

CPIC: A Curvilinear Particle-in-Cell Code for Plasma–Material Interaction Studies

Gian Luca Delzanno, Enrico Camporeale, J. David Moulton, Joseph E. Borovsky,
Elizabeth A. MacDonald, and Michelle F. Thomsen

Abstract—We describe a new electrostatic particle-in-cell (PIC) code in curvilinear geometry called curvilinear PIC (CPIC). The code models the microscopic (kinetic) evolution of a plasma with the PIC method, coupled with an adaptive computational grid that can conform to arbitrarily shaped domains. CPIC is particularly suited for multiscale problems associated with the interaction of complex objects with plasmas. A map is introduced between the physical space and the logical space, where the grid is uniform and Cartesian. In CPIC, most of the operations are performed in logical space. CPIC was designed following criteria of versatility, robustness, and performance. Its main features are the use of structured meshes, a scalable field solver based on the black box multigrid algorithm and a hybrid mover, where particles' position is in logical space while the velocity is in physical space. Test examples involving the interaction of a plasma with material boundaries are presented.

Index Terms—Simulation software, spacecraft charging.

I. INTRODUCTION

PARTICLE-IN-CELL (PIC) codes have been the main tool for plasma physics kinetic simulations for several decades [1], [2]. In the PIC algorithm, a number of macroparticles (each representing many particles of the physical system) move through a computational grid due to the electromagnetic fields. The latter are self-consistently calculated from the particles on the grid. It is the interplay between the particles and the grid that makes PIC an efficient algorithm: the calculation of the force on the particles scales linearly with the number of particles (assuming that these are many more than the number of grid cells), as opposed to the quadratic scaling typical of grid-free molecular dynamics methods, where the particle force is calculated summing the interaction with every other particle [2].

Manuscript received October 31, 2012; revised June 18, 2013 and October 15, 2013; accepted October 25, 2013. Date of current version December 9, 2013. This work was supported by the Laboratory Directed Research and Development Program of the Los Alamos National Laboratory and was performed under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy by Los Alamos National Laboratory, operated by Los Alamos National Security, LLC, under Contract DE-AC52-06NA25396.

G. L. Delzanno, E. Camporeale, and J. D. Moulton are with the T-5 Applied Mathematics and Plasma Physics Group, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: delzanno@lanl.gov; enrico@lanl.gov; moulton@lanl.gov).

J. E. Borovsky is with the Space Science Institute, Boulder, CO 80301 USA (e-mail: jborovsky@SpaceScience.org).

E. A. MacDonald and M. F. Thomsen are with the ISR-1 Space Science Applications Group, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: macdonald@lanl.gov; mthomsen@lanl.gov).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TPS.2013.2290060

Traditionally, PIC codes use a uniform, Cartesian grid, and integrate the particle orbit using explicit schemes. As such, they need to resolve the shortest length scale and the fastest frequency of the plasma for numerical stability reasons. Since plasmas are inherently multiscale, it follows that PIC methods require a significant amount of computational resources to handle this disparity of scales. (We note that PIC methods with implicit time stepping, such as the implicit moment method PIC [3] or the fully implicit PIC methods recently developed in [4], [5], avoid the stability constraints typical of explicit PIC.)

The problems of PIC for multiscale plasma simulations can become worse if one needs to simulate the interaction of a plasma with material objects (for instance, spacecraft, dust grains or the walls of laboratory devices), since the objects introduce additional scales in the system. As an example, consider a spacecraft at geosynchronous orbit (an altitude of approximately 36000 km above the Earth's surface). The typical length scales are the spacecraft characteristic size $L_{\text{spacecraft}} \sim 1 - 10$ m, the plasma Debye length $\lambda_D \sim 250$ m, the electron gyroradius $\rho_e \sim 750$ m, and the proton gyroradius $\rho_i \sim 30$ km. One quickly realizes that attempting to simulate such systems with a uniform, Cartesian grid explicit PIC in three dimensions, and with grid size set by the spacecraft, is unfeasible even on today's supercomputers. We note that some approaches [6], [7] assume Boltzmann electrons and therefore only need to resolve the ions scales. These approaches do not have the cell size constraints of the explicit, full PIC method, and are typically applicable when the material object is negatively charged and in the absence of potential barriers.

Focusing on spatial scales, this discussion points to the importance of introducing some kind of grid adaptivity in the PIC method, to efficiently handle: 1) complex geometries and/or 2) the interaction of plasmas with objects whose characteristic size is much smaller than typical plasma length scales. Traditionally, PIC with nonuniform/adaptive meshes has followed two paths. One is the use of adaptive mesh refinement (AMR) techniques [8]. The other is the use of body-fitted grids, namely grids that conform exactly to a given surface [9] and avoid the stair-stepping typical of AMR grids.

In this paper, we focus on the body-fitted PIC approach. References [9]–[11] used this approach to model pulsed-power diode devices with complex geometries. In [9]–[11], a coordinate transformation (i.e. a map) from the physical space to the logical space, where the grid is uniform and Cartesian, is introduced. The quasi-stationary Maxwell equations are solved with finite differences in logical space, while particle orbits

are integrated in physical space. Since the grid is distorted in physical space, a tracking algorithm to locate in which cell a particle resides must be used [12]. Eastwood *et al.* [13] follow the body-fitted PIC approach for modeling microwave devices. They use a finite element formulation and solve both Maxwell's equations and the particle orbits in logical space. (See also the recent work in [14] on an electrostatic PIC code completely designed in logical space with a particle orbit integrator that preserves phase-space area.) Wang *et al.* [15], on the other hand, update Maxwell's equations in physical space but use a hybrid orbit integrator (where the particle position is in logical space and the velocity is in physical space). We also note that in the field of spacecraft-plasma interaction there are several efforts currently being developed that can be classified as body-fitted PIC. These are NASCAP-2k [16], SPIS [17] and PTetra [18]. While the specific details of the numerical implementations differ (see [16]–[18]), in general these approaches are in the electrostatic limit and Poisson's equation is solved with an iterative solver (conjugate gradient or GMRES method) with some form of preconditioning. Furthermore, they do not introduce the logical space and all the operations are performed in physical space. Some of these approaches (SPIS and PTetra) use unstructured computational meshes. The other major code, MUSCAT [19], is not body-fitted: it uses a structured, Cartesian, uniform mesh and Poisson's equation is solved with the fast Fourier transform algorithm, using parallel domain decomposition on multiple processors.

In this paper, we present a fully kinetic, electrostatic, body-fitted PIC code in general Curvilinear geometry called Curvilinear PIC (CPIC). While the general formulation of the method follows earlier works [9], [13], [15], CPIC was designed with flexibility, robustness, and performance considerations as the main target. For this reason, its main features are as follows.

- 1) The use of structured computational meshes. Modern multi and many-core computer architectures achieve their best performance when data movement are minimized, and computations can be expressed in a data parallel fashion without indirect memory access patterns. These characteristics give a significant advantage to structured grid approaches that are able to take advantage of known direct access patterns, simpler geometric relationships, and generally lower computational complexity. For instance, the black box multigrid (BoxMG) algorithm used in CPIC takes advantage of both known data access patterns and the bounded complexity of coarse-grid operators to solve general Poisson problems approximately 10 times faster than algebraic multigrid methods [20].
- 2) The field solver. We use the BoxMG algorithm [21]–[23] as solver. This algorithm is scalable (namely optimal): the computational cost to converge to the problem solution is linearly proportional to the number of grid cells. This means that, at least theoretically, there is no loss of performance when the problem size increases, as typical of unpreconditioned iterative solvers. In addition, robust variationally-based structured grid methods

such as BoxMG [21], [22] are able to solve problems with strongly discontinuous coefficients on the logically structured (i.e., body-fitted) grids needed here.

- 3) The particle mover. We use the hybrid mover of [15]. We have compared the performance of the hybrid mover with the (more common) physical space mover on test cases and found that the hybrid mover is typically faster and much more robust. Indeed, it avoids the issue of particle tracking needed by the physical space mover, and it also deals more efficiently with complex geometries, without the need of an extra routine to assess whether a particle crosses the domain boundaries.

This paper is organized as follows. In Section II, we describe the main algorithmic and implementation aspects of CPIC. In Section III, we present some benchmark tests involving the interaction of a plasma with material boundaries. In Section IV, we draw conclusions.

II. OVERVIEW OF THE CURVILINEAR PIC

We model a collisionless, magnetized plasma described by Vlasov's equations

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v} \times \mathbf{B}_0) \cdot \nabla_v f_\alpha = 0 \quad (1)$$

where the subscript α refers to the plasma species, $\alpha = e$, (i) for electrons (ions). In (1), $f_\alpha(\mathbf{x}, \mathbf{v}, t)$ is the plasma distribution function, $\mathbf{E} = -\nabla\phi$ is the electric field (ϕ is the electrostatic potential), $\mathbf{B}_0(\mathbf{x})$ is the magnetic field, while q_α and m_α are the charge and the mass of the plasma particles. We focus on the electrostatic limit; therefore, (1) is coupled through the electrostatic potential via Poisson's equation

$$\nabla^2 \phi = -\frac{\sum q_\alpha n_\alpha}{\epsilon_0} = -\frac{\rho}{\epsilon_0} \quad (2)$$

where the plasma densities are given by

$$n_\alpha = \int f_\alpha d\mathbf{v} \quad (3)$$

and the magnetic field $\mathbf{B}_0$ does not evolve in time. In (2), ϵ_0 is the permittivity of vacuum.

The PIC method solves (1) numerically by introducing macroparticles, each representing a large number of actual particles of the system, and following their characteristics:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \quad (4)$$

$$m_p \frac{d\mathbf{v}_p}{dt} = q_p [\mathbf{E}(\mathbf{x}_p) + \mathbf{v}_p \times \mathbf{B}_0(\mathbf{x}_p)] \quad (5)$$

where $\mathbf{x}_p$ and $\mathbf{v}_p$ are position and velocity of each particle. Hence, a standard electrostatic PIC cycle, repeated at each time step of the simulation, consists of the following four steps.

- 1) *Particle mover*: particles are moved according to (4) and (5), given the electromagnetic fields.
- 2) *Particle-to-Grid ($P \rightarrow G$) interpolation*: the charge carried by the particles is accumulated on the computational grid via interpolation, to obtain the plasma charge density.
- 3) *Field solver*: Poisson's equation is solved on the computational grid, given the charge density.

- 4) *Grid-to-Particle ($G \rightarrow P$) interpolation*: The electromagnetic fields are interpolated to the particle position.

We now proceed to discuss each of these steps as they are treated in CPIC.

A. Curvilinear Formulation

First, we introduce a coordinate transformation

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) \quad (6)$$

between the physical space X with coordinates $\mathbf{x} = (x^1, x^2, x^3)$, and the logical space Ξ (the unit cube), with coordinates $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3) = (\xi, \eta, \zeta)$. Given a computational mesh in physical space corresponding to the simulation domain, (6) maps it to the unit cube in logical space, where the computational mesh is uniform and Cartesian. We define the Jacobi matrix as [24]

$$j_{\alpha\beta}(\boldsymbol{\xi}) = \frac{\partial x^\alpha}{\partial \xi^\beta}, \quad \alpha, \beta = 1, 2, 3 \quad (7)$$

while its inverse is

$$k^{a\beta}(\mathbf{x}) = \frac{\partial \xi^a}{\partial x^\beta}, \quad \alpha, \beta = 1, 2, 3. \quad (8)$$

The Jacobian of the transformation is the determinant of the Jacobi matrix

$$J(\boldsymbol{\xi}) = \det[j_{\alpha\beta}]. \quad (9)$$

Similarly, the covariant metric tensor is defined as [24]

$$g_{\alpha\beta}(\boldsymbol{\xi}) = \frac{\partial x^\gamma}{\partial \xi^\alpha} \frac{\partial x^\gamma}{\partial \xi^\beta}, \quad \alpha, \beta = 1, 2, 3 \quad (10)$$

while the contravariant metric tensor is

$$g^{a\beta}(\mathbf{x}) = \frac{\partial \xi^a}{\partial x^\gamma} \frac{\partial \xi^\beta}{\partial x^\gamma}, \quad \alpha, \beta = 1, 2, 3. \quad (11)$$

Here and everywhere else in the text, repeated indices imply summation (Einstein's notation). Note that the metric tensor matrix is symmetric and that grids are orthogonal if $g_{12} = g_{13} = g_{23} = 0$. With these quantities, we can express all the metric elements from physical space to logical space. For instance, Poisson's equation in logical space becomes

$$\nabla^2 \phi = \frac{1}{\Omega J} \frac{\partial}{\partial \xi^i} \left(\Omega J g^{ij} \frac{\partial \phi}{\partial \xi^j} \right) = -\frac{\rho}{\epsilon_0} \quad (12)$$

where we have introduced a geometric factor Ω that can be used when the geometry of the physical space is non-Cartesian. For instance, for Cartesian geometry we have ($x^1 = x$, $x^2 = y$, $x^3 = z$) and $\Omega = 1$, while for cylindrical geometry we have ($x^1 = r$, $x^2 = \theta$, $x^3 = z$) and $\Omega = r$.

From the discussion so far, it is clear that the construction of the map from physical to logical space, (6), is a critical step in CPIC. In the case of extremely simple geometries, like the test examples considered in this paper, one can specify such map analytically. Alternatively, one can use a suitable mesh generator. For cases that are still relatively simple, where the map is characterized by a single or few structured blocks, we normally use methods based on the solution of a set of partial differential equations such as Winslow's method [25] or the optimal distortion method [26].

On the other hand, to capture the full complexity of a spacecraft we plan to use commercially available software for structured meshes, typically developed in the computational fluid dynamics community (see, for instance, GridPro, <http://www.gridpro.com>). The resulting computational grid is block structured and, for complex geometries, might involve hundreds of blocks. This will require upgrading CPIC to handle block-structured meshes. However, the fact that each block maps to a logically rectangular Cartesian mesh ensures that the localization of the particles on the mesh of each block and the particle mover remain essentially unchanged, while there is an additional computational cost in assigning each particle to a block. In CPIC, this last step is trivial provided that each grid block stores the maximum and minimum value of its logical coordinates (as done in the domain decomposition for the parallelization), since the particle position is in logical space. We also note that in CPIC the metric coefficients (the contravariant metric tensor, the inverse of the Jacobi matrix and the Jacobian) are computed from finite difference approximation given the discrete map (6) known at cell vertices (see [26]). There is no conceptual difference or additional computational cost between a structured and block-structured mesh since the metric coefficients are defined in each block. Furthermore, there are three popular approaches to developing a Poisson solver for block-structured grids. First, one may use a domain decomposition approach, with BoxMG as the subdomain solver. This shares convergence property of domain decomposition methods, and for large problems would require overlap to obtain satisfactory scaling. Second, BoxMG could be extended following the approach used in the HYPRE libraries semistructured-grid interface to its parallel semicoarsening multigrid solver, which manages the coarsening of each block independently, while maintaining the full connectivity of the global problem [27]. Finally, a bounding box approach can be used to include the entire global domain in a single BoxMG solve, with internal boundary conditions used at the block interfaces, and this is the approach that we plan to adopt.

B. Computational Aspects of CPIC

In CPIC, quantities are expressed in normalized units: lengths are normalized to the electron Debye length, velocities to the electron thermal velocity, time to the electron plasma frequency, the electrostatic potential to the electron temperature, densities to a reference density, and the magnetic field to a reference magnetic field. In the tests presented throughout this paper, we will always use normalized quantities.

CPIC is designed in a staggered formulation, where the electrostatic potential and the density are at cell centers (labeled with subscript c) and the electric field is at vertices (v). The staggered formulation and the fact that the assignment function for the interpolation from particles to the grid is one order higher than the one for the interpolation from the grid to the particles (see below) imply that CPIC is an energy-conserving PIC as defined in [1].

We have developed and successfully tested a 2-D and 3-D version of CPIC. CPIC2-D is fully parallelized with domain

decomposition and the MPI library to handle the communication among processors. Parallelization of CPIC3-D is ongoing.

1) $P \rightarrow G$ Interpolation: The accumulation of the particle charge on the grid is given by

$$n_{a,c} = \frac{\sum_p q_p W_{pc} (\xi_p - \xi_c)}{J_c \Delta \xi \Delta \eta \Delta \zeta} \quad (13)$$

where the assignment function is

$$W_{pc} = b_2(\xi_p - \xi_c) b_2(\eta_p - \eta_c) b_2(\zeta_p - \zeta_c) \quad (14)$$

and b_2 is the b-spline of order 2

$$b_2(\xi) = \begin{cases} \frac{3}{4} - \xi^2, & |\xi| < \frac{1}{2} \\ \left(\frac{3}{2} - |\xi|\right)^2, & \frac{1}{2} < |\xi| < \frac{3}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

In (13), $\sum_p$ implies summation over all the particles.

2) $G \rightarrow P$ Interpolation: The interpolation of the force field from the computational mesh to the particle position is given by

$$\mathbf{E}_p = \sum_v \mathbf{E}_v W_{vp} (\xi_v - \xi_p) \quad (16)$$

with the assignment function

$$W_{vp} = b_1(\xi_v - \xi_p) b_1(\eta_v - \eta_p) b_1(\zeta_v - \zeta_p) \quad (17)$$

given as the product of b-splines of order 1

$$b_1(\xi) = \begin{cases} 1 - |\xi|, & |\xi| < 1 \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

We note that the interpolation operations are performed in logical space and therefore are equivalent to what is done in the standard PIC with uniform grid. The same approach is used in [28].

3) *Field Solver*: We solve Poisson's equation on the logical grid. We use a second order, conservative discretization, and impose boundary conditions via ghost cells. The resulting linear system is solved with a robust variational multigrid method [21]. The need for a scalable algorithm motivates the use of a multigrid solver, as it is one of very few truly scalable solvers (it can be shown that the order of complexity of the method is $O(N)$, with N the number of unknowns of the linear system [29]). Moreover, efficient parallel implementations are readily available.

Multigrid methods are iterative, and achieve their efficiency through the recursive use of successively coarser discrete problems (i.e., a sequence of coarse-grid discrete operators) in conjunction with smoothing on each level (e.g., a single Gauss-Seidel iteration on each level) to damp the highly oscillatory errors associated with each grid [29]. Early work on multigrid began with structured grids and took a natural but simple geometric viewpoint: coarsening removed every other point in each coordinate direction; the same discretization was applied directly at each grid resolution; and the interpolation and restriction operators were defined based on the grid geometry alone. Unfortunately, this approach was not robust for challenging problems with discontinuous coefficients or

TABLE I
CONVERGENCE STUDY OF THE BoxMG SOLVER CHANGING THE
NUMBER OF GRID POINTS

Grid	L_2 error	Error ratio	Number of iterations	Time [s]	Time ratio
16^2	$5.3 \cdot 10^0$		5		
32^2	$2.0 \cdot 10^{-1}$	28.6	5		
64^2	$3.7 \cdot 10^{-2}$	5.1	5		
128^2	$8.8 \cdot 10^{-3}$	4.1	5	0.06	
256^2	$2.2 \cdot 10^{-3}$	4.0	6	0.27	4.5
512^2	$5.4 \cdot 10^{-4}$	4.0	6	1.2	4.4
1024^2	$1.3 \cdot 10^{-4}$	4.0	5	4.2	3.4
2048^2	$3.4 \cdot 10^{-5}$	4.0	5	16.6	4.0
4096^2	$8.4 \cdot 10^{-6}$	4.0	5	66.4	4.0

severely distorted grids. To address this shortcoming, a robust approach for problems on logically structured grids was developed that only requires the user to provide the fine-grid matrix and the right-hand side [21]. This BoxMG algorithm uses a variational coarse-grid operator, as it minimizes the error in the range of the interpolation. Moreover, it uses the entries in the matrix (the discrete operator) to define the interpolation, dubbed operator-induced interpolation. This approach ensures that important properties of the fine-scale PDE are well approximated at all resolutions, and leads to a robust scalable solver suitable for grids of any dimension, discontinuous coefficients, and any type of boundary condition (Dirichlet, Neumann, Robin, and periodic) [23]. In addition, it is robust for anisotropic problems, provided line-relaxation is used instead of point smoothing.

As an illustration of this capability, we use the method of manufactured solutions [30] on the (normalized) Poisson equation in spherical geometry in 2-D. This can be done by assuming that the physical space corresponds to the cylindrical geometry ($x^1 = r$, $x^2 = \theta$, $x^3 = z$), with $\partial/\partial\theta = 0$, and introducing the following coordinate transformation:

$$r = [r_1 + (r_2 - r_1)\xi] \sin[\pi(1 - \eta)] \quad (19)$$

$$z = [r_1 + (r_2 - r_1)\xi] \cos[\pi(1 - \eta)] \quad (20)$$

where r_1 and r_2 are the inner and outer radii of the physical domain (we choose $r_1 = 1$ and $r_2 = 10$ in this example). The geometric factor is set to $\Omega = r$. Next, we seek the following solution:

$$\phi(\xi, \eta) = \sin[(r_2 - r_1)^2 \xi (1 - \xi)] \cos(\pi \eta^2) \quad (21)$$

which is inserted in Poisson's equation to obtain the density. We then solve Poisson's equation with the BoxMG solver on a serial machine. We use line-relaxation since this example is fairly anisotropic in terms of metric tensor coefficients. The results are shown in Table I, in the form of a convergence study changing the number of grid points. The second column of Table I shows the L_2 norm of the error of the numerical solution relative to the analytical solution in (21), while the third column is the inverse of the ratio of the error calculated on a given grid divided by the error from the previous coarser grid. The second-order convergence of our discretization scheme is clearly recovered (i.e., when doubling the resolution, the error ratio is equal to 4 for sufficiently

large grids). The fourth column shows the number of iterations needed by the solver to converge, with relative tolerance set to $r_{\text{tol}} = 10^{-10}$. The iteration count remains practically constant. The fifth and sixth columns show the time to reach convergence and the ratio of such time on a given grid relative to the previous coarser grid. Here, we have only plotted the time for the more refined grids to identify the asymptotic scaling. As expected, it shows that the algorithm is scalable: doubling the grid resolution in each direction requires about four times more time for convergence. This is also consistent with the iteration count remaining constant as the grid is refined.

4) *Particle Mover*: We initially specialize the discussion to the case of Cartesian physical space. The most common approach for particle mover in PIC codes with nonorthogonal grids is to move particles in physical space via (4) and (5). As the particles move through a distorted mesh, a tracking/localization procedure is required to locate the particle in a cell, as needed by the interpolation and accumulation routines. (An alternative, used in [28], is to move particles in physical space and then invert the map $\mathbf{x}(\xi)$ to obtain the particles' position in logical space. This technique can be very efficient for cases where the map can be inverted analytically.) In addition, the time discretization of (4) and (5) is usually explicit, with the leap-frog integrator [1] as the most popular choice. The leap-frog algorithm staggers particles position and velocity by half time step, and is second-order accurate in time. In CPIC, however, we follow a different approach, which was proposed in [15]: particles retain their physical space velocity but are characterized by their logical space position (and therefore move in logical space). In other words, our mover consists of (5), while (4) is replaced by its logical space equivalent [obtained by projecting (4) on the contravariant base vector]

$$\frac{d\xi_p^i}{dt} = k^{ij} v_p^j, \quad i = 1, 2, 3 \quad (22)$$

where the inverse of the Jacobi matrix is given by (8) and we have used index notation. We refer to (5) and (22) as the hybrid mover.

The hybrid mover offers some clear advantages over a physical space mover, mainly because of its simplicity.

- 1) There is no need of a tracking algorithm to locate the particles: a particle is immediately assigned to a cell through its logical space coordinate.
- 2) While tracking algorithms applied to complex boundaries (multiply connected or with concave boundaries) are typically not robust, this is not the case for the hybrid mover. In fact, a particle trivially leaves the domain if one of its logical coordinates becomes less than zero or greater than one.

There is, however, an important disadvantage of the hybrid mover relative to its physical space counterpart: a simple leap-frog integrator loses second-order accuracy on a nonorthogonal mesh. The simplest way to fix this problem and regain second-order accuracy is to use a leap-frog integrator modified with a predictor-corrector approach [15]. The algorithm is as follows (index n refers to the time level, Δt is the time step).

- 1) Perform a full-step update of the particle velocity to obtain $\mathbf{v}_p^{n+1/2}$

$$\frac{\mathbf{v}_p^{n+1/2} - \mathbf{v}_p^{n-1/2}}{\Delta t} = \frac{q_p}{m_p} \left[\mathbf{E}(\xi_p^n) + \mathbf{v}_p^n \times \mathbf{B}_0(\xi_p^n) \right] \quad (23)$$

with

$$\mathbf{v}_p^n = \frac{\mathbf{v}_p^{n+1/2} + \mathbf{v}_p^{n-1/2}}{2}. \quad (24)$$

This step is performed explicitly with the algorithm introduced in [31], which separates the velocity update due to the electric field from the update due to the magnetic field.

- 2) Predictor: Perform a half-step update of the particle logical position to obtain ξ_p'

$$\frac{\xi_p^{i'} - \xi_p^{i,n}}{\Delta t} = k^{ij}(\xi_p^n) v_p^{j,n+1/2}, \quad i = 1, 2, 3. \quad (25)$$

- 3) Reevaluate the metric coefficients at ξ_p' via interpolation

$$k^{ij}(\xi_p') = \sum_v k^{ij}(\xi_v) W_{vp}(\xi_v - \xi_p'). \quad (26)$$

- 4) Corrector: Perform a full-step update of the particle logical position with the new metric coefficients, to obtain ξ_p^{n+1}

$$\frac{\xi_p^{i,n+1} - \xi_p^{i,n}}{\Delta t} = k^{ij}(\xi_p') v_p^{j,n+1/2}, \quad i = 1, 2, 3. \quad (27)$$

With these considerations, an important question is whether the hybrid mover is more or less efficient relative to the physical space mover from a performance standpoint. To answer this question, we have compared the hybrid mover with a physical space mover for the following test. We consider a square domain in 2-D (Cartesian geometry), with the following physical-to-logical space map:

$$\begin{aligned} x &= \xi + \varepsilon \sin(2\pi\xi) \sin(2\pi\eta) \\ y &= \eta + \varepsilon \sin(2\pi\xi) \sin(2\pi\eta). \end{aligned} \quad (28)$$

The parameter ε controls the level of distortion of the grid. We fix the electromagnetic fields on the grid, the mesh size (64×64), the number of particles per cell $N_{p,\text{cell}} = 100$ and we move the particles for 10^4 time steps. The boundary conditions on the particles are periodic on all sides. The physical space mover requires a particle tracking and localization technique and we use the method proposed in [12]. The results of our comparison changing time step are shown in Tables II and III for orthogonal (uniform) ($\varepsilon = 0$) and nonorthogonal ($\varepsilon = 0.1$) grids. For the latter, the ratio of the largest to smallest cell size is 4.4. Tables II and III show the average time per particle per time step to complete the simulation, normalized to the time required by the hybrid mover on the uniform mesh with $\Delta t = 0.001$. Let us focus first on Table II. We observe that the time required by both movers to complete the simulation increases with the time step. For the hybrid mover, this is because additional operations are performed due to particles crossing the domain boundaries.

TABLE II
COMPARISON OF THE TIME (PER PARTICLE PER TIME STEP) REQUIRED
BY THE HYBRID MOVER AND THE PHYSICAL SPACE MOVER ON AN
ORTHOGONAL (UNIFORM) GRID FOR VARIOUS TIME STEPS

Time step	Hybrid mover: time	Physical space mover: time
0.001	1.00	0.80
0.002	1.02	0.86
0.004	1.08	1.05
0.01	1.17	1.31
0.025	1.17	1.31
0.05	1.17	1.44

TABLE III
COMPARISON OF THE TIME (PER PARTICLE PER TIME STEP) REQUIRED
BY THE HYBRID MOVER AND THE PHYSICAL SPACE MOVER ON A
NONORTHOGONAL GRID ($\varepsilon = 0.1$) FOR VARIOUS TIME STEPS

Time step	Hybrid mover: time	Physical space mover: time
0.001	1.04	1.04
0.002	1.06	1.14
0.004	1.10	1.31
0.01	1.16	1.71
0.025	1.16	1.71
0.05	1.17	1.84

For the physical space mover, however, there is an extra work associated with the tracking/localization algorithm: with larger time steps, particles move further away and there are more particles crossing cell boundaries within the domain. Thus, for realistic time steps used in kinetic simulations, even on a uniform grid the hybrid mover can be faster. Let us now look at the results on the nonuniform grid, Table III. We notice that the average time required by the hybrid mover is comparable with the one on the uniform grid. This result is not surprising since, given the metric coefficients, the hybrid mover treats different grids equally. It indicates, however, that the hybrid mover is extremely robust. The physical space mover, on the other hand, requires more time relative to its performance on the uniform grid. Again, this is not surprising as now the tracking/localization algorithm has to deal with distorted meshes. On this example, with time step $\Delta t = 0.05$ the hybrid mover is about 60% faster. We note that the results shown in Tables II and III involve the time spent by the algorithm to move and localize the particles and the time spent to apply the (periodic) boundary conditions when a particle crosses the domain boundaries. The latter time is the same for the hybrid and physical space movers since the example considered consists of a simple square domain. Therefore, the performance loss of the physical space mover is due to the particle tracking/localization only. A further point worth noting is that particle tracking/localization algorithms are notoriously less robust on complex boundaries, and this will make the comparison even more favorable with the hybrid mover.

One final comment is with regard to the application of CPIC to non-Cartesian geometry. This is useful for instance when one wants to simulate the interaction of a plasma and

an object in a spherically symmetric system, and we will show some of these examples in Section III. In this case, one can run CPIC in 2-D cylindrical geometry and impose azimuthal symmetry, thus allowing a much higher resolution than in a fully 3-D simulation. From the perspective of the solver, this implies setting $\Omega = r$ and $\partial/\partial\theta = 0$ in Poisson's equation. For the mover, on the other hand, one must consider the contribution of the inertial forces to the particle motion. This can be done with the cylindrical version of the Boris mover [31], suitably modified in its hybrid mover version. In addition, for simulations involving a uniform background magnetic field, this algorithm can be improved to resolve the particle gyromotion exactly [32], [33]. Its importance resides in the fact that, by describing the gyromotion exactly, one can relax the time step constraints that a PIC code with a standard leap-frog mover would have when simulating a strongly magnetized plasma with $\omega_{ce} \gg \omega_{pe}$ (with ω_{ce} the electron cyclotron frequency and ω_{pe} the electron plasma frequency) and deliver a significant computational savings.

III. TESTS

We have successfully performed several tests to check the validity of CPIC against known solutions. These tests include Langmuir waves, Landau damping and two-stream instabilities, for cases where the plasma is not in contact with a material wall. Here, we present some additional tests where the plasma is bounded by a material boundary. These tests are performed in 2-D (for the fields, while the particles retain three components of the velocity), by exploiting some symmetry of the system.

A. Charging and Shielding of a Conducting Planar Wall in an Unmagnetized Plasma

First, we study the charging and shielding of a perfectly conducting planar wall in an unmagnetized plasma consisting of electrons and singly charged ions. We consider a rectangular domain $L_x \times L_y = 20 \times 5$ (in units of electron Debye length) in Cartesian geometry and use a uniform mesh

$$\begin{aligned} x &= L_x \xi \\ y &= L_y \eta. \end{aligned} \quad (29)$$

The wall is located at the left boundary, $\xi = 0$, and absorbs plasma. The computational domain is initially empty. At each time step, we inject plasma particles from the right boundary ($\xi = 1$). The injection fluxes are calculated according to a Maxwellian at rest for the electrons and a drifting Maxwellian (with drift velocity $v_{di} = -2.1$, normalized to the ion thermal speed $v_{thi} = \sqrt{T_i/m_i}$ with T_i the ion temperature) for the ions. The boundary conditions for the field solver are periodic at the bottom ($\eta = 0$) and top ($\eta = 1$) boundaries, $\phi = 0$ at the right boundary, and Gauss' law is applied at the wall. Particles can be absorbed by the wall (namely, they are removed from the simulation and their charge is accumulated to the wall), can leave the system at the right boundary, while they reenter at the bottom (top) if they cross the top (bottom) boundary. Other parameters of the simulation are $\omega_{pe}\Delta t = 0.05$, the temperature ratio

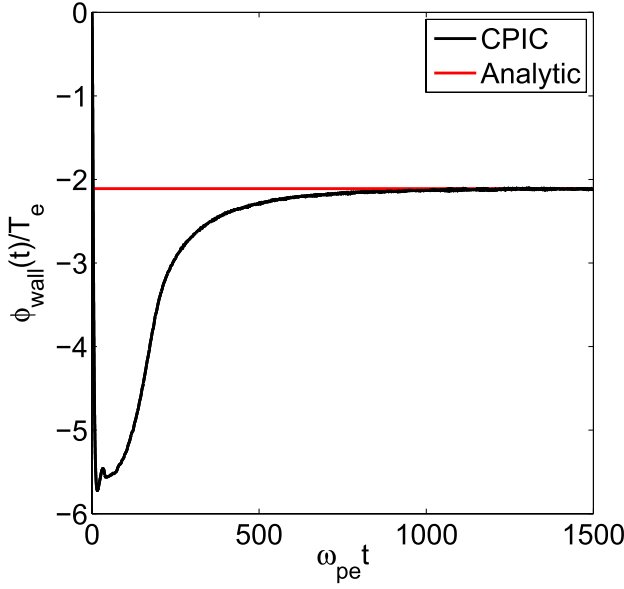


Fig. 1. Test A: time evolution of the wall potential.

of the injected plasma is $T_e/T_i = 1$ and the mass ratio is $m_i/m_e = 1836$.

Given the plasma fluxes at the right boundary, it is possible to calculate the (1-D) steady-state solution of the system. The theory is based on particle and energy conservation. It requires the solution of a nonlinear Poisson equation, where the plasma densities are obtained analytically as function of the electrostatic potential. We do not report these expressions here for brevity. A similar theory, which does not include the ion drift velocity, can be found in [34]. We refer to this procedure as the analytic solution and use it to benchmark against CPIC.

Fig. 1 shows the time evolution of the wall potential (normalized to the electron temperature) obtained with CPIC for a 128×128 mesh. The computational domain is initially empty. As the electrons move toward the wall much faster than the ions, at the beginning there is a strong negative spike in the potential: $\phi_{\text{wall}}/T_e \simeq -5.7$. Slowly, the ion distribution function adjusts to its equilibrium value and a steady state is reached. At equilibrium, the wall potential (averaged over the last sixth of the simulation) is $\phi_{\text{wall}}/T_e \simeq -2.11$, in good agreement with the one from the analytic theory $\phi_{\text{wall}}^{\text{analytic}}/T_e \simeq -2.10$. Fig. 2 shows the shielding potential at the end of the simulation. As expected, the structure of the potential is 1-D (aside from some fluctuations in the region where the potential is very small). A comparison of the equilibrium shielding potential with the analytic solution (not shown) indicates that the screening of the wall by the plasma is captured correctly in the simulation.

B. Charging and Shielding of a Conducting Sphere in an Unmagnetized Plasma

The next natural test is to study the charging and shielding of a perfectly conducting spherical object in an unmagnetized plasma (see [35]). We can exploit the curvilinear geometry

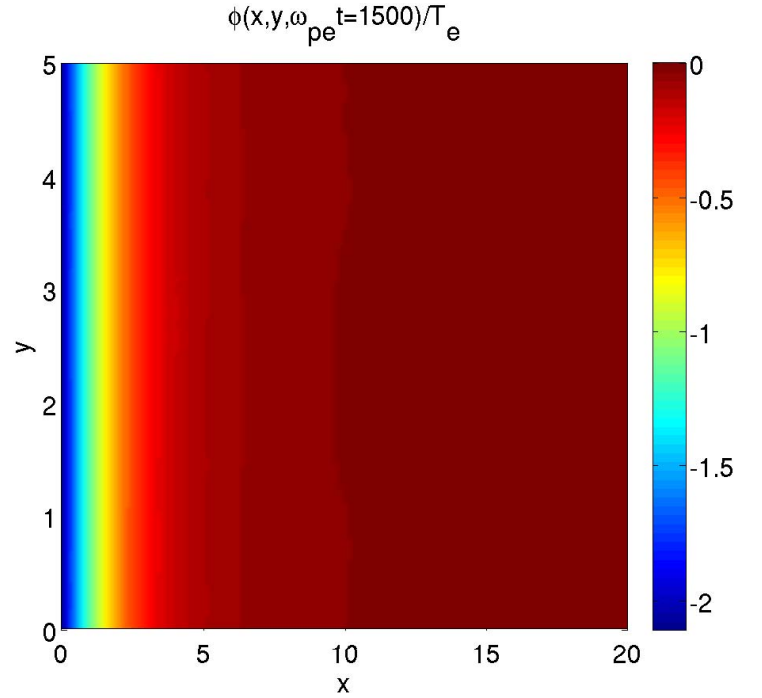


Fig. 2. Test A: equilibrium shielding potential at the end of the simulation.

of CPIC and the fact that the system has spherical symmetry. Thus, we run CPIC in 2-D with a physical space represented in cylindrical geometry ($x^1 = r$, $x^2 = \theta$, $x^3 = z$), and imposing azimuthal symmetry. The coordinate transformation between the physical and logical space is then given in (19) and (20). This corresponds to a physical domain between two spheres of radii r_1 and r_2 .

The simulation setup is as follows. The computational box is initially loaded with a Maxwellian plasma at rest. The boundary conditions on the left and right boundaries in logical space (inner and outer spheres, respectively) remain the same as in the previous example. However, we use homogeneous Neumann boundary conditions (zero normal derivative of the potential, $\partial\phi/\partial r = \partial\phi/\partial\zeta = 0$ since in this case the grid is orthogonal) for the fields at the bottom and top boundaries in logical space (corresponding to the z axis in physical space) and particles are specularly reflected there. At the right boundary, we inject Maxwellian fluxes with no drift velocities. The other simulation parameters are the same as in Test A, except for $\omega_{pe}\Delta t = 0.1$. We also set $r_1 = 1$ and $r_2 = 10$.

The analogue of the equilibrium theory briefly discussed in the previous example can be developed in spherical geometry (see [36]). It is more complicated because the conservation of angular momentum and at least two components of the particle velocity must be included in the calculation. For these reasons, it is common to use an approximated theory, the orbital-motion-limited (OML) theory [37], to obtain the sphere floating potential. While OML is an approximated theory, previous studies have shown that OML indeed provides a quite accurate result even when the sphere radius is comparable with the plasma Debye length [38].

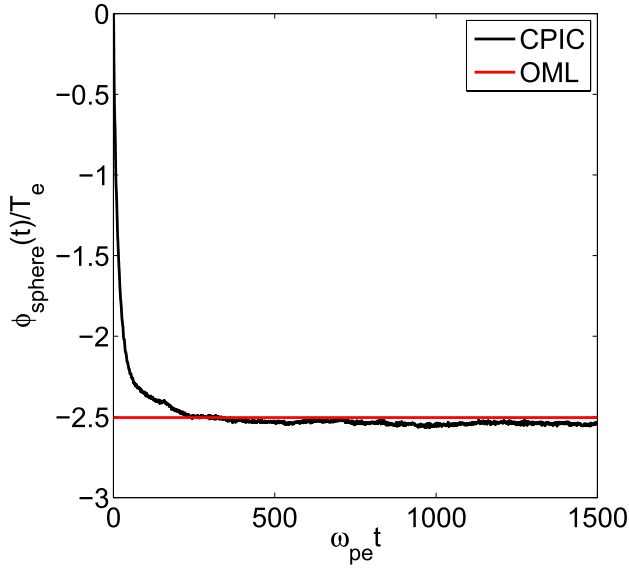


Fig. 3. Test B: time evolution of the wall potential.

Fig. 3 shows the time evolution of the normalized floating potential obtained for a 128×128 mesh. Unlike the case of Fig. 1, the potential approaches its asymptotic equilibrium value monotonically. This is because the computational domain is loaded with plasma at $t = 0$, and the ions immediately start charging the sphere. One can also observe that the asymptotic equilibrium value, $\phi_{\text{sphere}}/T_e \simeq -2.54$, matches reasonably well the equilibrium value obtained by the OML theory, $\phi_{\text{OML}}/T_e = -2.50$. The fact that the equilibrium value is slightly lower than OML can be attributed to the development of an absorption barrier for the ions, as discussed in [39]. The equilibrium shielding potential is shown in Fig. 4. The spherical symmetry is evident from the plot. In Fig. 5, we compare the equilibrium shielding potential averaged over η with that obtained with a 1-D PIC code designed in spherical geometry. The latter was used for instance in [38] and [40] to study charging and shielding of electron emitting spherical dust grains and is well benchmarked. There is a good agreement between the two, indicating that CPIC captures correctly the screening of the charged sphere by the plasma. Furthermore, we have performed a convergence study changing the number of grid points with the goal of checking the scalability of the solver in a practical situation. The results are presented in Table IV (which also shows the average number of particles per cell at the end of the simulation). The number of solver iterations averaged over the entire simulation remains fairly flat.

C. Charging and Shielding of a Conducting Sphere in a Magnetized Plasma

Our last example involves the charging and shielding of a conducting, spherical object in a magnetized plasma. We consider a uniform magnetic field directed along z . For this case, since there is no analytic theory that relates the sphere floating potential to the plasma parameters for arbitrary values of the magnetic field, we perform some of the studies in [18] for benchmark. Specifically, we consider two cases. In the first

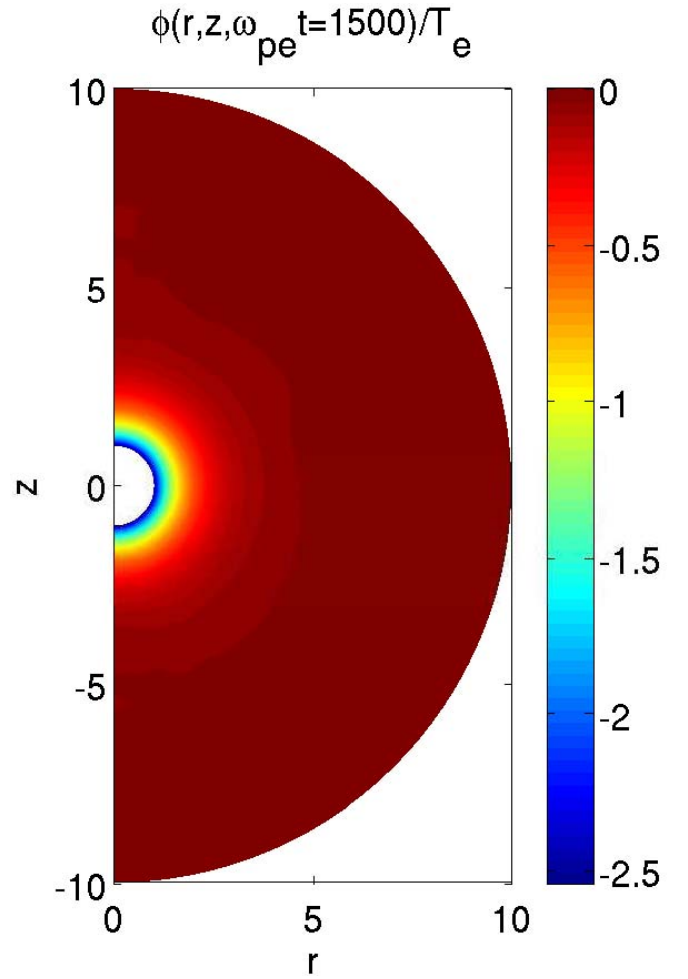


Fig. 4. Test B: equilibrium shielding potential at the end of the simulation.

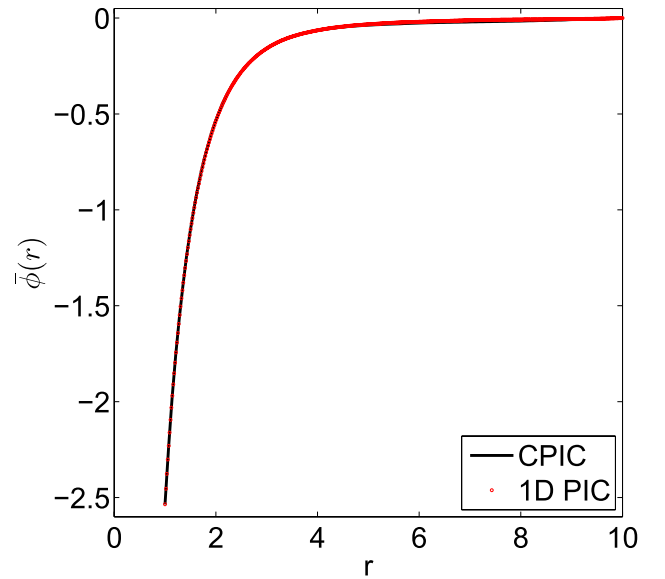


Fig. 5. Test B: equilibrium shielding potential at the end of the simulation, averaged over η .

one, the magnetic field has magnitude $B_{z0} \simeq 1.9$ such that the electron thermal gyroradius is $\rho_e/\lambda_{De} = 0.533$ and the electron gyrofrequency is $\omega_{ce}/\omega_{pe} \simeq 1.9$. Other parameters

TABLE IV
CONVERGENCE STUDY OF THE BoxMG SOLVER FOR TEST B

Grid	Average number of iterations	Average number of electrons per cell	Average number of ions per cell
32^2	8.0	231	330
64^2	8.0	230	276
128^2	8.0	230	248
256^2	9.0	230	230

and the simulation settings are as in Test B. In particular, the ion thermal gyroradius is $\rho_i/\lambda_{De} \simeq 22.8$ and the ion gyrofrequency is $\omega_{ci}/\omega_{pe} \simeq 0.001$. Thus, in this example, the electron gyroradius is comparable with the sphere radius and we refer to this case as “weakly magnetized.” In the second example, we increase the value of the magnetic field by a factor of 10, $B_{z0} \simeq 19$. Consistently, the electron thermal gyroradius is $\rho_e/\lambda_{De} = 0.0533 \ll r_1/\lambda_{De}$, while the ion thermal gyroradius is $\rho_i/\lambda_{De} \simeq 2.3 > r_1/\lambda_{De}$. The electron gyrofrequency is $\omega_{ce}/\omega_{pe} \simeq 18.8$, while for the ions $\omega_{ci}/\omega_{pe} \simeq 0.01$. We refer to this case as “moderately magnetized,” to distinguish from the strongly magnetized case when $\rho_i \ll r_1$. We note that with these values of the magnetic field the electron gyromotion corresponds to the shortest length scale and fastest frequency in the system.

We have performed simulations with CPIC for the systems just described using a 128×128 grid with time step $\omega_{pe}\Delta t = 0.05$. For the moderately magnetized case, we note the important fact that this choice of the time step under-resolves the electron gyromotion ($\omega_{ce}\Delta t \approx 1$). A PIC code with a standard leap-frog particle mover would be very inaccurate and possibly numerically unstable for these parameters. As we have argued above, the reason why we can afford such large time step is that for a uniform magnetic field CPIC uses the cyclotronic particle integrator [32], [33] and only needs to resolve accurately the dynamics associated with the electric field.

For the weakly magnetized case, the floating potential averaged over the last sixth of the simulation is $\phi_{\text{sphere}}/T_e \simeq -2.51$, slightly lower than the one obtained for the unmagnetized plasma. In addition, the shielding potential remains symmetric (not shown). For the moderately magnetized case, the floating potential averaged over the last sixth of the simulation is $\phi_{\text{sphere}}/T_e \simeq -2.43$, further lowered relative to the weakly magnetized case. In addition, the shielding potential is not symmetric anymore (not shown).

To interpret the results just discussed, we have studied the trajectories of test particles moving in the equilibrium configuration obtained from CPIC. We have injected 500 000 particles for each species, uniformly located on the outer boundary (in the region defined by $\cos[\pi(1-\eta)] \geq 0.8$) and with injection velocities distributed according to a Maxwellian, and have followed them until they either hit the sphere or leave the system. Fig. 6 shows the number of particles that were able to hit the sphere in terms of their initial angular distribution on the outer boundary $\cos(\delta)$, with

$$\delta = \pi(1 - \eta). \quad (30)$$

The parameter δ is the colatitude in the spherical coordinate system with axis along the magnetic field: $\cos \delta = 0$

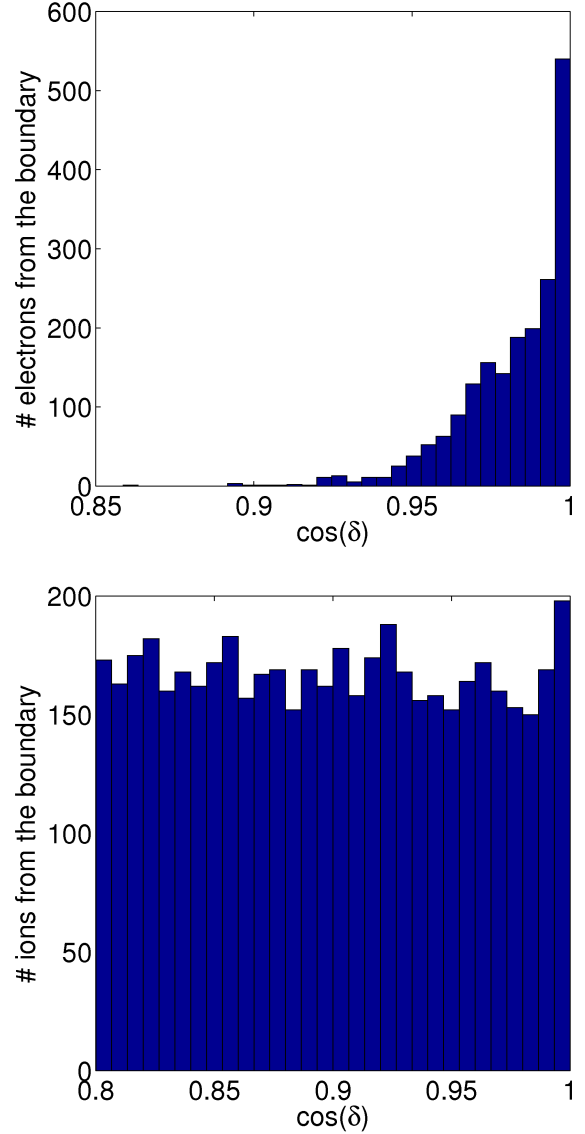


Fig. 6. Test C, $B_{z0} \simeq 1.9$: Outer boundary angular distribution, $\cos(\delta) = \cos[\pi(1 - \eta)]$, of the test particles that hit the sphere. Initially 500 000 test particles for each species were injected from the outer boundary with Maxwellian distribution functions. The test particles were distributed uniformly on the boundary in the region defined by $\cos[\pi(1 - \eta)] \geq 0.8$.

corresponds to the equatorial plane, while $\cos \delta = \pm 1$ is the sphere axis. The plot in Fig. 6 is obtained for the weakly magnetized case ($B_{z0} \simeq 1.9$). One can clearly observe that the ions (bottom panel) can reach the sphere isotropically from the outer boundary. On the other hand, the majority of electrons (top panel) are collected in a flux tube of characteristic radius $r_{\text{flux}} \simeq r_1 + \rho_e$ directed along the magnetic field, for which $\cos(\delta) \simeq 0.98$. However, it is interesting to note that some electrons are collected in a larger flux tube. For instance, $\cos(\delta) = 0.9$ corresponds to a radius $r \simeq 4.4$. These are electrons that have higher initial velocities, toward the tail of the Maxwellian distribution function, and therefore have a large gyroradius.

The diagnostic just described is complemented with the distribution of particle collection on the surface of the sphere. This is shown in Fig. 7 with data obtained directly from CPIC over the last tenth of each simulation in Tests B and C. The top

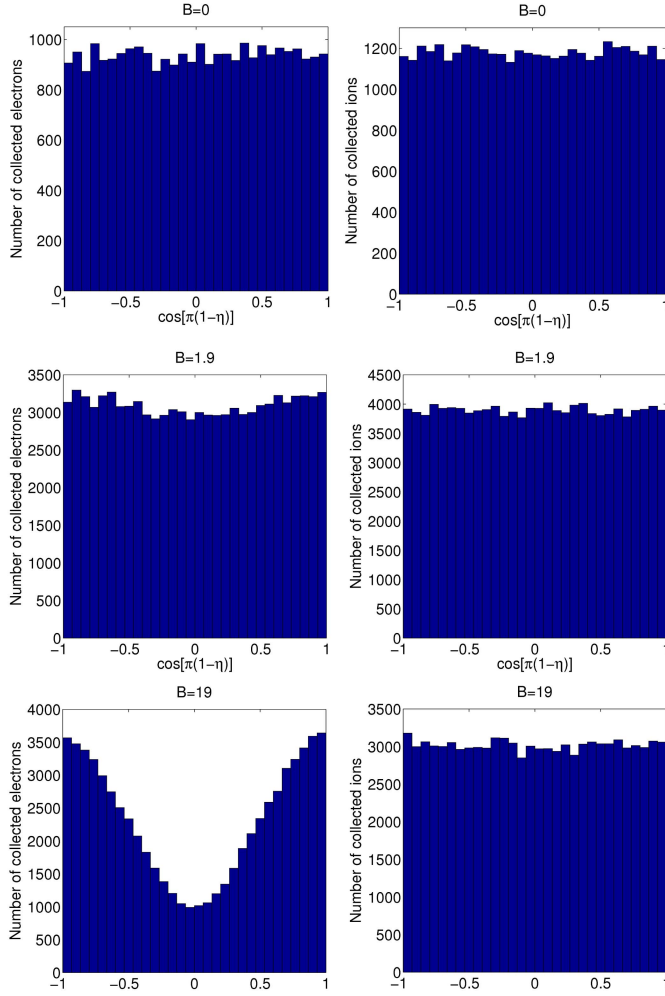


Fig. 7. Test C: Angular distribution, $\cos(\delta) = \cos[\pi(1 - \eta)]$, of particles collected by the sphere over the last tenth of each simulation performed for Tests B and C.

panels are for the unmagnetized case, the middle panels for the weakly magnetized case and the bottom panels for the moderately magnetized case. As expected, when the plasma is unmagnetized, both electrons and ions are collected isotropically on the sphere. When the plasma is weakly magnetized the ions are still collected isotropically, since $\rho_i \gg r_1$. The electrons start to show a small depletion around $\cos(\delta) = 0$. Interestingly, even though the electrons are collected primarily in a flux tube centered around the sphere with characteristic radius of a few electron thermal gyroradii, they can still hit the sphere quite isotropically since their gyroradius is comparable with the sphere radius. Therefore, the shielding potential remains symmetric. For the moderately magnetized case ($B_{z0} \simeq 19$), $\rho_e \ll r_1$ and the electron motion is severely constrained by the magnetic field. Therefore, the electrons do not hit the sphere isotropically, and there is a large depletion in the number of electrons around $\cos(\delta) \approx 0$. The ions, on the other hand, still hit the sphere isotropically since $\rho_i > r_1$.

As a final comment, we note that these results are consistent with the findings in [18]. For weakly to moderately magnetized plasmas, the electron collection is restricted by the magnetic field and the resulting electron current is lower relative to the unmagnetized case. Therefore, the floating

potential on the sphere becomes less negative. Quantitatively, [18] gives $\phi_{\text{sphere}}/T_e = -2.41$ ($B_{z0} = 0$), $\phi_{\text{sphere}}/T_e = -2.37$ ($B_{z0} = 1.9$), and $\phi_{\text{sphere}}/T_e = -2.21$ ($B_{z0} = 19$), which is in reasonable agreement (within $\sim 10\%$) with our results.

IV. CONCLUSION

We have described CPIC: a fully kinetic, electrostatic, body-fitted PIC code in curvilinear geometry for structured grids. CPIC was designed with the goal of flexibility (namely the ability to handle different geometries, complex computational domains, and the interaction of plasmas with complex objects with characteristic size smaller than the characteristic plasma length scales), robustness, and performance. It introduces a coordinate transformation from the physical space to the logical space, where the grid is uniform and Cartesian. In CPIC, most of the operations are performed in logical space. Its main features are: 1) the use of structured grids; 2) a scalable and robust field solver based on the BoxMG algorithm; and 3) an hybrid particle mover, where the particles' position is updated in logical space while the particles' velocity is updated in physical space. In our tests, the hybrid mover has proven more efficient and robust than the conventional physical space mover. We have presented some successful benchmark tests involving the interaction of a plasma with material boundaries, in Cartesian and spherical geometries, with and without magnetic field.

ACKNOWLEDGMENT

The authors wish to thank Luis Chacon, Chris Fichtl, John M. Finn, Rao Garimella, Richard Marchand, and Xianzhu Tang.

REFERENCES

- [1] C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation*. New York, NY, USA: Taylor & Francis, 2004.
- [2] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. New York, NY, USA: Taylor & Francis, 1988.
- [3] J. Brackbill and D. Forslund, "An implicit method for electromagnetic plasma simulation in two dimensions," *J. Comput. Phys.*, vol. 46, no. 2, pp. 271–308, May 1982.
- [4] G. Chen, L. Chacon, and D. C. Barnes, "An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm," *J. Comput. Phys.*, vol. 230, no. 18, pp. 7018–7036, Jan. 2011.
- [5] S. Markidis and G. Lapenta, "The energy conserving particle-in-cell method," *J. Comput. Phys.*, vol. 230, no. 18, pp. 7037–7052, Aug. 2011.
- [6] J.-F. Roussel, "Spacecraft plasma environment and contamination simulation code: Description and first tests," *J. Spacecraft Rockets*, vol. 35, no. 2, pp. 205–211, Mar. 1998.
- [7] L. Patacchini and I. H. Hutchinson, "Spherical conducting probes in finite debye length plasmas and $E \times B$ fields," *Plasma Phys. Controlled Fusion*, vol. 53, no. 2, pp. 025005-1–025005-3, Feb. 2011.
- [8] J.-L. Vay, P. Colella, J. Kwan, P. McCorquodale, D. Serafini, A. Friedman, *et al.*, "Application of adaptive mesh refinement to particle-in-cell simulations of plasmas and beams," *Phys. Plasmas*, vol. 11, no. 5, pp. 2928–2934, 2004.
- [9] T. Westermann, "Numerical modelling of the stationary Maxwell-Lorentz system in technical devices," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 7, no. 1, pp. 43–67, Jan./Feb. 1994.
- [10] T. Westermann, "Particle-in-cell simulations with moving boundaries—Adaptive mesh generation," *J. Comput. Phys.*, vol. 114, no. 2, pp. 161–75, Oct. 1994.
- [11] C.-D. Munz, R. Schneider, E. Sonnendrucker, E. Stein, U. Voss, and T. Westermann, "A finite-volume particle-in-cell method for the numerical treatment of Maxwell-Lorentz equations on boundary-fitted meshes," *Int. J. Numer. Methods Eng.*, vol. 44, no. 4, pp. 461–87, Feb. 1999.

- [12] T. Westermann, "Localization schemes in 2D boundary-fitted grids," *J. Comput. Phys.*, vol. 101, no. 2, pp. 307–313, Aug. 1992.
- [13] J. W. Eastwood, W. Arter, N. J. Brealey, and R. W. Hockney, "Body-fitted electromagnetic PIC software for use on parallel computers," *Comput. Phys. Commun.*, vol. 87, nos. 1–2, pp. 155–178, May 1995.
- [14] C. A. Fichtl, J. M. Finn, and K. L. Cartwright, "An arbitrary curvilinear-coordinate method for particle-in-cell modeling," *Comput. Sci. Discovery*, vol. 5, no. 1, pp. 0140111–0140111–25, Jan. 2012.
- [15] J. Wang, D. Kondrashov, P. Liewer, and S. Karmesin, "Three-dimensional deformable-grid electromagnetic particle-in-cell for parallel computers," *J. Plasma Phys.*, vol. 61, no. 3, pp. 367–389, Apr. 1999.
- [16] M. J. Mandell, V. A. Davis, D. L. Cooke, A. T. Wheelock, and C. J. Roth, "Nascap-2k spacecraft charging code overview," *IEEE Trans. Plasma Sci.*, vol. 34, no. 5, pp. 2084–2093, Oct. 2006.
- [17] J.-F. Roussel, F. Rogier, G. Dufour, J.-C. Mateo-Velez, J. Forest, A. Hilgers, *et al.*, "SPIS open-source code: Methods, capabilities, achievements, and prospects," *IEEE Trans. Plasma Sci.*, vol. 36, no. 5, pp. 2360–8, Oct. 2008.
- [18] R. Marchand, "PTetra, a tool to simulate low orbit satellite-plasma interaction," *IEEE Trans. Plasma Sci.*, vol. 40, no. 2, pp. 217–229, Feb. 2012.
- [19] T. Muranaka, S. Hosoda, J.-H. Kim, S. Hatta, K. Ikeda, T. Hamanaga, *et al.*, "Development of multi-utility spacecraft charging analysis tool (MUSCAT)," *IEEE Trans. Plasma Sci.*, vol. 36, no. 5, pp. 2336–49, Oct. 2008.
- [20] S. P. MacLachlan, J. M. Tang, and C. Vuik, "Fast and robust solvers for pressure-correction in bubbly flow problems," *J. Comput. Phys.*, vol. 227, pp. 9742–9761, Dec. 2008.
- [21] J. E. Dendy, "Black box multigrid," *J. Comput. Phys.*, vol. 48, no. 3, pp. 366–386, 1982.
- [22] J. E. Dendy, "Black box multigrid for periodic and singular problems," *Appl. Math. Comput.*, vol. 25, pp. 1–10, Jan. 1988.
- [23] J. E. Dendy and J. D. Moulton, "Black box multigrid with coarsening by a factor of three," *Numer. Linear Algebra Appl.*, vol. 17, nos. 2–3, pp. 577–598, 2010.
- [24] V. D. Liseikin, *Grid Generation Methods*. New York, NY, USA: Springer-Verlag, 1999.
- [25] A. Winslow, "Adaptive mesh zoning by the equipotential method," Lawrence Livermore Nat. Lab., Livermore, CA, USA, Tech. Rep. UCID-19062, 1981.
- [26] G. L. Delzanno, L. Chacon, J. M. Finn, Y. Chung, and G. Lapenta, "An optimal robust equidistribution method for two-dimensional grid adaptation based on monge-kantorovich optimization," *J. Comput. Phys.*, vol. 227, no. 23, pp. 9841–9864, 2008.
- [27] *HYPRE High Performance Preconditioners User's Manual, Version 2.9.0b*, Lawrence Livermore Nat. Lab., Livermore, CA, USA, 2012.
- [28] G. Lapenta, "DEMOCRITUS: An adaptive particle in cell (PIC) code for object-plasma interactions," *J. Comput. Phys.*, vol. 230, no. 12, pp. 4679–95, 2011.
- [29] W. L. Briggs, *A Multigrid Tutorial*. Philadelphia, PA, USA: SIAM, 1987.
- [30] P. J. Roache, "Code verification by the method of manufactured solutions," *J. Fluids Eng.*, vol. 124, no. 1, pp. 4–10, 2002.
- [31] J. P. Boris, "Relativistic plasma simulation—Optimization of a hybrid code," in *Proc. 4th Conf. Numer. Simul. Plasmas*, 1970, pp. 3–67.
- [32] L. Patacchini and I. Hutchinson, "Explicit time-reversible orbit integration in particle in cell codes with static homogeneous magnetic field," *J. Comput. Phys.*, vol. 228, no. 7, pp. 2604–2615, Apr. 2009.
- [33] G. L. Delzanno and E. Camporeale, "On particle movers in cylindrical geometry for particle-in-cell simulations," *J. Comput. Phys.*, vol. 253, pp. 259–277, Nov. 2013.
- [34] L. A. Schwager and C. K. Birdsall, "Collector and source sheaths of a finite ion temperature plasma," *Phys. Fluids B*, vol. 2, no. 5, pp. 1057–1068, 1990.
- [35] J. G. Laframboise, "Theory of spherical and cylindrical langmuir probes in a collisionless Maxwellian plasma at rest," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 100, Jun. 1966.
- [36] R. Kennedy and J. Allen, "The floating potential of spherical probes and dust grains. II: Orbital motion theory," *J. Plasma Phys.*, vol. 69, no. 6, pp. 485–506, Dec. 2003.
- [37] H. Mott-Smith and I. Langmuir, "Theory of collectors in gaseous discharges," *Phys. Rev.*, vol. 28, no. 4, pp. 727–763, Oct. 1926.
- [38] G. L. Delzanno, G. Lapenta, and M. Rosenberg, "Attractive potential around a thermionically emitting microparticle," *Phys. Rev. Lett.*, vol. 92, no. 3, pp. 035002-1–035002-3, Jan. 2004.
- [39] J. E. Daugherty, R. K. Porteous, M. D. Kilgore, and D. B. Graves, "Sheath structure around particles in low-pressure discharges," *J. Appl. Phys.*, vol. 72, no. 9, pp. 3934–42, 1992.
- [40] G. L. Delzanno, A. Bruno, G. Sorasio, and G. Lapenta, "Exact orbital motion theory of the shielding potential around an emitting, spherical body," *Phys. Plasmas*, vol. 12, no. 6, pp. 62102-1–62102-4, 2005.

Gian Luca Delzanno received the M.Sc. degree in nuclear engineering and the Ph.D. degree in plasma physics from Politecnico di Torino, Torino, Italy, in 1999 and 2003, respectively.

He was a Post-Doctoral Associate with the Burning Plasma Research Group, Politecnico di Torino, from 2003 to 2004, and he was a Post-Doctoral Associate with the T-15 Plasma Theory Group, Los Alamos National Laboratory, Los Alamos, NM, USA, from 2005 to 2006. Since 2007, he has been a Staff Member with the T-5 Applied Mathematics and Plasma Physics Group, Los Alamos National Laboratory.

Enrico Camporeale received the M.Sc. degree in nuclear engineering from Politecnico di Torino, Torino, Italy, in 2004, and the Ph.D. degree in space plasma physics from the Queen Mary University of London, London, U.K., in 2008.

He was a Post-Doctoral Research Assistant with the Astronomy Unit, Queen Mary University of London, from 2008 to 2010. He was with the Los Alamos National Laboratory, Los Alamos, NM, USA, in 2010, where he currently is a Post-Doctoral Associate with the T-5 Applied Mathematics and Plasma Physics Group.



J. David Moulton received the B.Eng. and M.Eng. degrees in physics from McMaster University, Hamilton, ON, Canada, and the Ph.D. degree in applied mathematics from the University of British Columbia, Vancouver, BC, Canada.

He is a Team Leader with the Applied Mathematics and Plasma Physics Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA, and he is the Multi-Process HPC Simulator Thrust lead in the Advanced Simulation Capability for Environmental Management Program. His

current research interests include advanced discretizations and multilevel algorithms for flow and reactive transport in highly heterogeneous porous media. In addition he is investigating new approaches to data assimilation and uncertainty quantification.

Joseph E. Borovsky received the B.S. degree in astronomy and mathematics from the University of Michigan, Ann Arbor, MI, USA, in 1974, and the M.S. and Ph.D. degrees in physics from the University of Iowa, Iowa City, IA, USA, in 1980 and 1981, respectively.

He was a Staff Member with the Los Alamos National Laboratory, Los Alamos, NM, USA, from 1982 to 2012. He is currently a Senior Research Scientist with the Space Science Institute and an Adjunct Research Scientist with the University of Michigan.

Elizabeth A. MacDonald received the B.S. degree in physics from the University of Washington, Seattle, WA, USA, and the Ph.D. degree in physics from the University of New Hampshire, Durham, NH, USA.

She is a Team Leader with the Los Alamos National Laboratory (LANL), Los Alamos, NM, USA. She is involved in plasma mass spectrometer instrumentation and data analysis for magnetospheric missions. She has been with LANL since 2005. Her current research interests include wave-particle interactions and the effect of plasma on radiation belt dynamics, mapping, coupling, and transport between the ionosphere and the inner magnetosphere, and the impact of heavy ions on geomagnetic storm processes.

Michelle F. Thomsen received the B.A. degree from Colorado College, Colorado Springs, CO, USA, in 1971, and the M.S. and Ph.D. degrees from the University of Iowa, Iowa City, IA, USA, in 1974 and 1977, respectively.

She was a Post-Doctoral Research Associate with the University of Iowa from 1977 to 1980 and the Max Planck Institute for Aeronomy from 1980 to 1981. From 1981 to 2009, she was a Staff Member with the Los Alamos National Laboratory (LANL), Los Alamos, NM, USA. She was a Laboratory Fellow in 1997. She is currently involved in contract to LANL.