^2} dv^2, \quad (3.5)$$

where $f_0(v)$ is the electron equilibrium distribution function. Assuming a distribution $f_0(v)$ equal to the one reported in (2.21), and substituting (2.21) in (3.5) yields

$$1 = \frac{1}{2} \omega_{pe}^2 \left[\frac{1}{(kv_D - \omega)^2} + \frac{1}{(kv_D + \omega)^2} \right]. \quad (3.6)$$

This is the dispersion relation for longitudinal waves in a counterstreaming electron plasma characterized by the distribution reported in (2.21). This equation can be rearranged as a quartic polynomial in ω (Bittencourt 2013) with two solutions called ω_1^2 and ω_2^2 . One of these two solutions (say ω_2^2) is a negative real quantity if

$$k^2 v_D^2 < \omega_{pe}^2. \quad (3.7)$$

It follows that ω_2 has two imaginary values

$$\omega_2 = \pm i\omega_{2i}, \quad (3.8)$$

where ω_{2i} is a real positive quantity. The positive imaginary value of ω_2 corresponds to an unstable mode and leads to a growth of instability, while a negative imaginary term in ω_2 leads to temporal decay of the wave amplitude (Landau damping). Solving (3.6), the positive imaginary value is (Bittencourt 2013)

$$\omega_{2i} = \left\{ -\left(\frac{1}{2}\omega_{pe}^2 + k^2 v_D^2\right) + \left[\left(\frac{1}{2}\omega_{pe}^2 + k^2 v_D^2\right)^2 - k^2 v_D^2(k^2 v_D^2 - \omega_{pe}^2)\right]^{1/2} \right\}^{1/2} \quad (3.9)$$

and it will be called the growth rate of the instability.

3.3.2. Growth rate, energy and momentum

To test EDI, we have computed the simulated growth rate ω_{2i} (see Lotov *et al.* 2015) and compared it with the theoretical growth rate obtained using (3.9). The procedure adopted for calculating ω_{2i} is described in the following. The energy associated with the electrostatic field W_E , at a fixed time t , can be computed as

$$W_E = \int_V \frac{1}{2} \epsilon_0 |\mathbf{E}(t)|^2 dV, \quad (3.10)$$

where \mathbf{E} is the electrostatic field, ϵ_0 is the vacuum permittivity and the integration is performed on the simulation volume. Equation (3.10) can be discretized as follows

$$W_E = \frac{1}{2} \epsilon_0 \sum_j |\mathbf{E}_j|^2 \Delta V_j, \quad (3.11)$$

where \mathbf{E}_j is the electric field computed at node j , ΔV_j is the covolume associated with node j and the sum is made over the number of nodes that define a discretization of the domain. If we consider periodic harmonic solutions for electric field and we

assume (3.8) for the unstable mode, it is possible to apply natural logarithm to (3.11) and rewrite it as

$$\ln(W_E) = 2\omega_{2i}t + h, \tag{3.12}$$

where h is a constant that is not time dependent. The growth rate can then be estimated as follows

$$\omega_{2i} = \frac{1}{2} \frac{\partial}{\partial t} \ln(W_E). \tag{3.13}$$

Another important quantity to monitor is the total energy W associated with the system. In fact PIC simulations can be affected by numerical heating (Markidis & Lapenta 2011), namely a non-physical growth of W . Usually this phenomenon can be reduced by a proper selection of mesh size and integration time step. We can consider the initial energy of the system as its kinetic energy W_{K0} , namely the sum of the kinetic energy of ions and electrons $W_{K0} \equiv W_K^{\text{elec}} + W_K^{\text{ion}}$. In this specific case, $W_{K0} \equiv W_K^{\text{elec}}$ having assumed fixed ions. Therefore

$$W_{K0} = \frac{1}{2} \sum_{i=1}^{N_{sp}} pm_e v_{i0}^2, \tag{3.14}$$

where N_{sp} is the number of electron super-particles in the system, v_{i0} is the speed of each super-particle at the initial time step, p is the number of particles associated with each super-particle and m_e is the electron mass. Consequently, the normalized kinetic energy at each time t can be computed as suggested by Hu (2016)

$$\bar{W}_K = \frac{\frac{1}{2} \sum_{i=1}^{N_{sp}} pm_e v_i^2}{\frac{1}{2} \sum_{i=1}^{N_{sp}} pm_e v_{i0}^2}, \tag{3.15}$$

where v_i is the speed of each super-particle at time t . Moreover, it is also possible to define the normalized field energy of the system at the time t as

$$\bar{W}_E = \frac{\frac{1}{2} \epsilon_0 \sum_j |\mathbf{E}_j|^2 \Delta V_j}{\frac{1}{2} \sum_{i=1}^{N_{sp}} pm_e v_{i0}^2}. \tag{3.16}$$

Therefore the normalized total energy, in the absence of a magnetic field, can be defined as

$$\bar{W} = \bar{W}_K + \bar{W}_E. \tag{3.17}$$

Finally, we have monitored the total momentum of the collection of particles \mathbf{Q} . In fact, due to the charge deposition algorithm that we have adopted (see § 2.4), particles might induce spurious forces on themselves unless the mesh is not rectangularly structured (Colella & Norgaard 2010). Therefore the conservation of total momentum

might fail due to this effect. At each time t , the two components of \mathbf{Q} , respectively Q_x along the x axis and Q_y along y , are defined as

$$Q_x = \sum_{i=1}^{N_{sp}} pm_e v_{ix} \quad (3.18)$$

$$Q_y = \sum_{i=1}^{N_{sp}} pm_e v_{iy}, \quad (3.19)$$

where v_{ix} and v_{iy} are the components, along the x and y axes, of the velocity of the super-particle i . Therefore, in order to quantify the effect of self-induced forces, a normalized total momentum has been defined as

$$\bar{\mathbf{Q}} = \frac{\mathbf{Q}}{Q_0}, \quad (3.20)$$

where Q_0 is a reference momentum $Q_0 = N_{sp} pm_e v_D$, and v_D is the drift velocity. It is worth recalling that the reference momentum has not been evaluated as the initial total momentum is $\mathbf{Q} \approx 0$ for $t=0$, provided that the loading distribution function has been assumed in accordance with (2.21).

3.3.3. Numerical analysis

As previously stated, the initial conditions of our simulation consist of two opposite electron beams with drift velocity v_D and a constant background of ions. More precisely, the initial distribution function f_0 is chosen in accordance with (2.21), where the drift velocity is $v_D = 1 \times 10^6 \text{ m s}^{-1}$. The simulation domain is a square aligned along the x and y directions, with edge $L = 2 \text{ cm}$. The dynamics of 5×10^5 super-particles has been followed; each super-particle is associated with 100 particles. Periodic boundary conditions, namely an electrostatic potential $\phi = 0$ on the sides of the square, have been adopted. During the simulation, the initial configuration is perturbed with a wave whose wavenumber is $k = 2\pi/L$ so that, thanks to the periodic boundary conditions, only one unstable mode is selected out of the continuous spectrum typical of the infinite plasma (Ghorbanalilu *et al.* 2014).

A convergence study of the numerical accuracy of EDI has been performed as the grid refinement and the integration time step are varied. The numerical accuracy has been evaluated in terms of the difference between the normalized total energy \bar{W} at the beginning of the simulation and that when the two stream instability is fully developed (i.e. $4 \times 10^{-8} \text{ s}$). In fact, if the grid size and time step are too coarse, PIC simulations can be affected by numerical heating (Markidis & Lapenta 2011). For this application, we have assumed that the total energy cannot vary of more than 2% because of the latter effect. The simulations have been performed on a machine equipped with an Intel R Core i7-6700HQ CPU 2.60 GHz \times 8, and 16 Gb of RAM. First the grid refinement effect has been analysed by adopting three different mesh configurations, namely mean edges of the triangular mesh equal to $a = 8 \times 10^{-4} \text{ m}$ (i.e. 2714 triangular elements), $a = 4 \times 10^{-4} \text{ m}$ (10491 elements) and $a = 2 \times 10^{-4} \text{ m}$ (41498 elements). In particular, these mesh configurations lead to a simulation parameter n_a ranging from 3700 up to 14800, and n_λ from 1.5 up to 5.0 (see § 2.2). In figure 6(a), the variation of the normalized total energy \bar{W} as a function of the time t is reported for the three meshes analysed and an integration time step $\Delta t = 1 \times 10^{-10} \text{ s}$. Reducing a leads to a reduction of the numerical heating

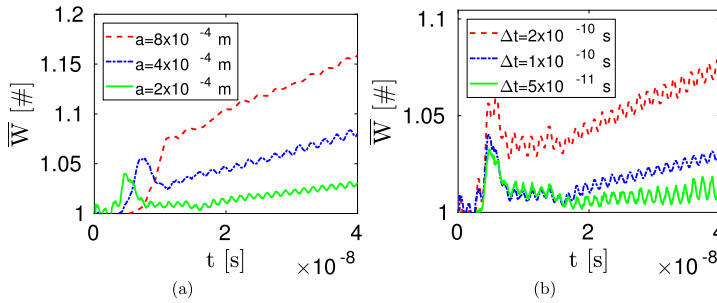


FIGURE 6. Normalized total energy \bar{W} (see (3.17)) plotted versus time t for: (a) different mean edge of the triangular mesh a and fixed integration time step $\Delta t = 1 \times 10^{-10}$ s; (b) different Δt and fixed $a = 2 \times 10^{-4}$ m.

| a (m) | Duration (s) | CD (%) | EM (%) | Int (%) | Track (%) |
|--------------------|--------------|--------|--------|---------|-----------|
| 8×10^{-4} | 315 | 8 | 15 | 9 | 68 |
| 4×10^{-4} | 1203 | 4 | 56 | 3 | 37 |
| 2×10^{-4} | 6180 | 2 | 85 | 1 | 12 |

TABLE 4. Timing analysis for different values of the mean edge of the triangular mesh a and fixed integration time step $\Delta t = 1 \times 10^{-11}$ s. Reported is the simulation duration, along with the percentage time required for accomplishing the charge deposition (CD), the solution of the electrostatic field (EM), the integration of the particle trajectories (Int) and the particles tracking (Track).

| Δt (s) | Duration (s) | CD (%) | EM (%) | Int (%) | Track (%) |
|---------------------|--------------|--------|--------|---------|-----------|
| 2×10^{-10} | 3490 | 2 | 75 | 1 | 22 |
| 1×10^{-10} | 6180 | 3 | 84 | 2 | 11 |
| 5×10^{-11} | 12440 | 3 | 89 | 2 | 6 |

TABLE 5. Timing analysis for different values of the integration time step Δt and fixed mean edge of the triangular mesh $a = 2 \times 10^{-4}$ m. Reported is the simulation duration, along with the percentage time required for accomplishing the charge deposition (CD), the solution of the electrostatic field (EM), the integration of the particle trajectories (Int) and the particles tracking (Track).

(i.e. reduction of the non-physical linear growth of \bar{W} for $t \geq 2 \times 10^{-8}$ s) as expected (Markidis & Lapenta 2011). Nonetheless, also with the finer mesh size the 2% threshold is not respected. Therefore, the influence of the integration time step has been investigated, by varying Δt from 5×10^{-11} s up to 2×10^{-10} s (maintaining $a = 2 \times 10^{-4}$ m). In particular, the simulation parameter n_i varies from 45 up to 190 (see § 2.2). From figure 6(b), it can be noticed that the numerical heating decreases with the integration time step. In particular the combination of $a = 2 \times 10^{-4}$ m and $\Delta t = 5 \times 10^{-11}$ s leads to a maximum heating lower than 2%, as required.

For the sake of completeness, the timing analysis for each simulation performed for the convergence study has been reported in tables 4 and 5. Specifically, the simulation duration has been monitored, along with the percentage time required for accomplishing the charge deposition (CD), the solution of the electrostatic field (EM),

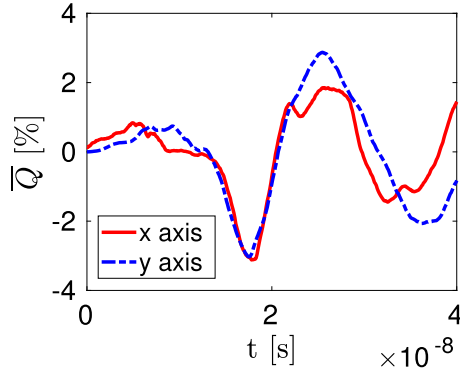


FIGURE 7. Normalized total momentum \bar{Q} (see (3.20)) plotted versus time t for mean edge of the triangular mesh $a = 2 \times 10^{-4}$ m and integration time step $\Delta t = 1 \times 10^{-10}$ s. Both components along the x (solid line) and y (dashed line) axes have been depicted.

the integration of the particle trajectories (Int) and the particles tracking (Track); see § 2 for further details on the nomenclature. The simulation duration increases roughly with the square of a as does the time required for EM (see table 4). In particular, the computational time for CD, Int and Track are less than quadratic with a ; therefore the percentage effort required for EM becomes predominant as a increases. Moreover, the simulation duration is inversely proportional to Δt as the time required for Track (see table 5). Being the computational time for CD, EM and Int almost independent of Δt , the percentage effort required for Track decreases with Δt .

Finally, we have monitored the normalized total momentum \bar{Q} in order to evaluate the effects of the self-induced forces arising because of the adoption of an unstructured and non-uniform grid (Colella & Norgaard 2010). The components along the x and y axes of \bar{Q} have been depicted in figure 7 for $a = 2 \times 10^{-4}$ m and $\Delta t = 5 \times 10^{-11}$ s. The normalized total momentum presents mild oscillations (amplitude lower than 3%) whose mean value is approximately zero. Therefore, the self-induced forces do not cause a significant departure from the theoretical condition $\bar{Q} = \mathbf{0}$ associated with the absence of external forces acting on the system. Moreover, \bar{Q} presents the same features (i.e. mild oscillations around zero) also for the other values of a and Δt investigated in the convergence analysis. In conclusion, the presence of self-induced forces does not significantly affect the results of the simulation in this application.

3.3.4. Comparison against theoretical results

In the following, the results obtained with a mesh characterized by $a = 2 \times 10^{-4}$ m and an integration time step $\Delta t = 5 \times 10^{-11}$ s, are reported and discussed. In figure 8 the phase space has been reported at different simulation times t in order to show the instability formation. In figure 8(a) ($t = 0$ s) it is possible to appreciate the two delta functions that correspond to the initial loading of the particles. Starting from figure 8(b) ($t = 5 \times 10^{-9}$ s) it is possible to see the appearance of an electron holes structure. In figure 8(c) ($t = 1 \times 10^{-8}$ s) the instability is fully formed. The trapped electrons oscillate inside holes and, as a first approximation, they are governed by the energy conservation (Che 2016). The adiabatic motions of trapped electrons quickly exchange energy between the electrons and waves (Sagdeev & Galeev 1969; Che *et al.*

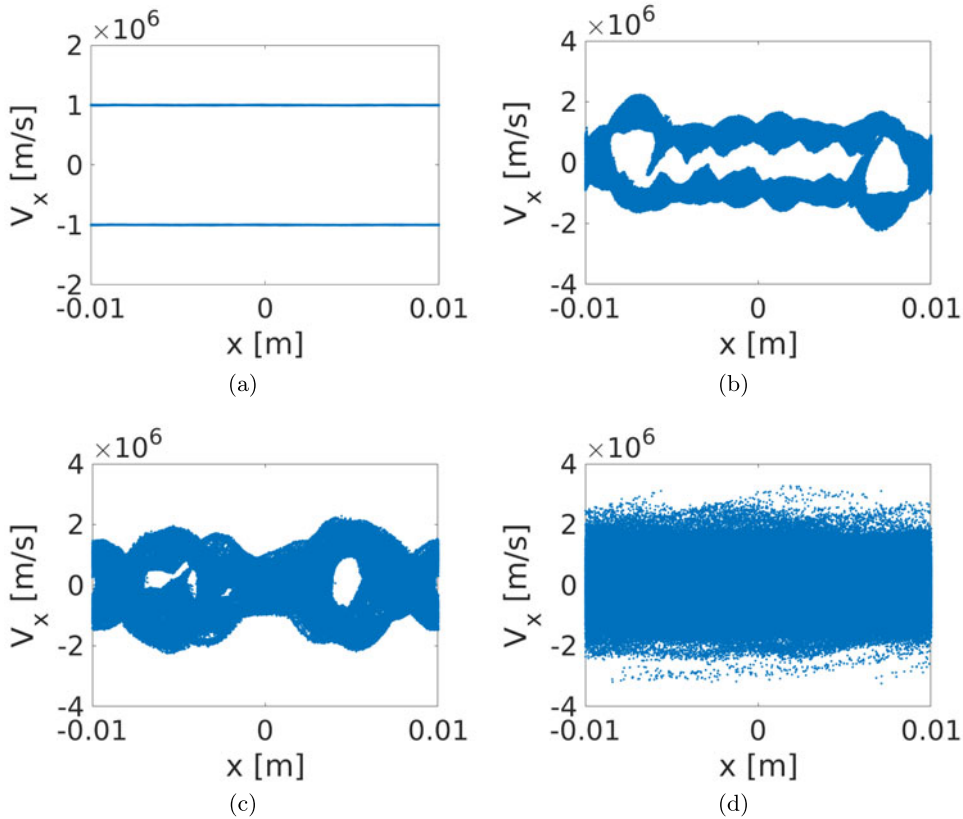


FIGURE 8. Particle phase space, position along the x axis versus component of the velocity in the same direction V_x . Different times t recorded: (a) $t=0$ s; (b) $t=5 \times 10^{-9}$ s; (c) $t=1 \times 10^{-8}$ s; (d) $t=4 \times 10^{-8}$ s.

2013). This leads to the decay of the electric fields and the trapped electrons with higher energy break the confinement of electron holes and escape. The non-adiabatic de-trapping of electrons is irreversible (Sagdeev & Galeev 1969). Eventually, these coherent structures are destroyed and transform the cold beam into a long hot tail and the plasma becomes warm. This phenomena is shown in figure 8(d) ($t=4 \times 10^{-8}$ s).

During the growth phase of the instability (figure 8a,b), most of the kinetic energy W_K of the beams is converted into the growth of the field energy W_E . Following the approach proposed in § 3.3.2, we have computed the simulated growth rate and we have compared it with the expected one obtained using (3.9). The results are shown in figure 9 where the red line is the linear fit performed using (3.12). The ratio between the theoretical (ω_{2i}^{THEO}) and the simulated (ω_{2i}^{SIM}) values of the growth rate is

$$\frac{\omega_{2i}^{THEO}}{\omega_{2i}^{SIM}} = 1.021 \pm 0.001. \tag{3.21}$$

The ratio is close to one as expected and the linear dependence, as shown by the fit, is well tested.

Moreover, the evolution of the energy of the system has been monitored during the entire simulation. Figure 10 displays the normalized total energy \bar{W} , kinetic energy

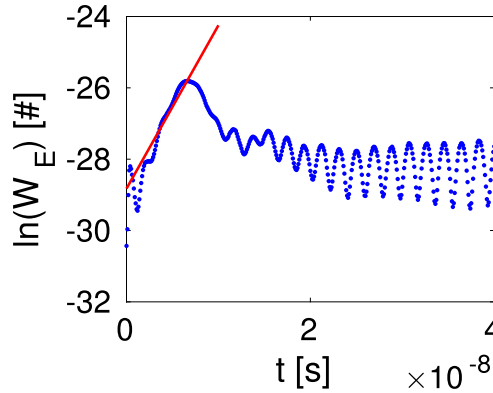


FIGURE 9. Linear growth rate analysis. Logarithm of the field energy $\ln(W_E)$ versus the simulation time t . The red line is the linear fit defined by (3.12) to verify the linear growth rate.

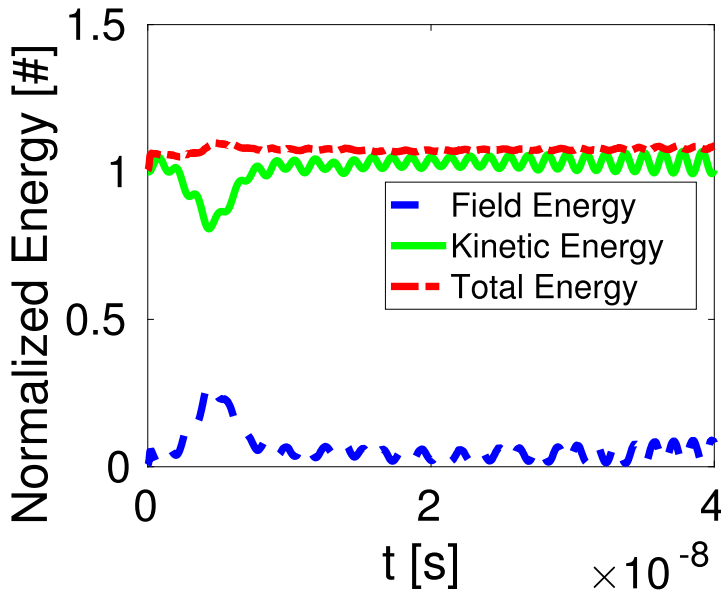


FIGURE 10. Evolution of the normalized energy of the system versus simulation time t . Reported are the normalized field energy \bar{W}_E (see (3.16)), kinetic energy \bar{W}_K (see (3.15)) and total energy \bar{W} (see (3.17)).

\bar{W}_K and field energy \bar{W}_E as a function of the simulation time t . In particular, it is possible to appreciate that the electrostatic field energy is zero initially, as expected. Between $t=0$ s and $t=5 \times 10^{-9}$ s there is a conversion of kinetic energy into field energy. The field energy grows with a growth rate ω_{2i} defined by (3.9), and the linear trend predicted by (3.12) is fully respected. After a time of roughly $1/\omega_{2i}$ from the beginning of the growth rate, the instability starts to thermalize. This leads to a decay of the electric field and, as a consequence, the trapped electrons with higher energy break the confinement of electron holes and escape.

4. Conclusions

In this paper, we have introduced a new electrostatic-magnetostatic PIC code named EDI. The latter is written in C, as it allows for a strong extensibility and reusability in adding, in the future, new models or modify the existing ones. EDI relies on an unstructured mesh of triangles, allowing for arbitrary geometries. Its structure has been explained in detail with particular attention paid to the typical characteristics of PIC codes. If compared with other PIC codes, EDI incorporates an external widely used and versatile solver called GetDP that has been properly modified in order to accept EDI input parameters. This allows us to follow the statement of flexibility, extensibility and efficiency (Verboncoeur *et al.* 1995) that should be the basis of every PIC code. EDI has been validated using as test benchmark some classic problems of plasma physics, namely the oscillation modes of a cold plasma, and the two stream instability. Moreover, the correct integration of GetDP on EDI has been checked comparing the results of our solver against FEMM.

Finally, it is worth recalling that further work is needed to make EDI a suitable instrument for the design and the analysis of real systems such as a helicon plasma thruster. First, the full set of Maxwell's equations needs to be solved in order to properly describe the propagation of the EM waves which characterize the helicon regime (Chen 1991). Second, the feeding network needed to provide the power to the thruster must be modelled properly in order to predict the total efficiency of the system. Third, EDI treats currently only 2-D geometries and the dynamics of helicon thrusters presents 2-D axisymmetric characteristics. Nonetheless, a 2-D electrostatic PIC simulation can be employed, for example, to preliminarily estimate the shape of the plasma density and electrostatic potential profiles in the presence of a diverging magnetostatic field (Rao & Singh 2012).

Acknowledgements

We would like to thank Dr D. Rondini, and Professor F. Marzari whose comments helped to improve the quality of this work.

REFERENCES

- ANDERSON, D., FEDELE, R. & LISAK, M. 2001 A tutorial presentation of the two stream instability and Landau damping. *Amer. J. Phys.* **69** (12), 1262–1266.
- BALAY, S. 2001 Petsc official web site. <https://www.mcs.anl.gov/petsc/>, accessed: 2018-09-19.
- BARBER, C. B., DOBKIN, D. P., DOBKIN, D. P. & HUHDANPAA, H. 1996 The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw. (TOMS)* **22** (4), 469–483.
- BIRDSALL, C. K. & LANGDON, A. B. 2004 *Plasma Physics via Computer Simulation*. CRC Press.
- BITTENCOURT, J. A. 2013 *Fundamentals of Plasma Physics*. Springer Science & Business Media.
- BOSSAVIT, A. 1998 *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*. Academic.
- BRIEDA, L. 2005 Development of the draco es-pic code and fully-kinetic simulation of ion beam neutralization. PhD thesis, Virginia Tech.
- BUKOWSKI, J., GRAVES, D. & VITELLO, P. 1996 Two-dimensional fluid model of an inductively coupled plasma with comparison to experimental spatial profiles. *J. Appl. Phys.* **80** (5), 2614–2623.
- BUNEMAN, O. 1959 Dissipation of currents in ionized media. *Phys. Rev.* **115** (3), 503.
- CARDINALI, A., MELAZZI, D., MANENTE, M. & PAVARIN, D. 2014 Ray-tracing WKB analysis of Whistler waves in non-uniform magnetic fields applied to space thrusters. *Plasma Sources Sci. Technol.* **23** (1), 015013.

- CARLSSON, J., MANENTE, M. & PAVARIN, D. 2009 Implicitly charge-conserving solver for Boltzmann electrons. *Phys. Plasmas* **16** (6), 062310.
- CHE, H. 2016 Electron two-stream instability and its application in solar and heliophysics. *Modern Phys. Lett. A* **31** (19), 1630018.
- CHE, H., DRAKE, J., SWISDAK, M. & GOLDSTEIN, M. 2013 The adiabatic phase mixing and heating of electrons in Buneman turbulence. *Phys. Plasmas* **20** (6), 061205.
- CHEN, F. F. 1984 *Introduction to Plasma Physics and Controlled Fusion*, 2nd edn. Plenum.
- CHEN, F. F. 1991 Plasma ionization by helicon waves. *Plasma Phys. Control. Fusion* **33** (4), 339.
- COLELLA, P. & NORGAARD, P. C. 2010 Controlling self-force errors at refinement boundaries for amr-pic. *J. Comput. Phys.* **229** (4), 947–957.
- D'AGOSTINI, G. 2003 *Bayesian Reasoning in Data Analysis: A Critical Introduction*. World Scientific.
- DAMYANOVA, M., SABCHEVSKI, S. & ZHELYAZKOV, I. 2010 Pre- and post-processing of data for simulation of gyrotrons by the gyrooss software package. *J. Phys.: Conf. Ser.* **207** (1), 012032.
- DAWSON, J. 1962 One-dimensional plasma model. *Phys. Fluids* **5** (4), 445–459.
- DULAR, P. & GEUZAIN, C. 2016 GetDP reference manual: the documentation for GetDP, a general environment for the treatment of discrete problems. <http://getdp.info>, accessed: 2018-09-19.
- DULAR, P., HENROTTE, F., ROBERT, F., GENON, A. & LEGROS, W. 1997 A generalized source magnetic field calculation method for inductors of any shape. *IEEE Trans. Magn.* **33** (2), 1398–1401.
- FABRIS, A. L., YOUNG, C. V., MANENTE, M., PAVARIN, D. & CAPPELLI, M. A. 2015 Ion velocimetry measurements and particle-in-cell simulation of a cylindrical cusped plasma accelerator. *IEEE Trans. Plasma Sci.* **43** (1), 54–63.
- FEHSKE, H., SCHNEIDER, R. & WEIßE, A. 2007 *Computational Many-Particle Physics*. Springer.
- FONSECA, R. A., SILVA, L. O., TSUNG, F. S., DECYK, V. K., LU, W., REN, C., MORI, W. B., DENG, S., LEE, S., KATSOULEAS, T. *et al.* 2002 Osiris: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators. In *International Conference on Computational Science*, pp. 342–351. Springer.
- GEUZAIN, C. 2007 Getdp: a general finite-element solver for the de rham complex. In *PAMM: Proceedings in Applied Mathematics and Mechanics*, vol. 7, pp. 1010603–1010604. Wiley Online Library.
- GEUZAIN, C. & REMACLE, J. F. 2009 Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Intl J. Numer. Meth. Engng* **79** (11).
- GHORBANALILU, M., ABDOLLAHZADEH, E. & RAHBARI, S. E. 2014 Particle-in-cell simulation of two stream instability in the non-extensive statistics. *Laser Part. Beams* **32** (3), 399–407.
- GIRAULT, V. & RAVIART, P.-A. 2012 *Finite element methods for Navier–Stokes equations: theory and algorithms*. Springer Science & Business Media.
- HASELBACHER, A., NAJJAR, F. M. & FERRY, J. P. 2007 An efficient and robust particle-localization algorithm for unstructured grids. *J. Comput. Phys.* **225** (2), 2198–2213.
- HOCKNEY, R. W. & EASTWOOD, J. W. 1988 *Computer Simulation Using Particles*. CRC Press.
- HU, Y. 2016 Particle in cell simulation. <http://theory.ipp.ac.cn/yj/>, accessed: 2018-09-19.
- INAN, U. S. & GOLKOWSKI, M. 2010 *Principles of Plasma Physics for Engineers and Scientists*. Cambridge University Press.
- JACOBS, G. & HESTHAVEN, J. S. 2006 High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *J. Comput. Phys.* **214** (1), 96–121.
- JACOBS, G. & HESTHAVEN, J. S. 2009 Implicit–explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning. *Comput. Phys. Commun.* **180** (10), 1760–1767.
- JACOBS, G., KOPRIVA, D. & MASHAYEK, F. 2001 A particle-tracking algorithm for the multidomain staggered-grid spectral method. In *39th Aerospace Sciences Meeting and Exhibit*, p. 630.
- LAPENTA, G., IINOYA, F. & BRACKBILL, J. 1995 Particle-in-cell simulation of glow discharges in complex geometries. *IEEE Trans. Plasma Sci.* **23** (4), 769–779.
- LINDMAN, E. 1970 Dispersion relation for computer-simulated plasmas. *J. Comput. Phys.* **5** (1), 13–22.

- LOTOV, K., TIMOFEEV, I., MESYATS, E., SNYTNIKOV, A. & VSHIVKOV, V. 2015 Note on quantitatively correct simulations of the kinetic beam-plasma instability. *Phys. Plasmas* **22** (2), 024502.
- MAGAROTTO, M., BOSI, F. J., DE CARLO, P., MANENTE, M., TREZZOLANI, F., PAVARIN, D., ALOTTO, P. & MELAZZI, D. 2016 Numerical investigation into the power deposition and transport phenomena in helicon plasma sources. In *COMSOL Conference*.
- MANENTE, M., TREZZOLANI, F., MAGAROTTO, M., FANTINO, E., SELMO, A., BELLOMO, N., TOSON, E. & PAVARIN, D. 2019 Regulus: a propulsion platform to boost small satellite missions. *Acta Astronautica* **157**, 241–249.
- MANZOLARO, M., MANENTE, M., CURRELI, D., VASQUEZ, J., MONTANO, J., ANDRIGHETTO, A., SCARPA, D., MENEGHETTI, G. & PAVARIN, D. 2012 Off-line ionization tests using the surface and the plasma ion sources of the spes project. *Rev. Sci. Instrum.* **83** (2), 02A907.
- MARKIDIS, S. & LAPENTA, G. 2011 The energy conserving particle-in-cell method. *J. Comput. Phys.* **230** (18), 7037–7052.
- MEEKER, D. 2014 Femm official web site. <http://www.femm.info>, accessed: 2018-09-19.
- MELZANI, M., WINISDOERFFER, C., WALDER, R., FOLINI, D., FAVRE, J. M., KRASTANOV, S. & MESSMER, P. 2013 Apar-t: code, validation, and physical interpretation of particle-in-cell results. *Astron. Astrophys.* **558**, A133.
- MOON, H., TEIXEIRA, F. L. & OMELCHENKO, Y. A. 2015 Exact charge-conserving scatter-gather algorithm for particle-in-cell simulations on unstructured grids: a geometric perspective. *Comput. Phys. Commun.* **194**, 43–53.
- NIETER, C. & CARY, J. R. 2004 Vorpai: a versatile plasma simulation code. *J. Comput. Phys.* **196** (2), 448–473.
- OSADA, R., FUNKHOUSER, T., CHAZELLE, B. & DOBKIN, D. 2002 Shape distributions. *ACM Trans. Graph. (TOG)* **21** (4), 807–832.
- PAVARIN, D., FERRI, F., MANENTE, M., CURRELI, D., MELAZZI, D., RONDINI, D. & CARDINALI, A. 2011 Development of plasma codes for the design of mini-helicon thrusters, IEPC-2011-240. In *32nd International Electric Propulsion Conference*.
- PINTO, M. C., JUND, S., SALMON, S., SONNENDRÜCKER, E. & LORRAINE, C.-I. 2008 Charge-conserving FEMPIC schemes on general grids. *Comptes Rendus Mecanique* **157**, 570–582.
- POLYCARPOU, A. C. 2005 Introduction to the finite element method in electromagnetics. *Synth. Lectures Comput. Electromagn.* **1** (1), 1–126.
- QUARTERONI, A., SACCO, R. & SALERI, F. 2010 *Numerical Mathematics*. Springer Science & Business Media.
- RAO, S. & SINGH, N. 2012 Numerical simulation of current-free double layers created in a helicon plasma device. *Phys. Plasmas* **19** (9), 093507.
- REN, J., GODAR, T., MENART, J., MAHALINGAM, S., CHOI, Y., LOVERICH, J. & STOLTZ, P. H. 2015 PIC algorithm with multiple Poisson equation solves during one time step. *J. Phys.: Conf. Ser.* **640** (1), 012033.
- SABARIEGO, R., GYSELINCK, J., DULAR, P., DE COSTER, J., HENROTTE, F. & HAMEYER, K. 2004 Coupled mechanical-electrostatic fe-be analysis with fmm acceleration: application to a shunt capacitive mems switch. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* **23** (4), 876–884.
- SAGDEEV, R. Z. & GALEEV, A. A. 1969 *Nonlinear Plasma Theory*. Benjamin.
- SEILER, H. 1983 Secondary electron emission in the scanning electron microscope. *J. Appl. Phys.* **54** (11), R1–R18.
- SPIRKIN, A. M. 2006 A three-dimensional particle-in-cell methodology on unstructured Voronoi grids with applications to plasma microdevices. PhD thesis, Worcester Polytechnic Institute.
- SPITKOVSKY, A. 2005 Simulations of relativistic collisionless shocks: shock structure and particle acceleration. *AIP Conf. Proc.* **801** (1), 345–350.
- TACCOGNA, F., LONGO, S. & CAPITELLI, M. 2004 Plasma-surface interaction model with secondary electron emission effects. *Phys. Plasmas* **11** (3), 1220–1228.

- TROPHIME, C., EGOROV, K., DEBRAY, F., JOSS, W. & AUBERT, G. 2002 Magnet calculations at the grenoble high magnetic field laboratory. *IEEE Trans. Appl. Superconductivity* **12** (1), 1483–1487.
- UMEDA, T. 2003 Study on nonlinear processes of electron beam instabilities via computer simulations. PhD thesis, Department of Communications and Computer Engineering Graduate School of Informatics Kyoto University, Kyoto, Japan.
- UMEDA, T., OMURA, Y., TOMINAGA, T. & MATSUMOTO, H. 2003 A new charge conservation method in electromagnetic particle-in-cell simulations. *Comput. Phys. Commun.* **156** (1), 73–85.
- VAHEDI, V. & SURENDRA, M. 1995 A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges. *Comput. Phys. Commun.* **87** (1–2), 179–198.
- VAY, J. L. 2008 Simulation of beams or plasmas crossing at relativistic velocity. *Phys. Plasmas* **15** (5), 056701.
- VERBONCOEUR, J. P. 2005 Particle simulation of plasmas: review and advances. *Plasma Phys. Control. Fusion* **47** (5A), A231.
- VERBONCOEUR, J. P., LANGDON, A. B. & GLADD, N. 1995 An object-oriented electromagnetic PIC code. *Comput. Phys. Commun.* **87** (1–2), 199–211.
- VILLASENOR, J. & BUNEMAN, O. 1992 Rigorous charge conservation for local electromagnetic field solvers. *Comput. Phys. Commun.* **69** (2–3), 306–316.