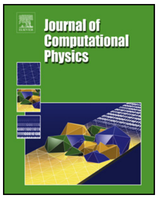
Journal of Computational Physics 350 (2017) 796–823

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



An electrostatic Particle-In-Cell code on multi-block structured meshes

Collin S. Meierbachtol [a,*], Daniil Svyatskiy [b], Gian Luca Delzanno [b], Louis J. Vernon [c], J. David Moulton [b]

[a] *Intelligence and Space Research Division, Los Alamos National Laboratory, Los Alamos, NM, USA*
[b] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA*
[c] *Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, USA*

ARTICLE INFO

Article history:
Received 9 January 2017
Received in revised form 6 September 2017
Accepted 9 September 2017
Available online 14 September 2017

Keywords:
Particle-In-Cell
Multi-block
Curvilinear
Plasma–material interactions

ABSTRACT

We present an electrostatic Particle-In-Cell (PIC) code on multi-block, locally structured, curvilinear meshes called Curvilinear PIC (CPIC). Multi-block meshes are essential to capture complex geometries accurately and with good mesh quality, something that would not be possible with single-block structured meshes that are often used in PIC and for which CPIC was initially developed. Despite the structured nature of the individual blocks, multi-block meshes resemble unstructured meshes in a global sense and introduce several new challenges, such as the presence of discontinuities in the mesh properties and coordinate orientation changes across adjacent blocks, and polyjunction points where an arbitrary number of blocks meet. In CPIC, these challenges have been met by an approach that features: (1) a curvilinear formulation of the PIC method: each mesh block is mapped from the physical space, where the mesh is curvilinear and arbitrarily distorted, to the logical space, where the mesh is uniform and Cartesian on the unit cube; (2) a mimetic discretization of Poisson's equation suitable for multi-block meshes; and (3) a hybrid (logical-space position/physical-space velocity), asynchronous particle mover that mitigates the performance degradation created by the necessity to track particles as they move across blocks. The numerical accuracy of CPIC was verified using two standard plasma–material interaction tests, which demonstrate good agreement with the corresponding analytic solutions. Compared to PIC codes on unstructured meshes, which have also been used for their flexibility in handling complex geometries but whose performance suffers from issues associated with data locality and indirect data access patterns, PIC codes on multi-block structured meshes may offer the best compromise for capturing complex geometries while also maintaining solution accuracy and computational efficiency.

1. Introduction

Since its original development [1,2] and swift adaptation to plasmas [3,4] over a half century ago, the Particle-In-Cell (PIC) method has successfully characterized many plasma systems [5–8]. In the PIC method, large numbers of physical particles are represented by a smaller number of macroparticles, which themselves move through a computational domain

---

* Corresponding author.
E-mail address: cmeierbachtol@lanl.gov (C.S. Meierbachtol).

http://dx.doi.org/10.1016/j.jcp.2017.09.016
0021-9991/

discretized by a mesh. Macroparticles bring an intrinsic adaptivity in velocity space and can easily capture complex distribution functions compared to other approaches. They also introduce statistical noise, which converges slowly as typical of Monte Carlo methods. While other methods such as Eulerian–Vlasov (where phase space is discretized with a mesh, [9,10]) or spectral methods (where phase space is discretized with a moment expansion, [11–13]) are under active development, PIC methods remain by far the most popular in the plasma physics community.

The majority of PIC codes employ uniform, Cartesian meshes and explicit time-stepping techniques, and must therefore resolve the shortest and fastest scales in the system for numerical stability [5,6]. These stringent numerical stability constraints imply that significant computational resources might be necessary for typical plasma physics problems with strong scale separation. Moreover, the multi-scale nature of problems where plasmas interact with material objects can become even more severe due to the scales of the physical objects coming into play. From the perspective of spatial scales, and for objects with features that are much smaller than the characteristic scales of the plasma, a uniform-mesh PIC approach would have serious limitations since the mesh spacing would necessarily be set by the object features. For example, an orbiting spacecraft with characteristic scale on the order of meters (and possibly even smaller features such as antennas) interacts with plasma in the magnetosphere evolving over several kilometers.

These considerations call for PIC methods employing non-uniform, adaptive meshes. Indeed, these methods have been developed in the plasma physics community, and follow the classic mesh generation and adaptation techniques of h-refinement ([14–17] increasing the local resolution of the mesh near important solution features), and r-refinement ([18–25] shifting vertices of the mesh to positions that improve solution accuracy).

Although these PIC methods have been applied to a variety of problems, the challenge of accurately representing objects in plasma–material interaction problems has led to a variety of specialized implementations in the realm of electrostatic PIC codes. Some approaches associate a fraction of the (possibly non-uniform) cells with the object [26–29]. For instance, Hutchinson [26] uses non-uniform meshes and locally modifies the stencil of the discretized Poisson operator to account for charge absorption on the surface of the object. Han et al. [29] developed an immersed finite-element approach with non-homogeneous flux jump boundary conditions at the object interface to account for charge accumulation. Lapenta [27] employed the immersed boundary method, where an object is represented by static particles that carry material properties. Object particles are then interpolated on the mesh in the same way as plasma particles, such that the field equations are solved on a mesh that also includes the object.

In contrast, meshes that conform exactly to the surface of the object such that the object becomes a boundary of the simulation domain, have been used by several researchers [15,19,21,23,25,30,31]. These meshes are often referred to as *conformal* or *body-fitted* meshes. Here both structured meshes (where the mesh has a predefined connectivity pattern) and unstructured meshes (where the mesh carries no predefined ordering, and where a connectivity map is necessary) are possible. Some structured-mesh approaches [25] have been limited to very simple geometries because these methods use connectivity defined by a single logically structured mesh on the global domain. Indeed, it is impossible to create a mesh that conforms to an arbitrarily complex object using this logically structured single-block approach. Furthermore, even with relatively simple objects, it can be difficult to maintain good mesh quality. Instead, single-block unstructured mesh approaches, which leverage the additional freedom of unstructured mesh connectivity, have been used to study the interaction of (quite complex) object geometries with plasmas [21,23].

Unstructured meshes give enormous flexibility for mesh generation and have become the method of choice in the computational fluid dynamics community. For this reason, a variety of mesh-generation tools are available and can be used with relative ease. Unstructured meshes, however, present several disadvantages relative to structured meshes in an electrostatic PIC code. First, the field solver is significantly slower (particularly on modern multi-core architectures) because of indirect memory access patterns that characterize unstructured meshes (versus the direct access patterns and minimal data movement of structured meshes). To be more concrete, MacLachlan et al. [32] studied a two-phase flow problem where bubbles or droplets of one phase move against the background of the other phase. The computational cost of this problem is often dominated by the solution of a Poisson equation with discontinuous coefficients. They compared different approaches, including geometric and algebraic multigrid methods. These two approaches mimic the difference in memory access patterns typical of structured (geometric multigrid) and unstructured (algebraic multigrid) and offer a performance comparison between the two. The conclusion of MacLachlan et al. [32] was that the geometric multigrid approach was a factor 5–10 faster. Second, PIC particles moving through an unstructured mesh must be tracked at all times to find their location in a given cell. Since we have not found any performance results on tracking algorithms on unstructured meshes in the literature, we have implemented the unstructured-mesh tracking algorithm discussed in Ref. [33] and compared it to the algorithm described in this paper. Our preliminary results indicate that our algorithm is at least a factor of 5 faster, confirming that indirect memory access patterns and tracking create a significant overhead for PIC on unstructured meshes. These considerations suggest that coupling the PIC method with multi-block structured meshes could deliver an optimal algorithm that maintains the mesh flexibility necessary for complex geometries, while achieving better computational performance.

This paper presents the development of an electrostatic PIC code on multi-block structured meshes. These meshes are structured at the level of the individual blocks but unstructured in a global sense and present new challenges that are not encountered in single-block structured meshes, such as discontinuities in the mesh properties and mesh orientation changes across adjacent blocks and the presence of polyjunction points where an arbitrary number of blocks meet. While few works on PIC with multi-block structured meshes exist in the literature [19,30,31,34], this paper differentiates from previous works by combining (1) the use of a curvilinear formulation of the PIC method, where for each block a map is introduced from