Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 214 (2006) 96–121
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids

G.B. Jacobs, J.S. Hesthaven *

Division of Applied Mathematics, Brown University, Box F, Providence, RI 02912, United States

Received 28 December 2004; received in revised form 10 September 2005; accepted 12 September 2005
Available online 27 October 2005

Abstract

We present a high-order particle-in-cell (PIC) algorithm for the simulation of kinetic plasmas dynamics. The core of the algorithm utilizes an unstructured grid discontinuous Galerkin Maxwell field solver combining high-order accuracy with geometric flexibility. We introduce algorithms in the Lagrangian framework that preserve the favorable properties of the field solver in the PIC solver. Fast full-order interpolation and effective search algorithms are used for tracking individual particles on the general grid and smooth particle shape functions are introduced to ensure low noise in the charge and current density. A pre-computed levelset distance function is employed to represent the geometry and facilitates complex particle–boundary interaction. To enforce charge conservation we consider two different techniques, one based on projection and one on hyperbolic cleaning. Both are found to work well, although the latter is found be too expensive when used with explicit time integration. Examples of simple plasma phenomena, e.g., plasma waves, instabilities, and Landau damping are shown to agree well with theoretical predictions and/or results found by other computational methods. We also discuss generic well known problems such as numerical Cherenkov radiation and grid heating before presenting a few two-dimensional tests, showing the potential of the current method to handle fully relativistic plasma dynamics in complex geometries.

Keywords: High-order methods; Unstructured grids; Discontinuous Galerkin methods; Particle-in-cell; Plasma physics; Maxwell equations

1. Introduction

The reliable, accurate, and efficient computational modeling of plasma dynamics remains very challenging. Not only do problems involving plasmas span from the smallest (atomistic particle–particle dynamics) to the largest scales like solar flares and galaxy dynamics, but there is also a strong interaction between the many scales and the long range electromagnetic forces. Moreover, the range of applications is very broad, e.g., fusion energy, both by means of magnetic confinement and laser ignited devices; high-power microwave generation;

* Corresponding author. Tel.: +1 401 863 2671; fax: +1 401 853 1355.
E-mail addresses: gjacobs2@dam.brown.edu (G.B. Jacobs), Jan.Hesthaven@brown.edu (J.S. Hesthaven).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.09.008

large scale particle accelerators; and a variety of plasma based technology. This warrants that significant resources be spent on the development of accurate, robust, and efficient tools for the modeling of such plasma problems.

The direct solution of many problems could in principle be accomplished by solving the Maxwell–Vlasov or Poisson–Vlasov equation in a fully adaptive fashion. However, its $(6 + 1)$-dimensional nature keeps such direct modeling out of reach for problems in complex geometries and of realistic complexity in general.

In the last few decades particle-in-cell (PIC) methods have proved valuable as an approach for the modeling of a variety of plasma problems, e.g., microwave and fusion devices, and astrophysical problems. Typically, in this approach, essentially solving the Maxwell–Vlasov equation by a Lagrangian approach, one solves the Maxwell equations and/or a Poisson equation on a Cartesian grid with a second order finite difference method or a Fourier spectral method [1]. The particles are forced by the fields and tracked in a Lagrangian framework. Subsequently, the particles are coupled with the field solver by weighing the sum of the Coulomb forces onto the grid.

A number of methods based on such ideas have been developed, mainly with the aim of improving on the computational efficiency and flexibility of PIC. In particular, the exact charge conserving scheme proposed in [3] is widely used since it eliminates the need to directly impose Gauss' Law to ensure charge conservation. In [4] this technique is extended to include a multi-block body-fitted finite element method with the aim of increasing geometric flexibility. Umeda et al. [5] suggest a zigzag particle trajectory to improve upon the computational efficiency in [3], while in [6] the time step restriction of explicit methods is tackled with an implicit Maxwell solver. All of these methods are second order accurate in space and time and are, in most cases, restricted to simple Cartesian or block structured geometries.

As powerful as these techniques are, their limitations are beginning to emerge as bottlenecks in the modeling of large scale phenomena and devices, e.g., high-frequency, high-power microwave generation and propagation, and high-frequency particle accelerator modeling [2]. In particular, the reliance on a simple staggered Cartesian grid in the Maxwell finite difference time-domain solver severely limits the geometric flexibility as well as the accuracy of the method. This latter restriction causes considerable problems when large scale problems are being considered, since the second order accuracy results in significant dispersion errors unless prohibitively fine grids are being used.

The classic finite difference based methods also suffer from a number of pathological problems intimately linked to the properties of the overall scheme. As discussed in [1] the use of simple particle shape functions leads to strong aliasing and enhanced numerical grid heating. Although filtering can reduce this finite grid instability [7], a more reasonable way to overcome grid heating is through the use of smoother particle shapes. This would, however, destroy many of the desirable properties of the original scheme like for example exact charge conservation and is therefore not used. Furthermore, the inherent dispersion properties of the finite difference approximation result in an artificial numerical Cherenkov radiation when modeling highly relativistic problems [8]. This purely numerical pollution of the solution is the result of the concave nature of the field solver's numerical dispersion relation, causing fast waves to propagate slower than is physically correct. In highly relativistic problems the unphysical interaction of fast particles with these waves creates a Cherenkov radiation. The only way to address this in a systematic way is to choose a scheme with a strictly convex dispersion relation.

This paper presents the first step in the development of a PIC method which has the potential to effectively address all of the shortcomings of the existing methods. The computational kernel for the field solver in the PIC method is the discontinuous Galerkin field solver developed for the time-domain Maxwell equations [10]. This method secures geometric flexibility through a fully unstructured body-fitted grid, arbitrary order of accuracy, inherent but controllable high-frequency dissipation through fluxes, and excellent stability properties. Furthermore, the dispersion relation of the discontinuous Galerkin method is strictly convex [9], effectively eliminating the source of numerical Cherenkov radiation as discussed above. The scheme has been tested extensively for pure 2D and 3D electromagnetic problems and shown itself to be highly efficient, accurate, and robust.

The particle mover relies on high-order interpolation, efficient local search algorithms to locate the particles, and a level set approach to represent geometries, enabling elastic/inelastic particle interactions with complex geometrical boundaries. Charge and current redistribution computations use smooth weight functions, enabling a significant decrease in the number of particles needed in a computational cell and reducing the finite grid instability as compared to the simple redistribution schemes typically used. Divergence cleaning is done

either through an advective approach, maintaining a very high-parallel efficiency, or through a classical projection scheme leading to the solution of a Poisson equation. Particles as well as fields are all being advanced in time with a high-order explicit Runge–Kutta method. As we shall illustrate through a number of computational experiments and benchmarks, this initial stage of the development suggests that the general algorithm has the potential to be a successful and powerful tool for the modeling of general plasma kinetics.

The remainder of this paper is organized as follows. In Section 2, we recall the basic governing equations while Section 3 contains a detailed discussion of the many elements of the algorithm, both for the field advancement and the particle dynamics. In this section, we also discuss particle shape functions and charge conservation schemes. This sets the stage in Section 4 for a number of numerical experiments, both of a simple nature as well as a few more complex two-dimensional examples. This section also contains a discussion of the finite grid instability and its behavior and control in the current algorithm. Section 5 contains a few concluding remarks as well as a number of suggestions for future work along the lines initiated here.

2. The physical model

For modeling of the fields, we consider the two-dimensional Maxwell equations in normalized vacuum TE form written in conservation from

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{F} = \boldsymbol{J}, \tag{1}$$

$$\nabla \cdot \boldsymbol{E} = \rho, \tag{2}$$

where $\boldsymbol{q} = [E_x, E_y, B_z]^{\mathrm{T}}$, $\boldsymbol{F} = [F_x, F_y], F_x = [0, -B_z, -E_y]^{\mathrm{T}}$, $F_y = [B_z, 0, E_x]^{\mathrm{T}}$, and $\boldsymbol{J} = [J_x, J_y, 0]^{\mathrm{T}}$. Throughout $\boldsymbol{E}$, $\boldsymbol{B}$, $\boldsymbol{J}$ and ρ represent the electric field, the magnetic field, the current, and the charge density, respectively, while the subscripts identify the direction of the vector field variable. Vacuum permittivity, permeability and the speed of light, c, are used for normalization of Eq. (1). A reference length, L_f, normalizes space and time as $\boldsymbol{x} = \tilde{\boldsymbol{x}}/L_f$ and $t = \tilde{t}/(L_f/c)$, respectively. In this normalization, the vacuum speed of light is one.

Particles are described in a purely Lagrangian manner using

$$\frac{\mathrm{d}\boldsymbol{x}_{\mathrm{p}}}{\mathrm{d}t} = \boldsymbol{v}_{\mathrm{p}}, \tag{3}$$

$$\frac{\mathrm{d}m\boldsymbol{v}_{\mathrm{p}}}{\mathrm{d}t} = q(\boldsymbol{E} + \boldsymbol{v}_{\mathrm{p}} \times \boldsymbol{B}), \tag{4}$$

where $\boldsymbol{x}_{\mathrm{p}}$ and $m\boldsymbol{v}_{\mathrm{p}}$ denote the non-dimensional particle coordinate and momentum, respectively, with q and m representing the particle charge and mass. For high-speed plasma the relativistic correction applies to m as $m = m_0/\sqrt{(1 - |\boldsymbol{v}_{\mathrm{p}}|^2)}$, where m_0 is the mass at rest.

The particles, represented by q, $\boldsymbol{x}_{\mathrm{p}}$, and $m\boldsymbol{v}_{\mathrm{p}}$, couple to the fields through the space charge, ρ, and the induced current density, $\boldsymbol{J}$, as

$$\rho(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{p}}} q_i S(|\boldsymbol{x}_{\mathrm{p}} - \boldsymbol{x}|), \tag{5}$$

$$\boldsymbol{J}(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{p}}} q_i \boldsymbol{v}_i S(|\boldsymbol{x}_{\mathrm{p}} - \boldsymbol{x}|). \tag{6}$$

Here i is particle index and N_{p} is the total number of particles. $S(|\boldsymbol{x}_p - \boldsymbol{x}|)$ is a particle weighing function that represent the manner in which the individual particle charge is distributed in space. We shall discuss its exact meaning and form in more detail later.

3. The numerical approach

As simple as the problem description given above may appear, its formulation as a numerical scheme is far from trivial. The main complication is found in the simple observation that the fields are described in

an Eulerian frame while the charge dynamics more naturally is discussed in a purely Lagrangian setting. A computational approach will need to effectively connect these two essentially different pictures. In the following, we shall discuss in some detail the individual components of the algorithm.

3.1. The field solver

To advance Maxwell's equations, Eq. (1), in time we shall use a nodal high-order discontinuous Galerkin method, described in detail in [10]. In this approach, the computational domain, Ω, is subdivided into non-overlapping triangular elements, D, to ensure geometric flexibility. On each element, we assume that the local solution can be represented as an nth order polynomial of the form

$$\boldsymbol{q}_N(\boldsymbol{x},t) = \sum_{j=1}^{N} \boldsymbol{q}(\boldsymbol{x}_j,t) L_j(\boldsymbol{x}) = \sum_{j=1}^{N} \hat{\boldsymbol{q}}_j(t) L_j(\boldsymbol{x}), \tag{7}$$

where L_j is the genuine multi-dimensional Lagrange interpolant associated with the N grid points, $\boldsymbol{x}_j$, on the triangular element. In this work, we use the nodes given in [11]. For an nth order polynomial, we have

$$N = \frac{(n+1)(n+2)}{2}$$

as the number of local grid points or degrees of freedom on each element for each variable.

To seek equations for these N local unknowns, we require the local approximate solution, $\boldsymbol{q}_N$, to Maxwell's equations to satisfy

$$\int_D \left(\frac{\partial \boldsymbol{q}_N}{\partial t} + \nabla \cdot \boldsymbol{F}_N - \boldsymbol{J}_N \right) L_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \oint_{\partial D} L_i(\boldsymbol{x}) \hat{\boldsymbol{n}} \cdot [\boldsymbol{F}_N - \boldsymbol{F}^*] \, \mathrm{d}\boldsymbol{x}. \tag{8}$$

Here, $\boldsymbol{F}^*$ signifies a numerical flux and $\hat{\boldsymbol{n}}$ is an outward pointing unit vector defined at the boundary of the element. The role of the numerical flux is to connect the elements and ensure stability of the computational scheme. If the numerical flux is consistent, the scheme is clearly consistent. On the other hand, boundary/interface conditions are not imposed exactly but rather weakly through the penalizing surface integral. Within this multi-element context, the formulation is inherently discontinuous and yields, through its very construction, a highly parallel local scheme.

With the operators,

$$\hat{M}_{ij} = \int_D L_i L_j \, \mathrm{d}\boldsymbol{x}, \quad \hat{S}_{ij} = \int_D \nabla L_j L_i \, \mathrm{d}\boldsymbol{x}, \quad \hat{F}_{ij} = \oint_{\partial D} L_i L_j \, \mathrm{d}\boldsymbol{x}, \tag{9}$$

we recover from Eq. (8) the fully explicit local scheme,

$$\hat{M} \frac{\mathrm{d}\hat{\boldsymbol{q}}}{\mathrm{d}t} + \hat{S} \cdot \hat{\boldsymbol{F}} - \hat{M} \hat{\boldsymbol{J}} = \hat{F} \hat{\boldsymbol{n}} \cdot [\hat{\boldsymbol{F}} - \hat{\boldsymbol{F}}^*], \tag{10}$$

where $\hat{\boldsymbol{q}}$ represents the $3N$-vector of nodal values, $\boldsymbol{q}_N$, at D. Similarly, $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{J}}$, and $\hat{\boldsymbol{F}}^*$ denote nodal values for the flux, the current density, and the numerical flux, respectively.

To finalize the formulation of the scheme, we must specify the numerical flux $\boldsymbol{F}^*$, which is responsible for passing information between the elements and imposing the boundary conditions. Given the linearity of Maxwell's equations, we use a flux like

$$\hat{\boldsymbol{n}} \cdot [\boldsymbol{F} - \boldsymbol{F}^*] = \begin{cases} \boldsymbol{n} \times (\gamma \boldsymbol{n} \times [\boldsymbol{E}] - [\boldsymbol{B}]), \\ \boldsymbol{n} \times (\gamma \boldsymbol{n} \times [\boldsymbol{B}] + [\boldsymbol{E}]), \end{cases} \tag{11}$$

where $[\boldsymbol{Q}] = \boldsymbol{Q}^- - \boldsymbol{Q}^+$ measures the jump in the values across an interface. Superscript '+' refers to the value from the neighbor element while superscript '−' refers to field value local to the element. Note that by taking $\gamma = 1$, one recovers the classic, dissipative, upwind flux [12], while $\gamma = 0$ leads to a purely dispersive central flux. Clearly one is free to take values in between these two extremes with α controlling the amount of dissipation added. A complete analysis in terms of accuracy and stability of the scheme above can be found in [10] with further details in [9].

The set of linear ODE's in (10) is integrated with the low storage, fourth order Runge–Kutta scheme (RK4) from Carpenter and Kennedy [13],

$$\left.\begin{aligned} \boldsymbol{w}_i &= \alpha_i \boldsymbol{w}_{i-1} + \Delta t \boldsymbol{F}(t_{i-1}, \boldsymbol{q}^{(i-1)}) \\ \boldsymbol{q}^{(i)} &= \boldsymbol{q}^{(i-1)} + \beta_i \boldsymbol{w}_i \end{aligned}\right\}, \quad i = 1, 2, \ldots, s, \tag{12}$$

where $\alpha_1 = 0$ for the algorithm to be self-starting, $\boldsymbol{q}^{(0)} = \hat{\boldsymbol{q}}^{n-1}$, $\boldsymbol{q}^{(s)} = \hat{\boldsymbol{q}}^{n}$, and $t_i = t^{n-1} + c_i \Delta t$. This is a $2N$ storage scheme, since only $\boldsymbol{q}$ and $\boldsymbol{w}$ require storage. Compared to a classic 4th order Runge–Kutta method the memory usage is reduced by half. The current scheme is a five stage method with the coefficients [13] given by

$$\begin{aligned} \alpha_1 &= 0.0, & \beta_1 &= 0.1496590219993, & c_1 &= 0.0, \\ \alpha_2 &= -0.4178904745, & \beta_2 &= 0.3792103129999, & c_2 &= 0.1496590219993, \\ \alpha_3 &= -1.192151694643, & \beta_3 &= 0.8229550293869, & c_3 &= 0.3704009573644, \\ \alpha_4 &= -1.697784692471, & \beta_4 &= 0.6994504559488, & c_4 &= 0.6222557631345, \\ \alpha_5 &= -1.514183444257, & \beta_5 &= 0.1530572479681, & c_5 &= 0.9582821306748. \end{aligned} \tag{13}$$

3.2. Tracking the particles

Lagrangian tracking of the particles consists of three stages per particle, including searching the element a particle is located in, interpolating the field variables to the particle location, and pushing the particle forward with a time integration method.

In [14] a tracking algorithm is discussed in a system that only couples the field equations to the particles in one direction, e.g., passive advection. It was shown that interpolation and temporation integration method may, in most cases, be of a lower order than the approximation order of the spatial and temporal discretization method used for the field equations. From numerous tests, however, it has become clear that only full order interpolation using Eq. (7) suffices in a system that fully couples particles and field equations in both directions. In full order interpolation the interpolating polynomial order is equal to the order of the polynomial used to represent the fields.

A fast full-order interpolation technique is discussed in Appendix A. Lower order interpolation severely influences the accuracy of the total scheme and may lead to instability in many situations. Similarly, the time scheme for integration of Eq. (4) should be the same as the time scheme that integrates the field equations, i.e., Eq. (12).

The particle localization scheme follows [14], where the particle's element is found by comparing the mapped particle coordinate to the coordinates of the standard element for each element on the grid. The mapping takes advantage of the simple inverse of the linear blending formula for triangles. For the smooth mapping of a straightsided triangle the inverse is given as,

$$\boldsymbol{\xi}_{\mathrm{p}} = C_1 \boldsymbol{x}_{\mathrm{p}} + \boldsymbol{C}_2, \tag{14}$$

where $\boldsymbol{\xi}_{\mathrm{p}}$ is the mapped coordinate. Matrix C_1 and vector $\boldsymbol{C}_2$ contains constants that are functions of the triangles' vertex coordinates. Even though the elements are typically large, scanning all elements in a large grid is prohibitive. We reduce the cost dramatically by storing information about the elements connected to one node, and scan only these elements if a particle leaves the element close to this node. Since high-order elements are typically large and we are considering purely explicit time stepping here, particles do in general not leave the bounds of this cloud of elements.

3.3. Weighing of the particles to the grid

To connect the fields and the particles, we must translate the action of the particles to the Eulerian grid using Eq. (6). Computational efficiency and accuracy suggest that the particles be treated as clouds rather than points [1]. Thus the shape function S in Eq. (6) is not chosen as a Coulomb distribution, but commonly as a compact distribution spanning approximately the area of a grid cell.

Classic particle-in-cell (PIC) methods [1] usually weigh with a zero or first order function, which is not suitable for a high-order method as the lack of smoothness of the particle shape results in a Gibbs type phenomenon that severely influences accuracy and introduces noise in ρ and $\boldsymbol{J}$. The non-smooth shape is also more likely to enhance the well-known finite grid heating and instability [1]. Thus, an unstructured grid high-order method requires a different approach, in which smoothness is desirable.

In the volume weighing approach [1] the interpolation function is different from the shape function. Assigning a particle according to this approach to a high-order element is difficult unless we use linear weighing. Rather, we choose to assign a smooth shape function to the grid directly. Thus, in the approach developed here, the shape function is the interpolation function.

We compare four potential smooth shape functions. These include a raised cut-off cosine function,

$$S_{\cos} = \frac{\pi}{R^2(\pi^2-4)}\left[\cos\left(\frac{r\pi}{R}\right)+1\right], \tag{15}$$

where $r = |\boldsymbol{x} - \boldsymbol{x}_p|$ is the Eulerian distance from the center of the particle cloud, and R is the influence radius of the cloud. (15) is normalized such that that integral $\int_0^{2\pi}\int_0^R S_{\cos} r\,\mathrm{d}r\,\mathrm{d}\phi = 1$, where ϕ is the azimuthal coordinate. However, the odd derivatives are not zero at $r = R$, leading to unfavorable behavior as we shall see shortly.

Secondly, we consider a Gaussian shape function,

$$S_{\text{gauss}} = \frac{1}{2\pi\epsilon^2}\mathrm{e}^{\frac{-r^2}{2\pi\epsilon^2}}, \tag{16}$$

where ϵ is the well-known variance. The spatial unboundedness of the Gaussian function does not suit the finite nature of a particle cloud, but in practice the Gaussian is zero to machine precision at a radius of five to seven times the variance and can be cut off. The integral of S_{gauss} is again unity, the Gaussian is analytic and, unlike the cosine, its derivative is approximately zero at the cutoff radius.

Thirdly, we consider the polynomial function,

$$S_{\text{pol}} = \frac{1}{2\pi A}\left[1-(2p+1)!/(p!)^2\int_0^{r/R}[\tau(1-\tau)]^p\,\mathrm{d}\tau\right], \quad r = 0,\ldots,R, \tag{17}$$

This function is p differentiable and has a finite radius, R. The parameter, A, ensures that the integral over its surface is unity (for example for $p = 4$, $A = 3R^2/22$). The first $p/2$ derivatives of S_{pol} are zero at $r = R$. If p (mostly $p = 4$) is set, the seemingly expensive evaluation of S_{pol} is relatively inexpensive, as the integral reduces to a few multiplications.

Finally, we consider the polynomial function,

$$S_{\text{poll}} = \frac{\alpha+1}{\pi R^2}\left[1-\left(\frac{r}{R}\right)^2\right]^{\alpha}, \quad r = 0,\ldots,R, \tag{18}$$

which is likewise smooth and has a unit integral. The evaluation of this function is the least expensive of the above functions. Note that all functions are isotropic as opposed to the rectangular cloud shape commonly used in standard PIC codes.

Fig. 1 plots the three distribution functions versus the radial coordinate. S_{gauss} is plotted for a cut-off at $R = 5\epsilon$ and $R = 7\epsilon$. S_{pol} is plotted for $p = 4$ and 6. S_{poll} is plotted for $\alpha = 10$ and 20. The Gaussian and S_{poll} show similar trends. Note that S_{poll} does not require a cut-off, whereas the Gaussian does. S_{gauss} and S_{poll} have larger maxima (at $r = 0$) than the cosine and S_{pol}, as they decay to zero faster when r goes to R. The cosine and polynomial function distribute their weight more evenly over the cloud influence area. The non-zero value of the cosine higher order derivatives is evident from the large slope the cosine has toward $r = R$ compared to the polynomial function. If p increases, the weight of the polynomial concentrates at $0 < r < R/2$ and the function goes to zero more rapidly for $r > R/2$ to ensure that the higher order derivatives vanish.

An accurate representation of $S(r)$ requires multiple interpolation points. In fact, approximating a single particle shape to order $\mathcal{O}(10^{-3})$ is found to require a number of grid points per particle that varies from 150 to 500 depending on the approximation order. This number is high, i.e., computationally expensive for

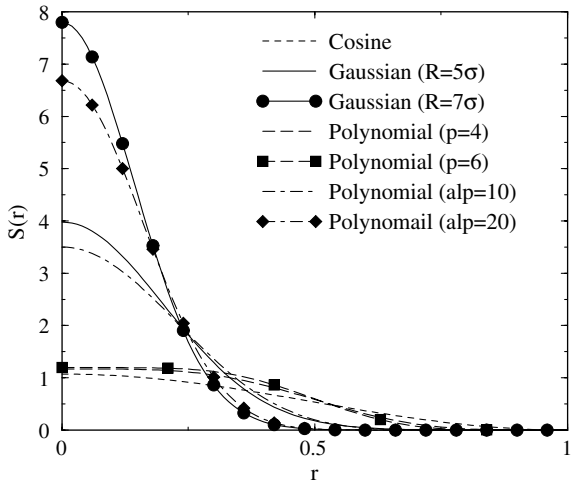


Fig. 1. Gaussian, cosine, polynomial Eq. (17) ($p = 4, 6$), and polynomial Eq. (18) ($\alpha = 10,20$) shape functions, $S(r)$, plotted versus the radial coordinate, r.

PIC simulations in plasmas that generally require a particle number that varies from a couple of hundred into the millions. In plasma simulations, however, the accurate representation of one particle is secondary to the accurate representation of the charge and current density by many particles.

To obtain an indication of the accuracy of the charge density, ρ, we consider particles being positioned equidistantly on a square domain with 148 elements. This particle distribution yields a constant charge density over the domain, which should be $\rho_{\text{exact}} = N_{\text{p}} q/\text{Area}$, where Area is the domain area. The deviation of the computed value from ρ_{exact} indicates the accuracy. Fig. 2 plots this deviation for the shape functions against

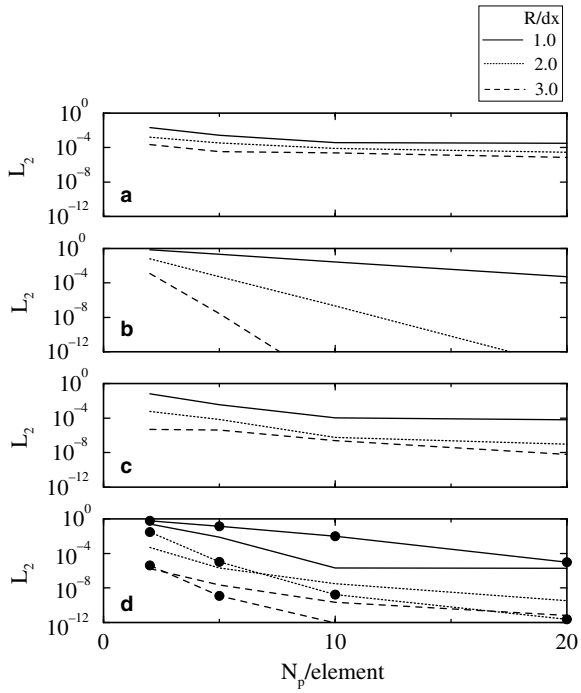


Fig. 2. The L_2 approximation error of ρ for a (a) raised cosine, (b) Gaussian, (c) polynomial ($p = 4$), and (d) polynomial1 with ($\alpha = 3$ and 10 (with filled circle)) shape function plotted versus the number of particles, N_{p}, per element with approximation order $N = 4$ at various ratios of the distribution radius, R, to a typical grid spacing, dx.

the number of particles per element at various $R/\mathrm{d}x$, the values of which are considerably smaller (fewer points per particles) than mentioned above. Indeed, all shape functions exhibit greater accuracy than what the individual particle approximation suggests. The accuracy of the Gaussian (Fig. 2(b)) is more dependent on N_p/element and $R/\mathrm{d}x$ than the accuracy of S_pol (Fig. 2(b)) and the cosine (Fig. 2(c)). The dependency of the accuracy of S_poll is influenced by α. For large values of α, S_poll behaves more like the Gaussian and for small α the function behaves like the cosine and S_pol.

For $R/\mathrm{d}x > 1$ and N_p/element > 5 the Gaussian and S_poll with large α are by far the most accurate. The polynomial is more accurate than the cosine for all $R/\mathrm{d}x$ and N_p/element plotted.

The cosine function, Eq. (15), generally performs poorly, both for the approximation of a single particle and of ρ and should not be considered. The choice between S_poll with large α (or Gaussian) or S_poll with small α (or S_pol) is less obvious and depends on simulation parameters. If there are many elements that contain less than five particles and if $R/\mathrm{d}x \sim 1$ (computational efficient, but not very accurate), then S_poll with small α is favorable. However, if the objective is to minimize noise in ρ at the cost of computational efficiency a larger α is preferred. The ability of S_poll, Eq. (18), to behave like a Gaussian as well as S_pol depending on α in addition to its computational efficiency makes S_poll the preferred particle distribution function and we shall use it subsequently.

We weigh particles throughout the domain with a constant particle cloud size, R, independent of the element size. Particles are weighed only to the elements that are influenced by the particle's distribution function, i.e., the elements for which $r < R$. These elements are identified for all vertices in a pre-processing stage. The closest vertex to a particle then provides the lookup table for the elements for the weighing. This weighing procedure may lead to a larger number of elements to be weighed per particle than established methods. One could reduce the number of weighing elements through a varying R. This, however, may have an impact on the accuracy of ρ and introduces a compressible particle that violates charge conservation. With constant R these problems are not present and the computational overhead is minimized by having local lookup tables to identify element regions. Nevertheless, for highly non-uniform grids, a range of particle cloud sizes is clearly desirable and we are currently exploring this important next stage of the development.

3.4. Charge conservation techniques

Gauss's law, Eq. (2), is generally not satisfied with the weighing technique described in the previous section and a correction to the electric field is required. To this end we consider two techniques, comprising a projection method [1] and a hyperbolic cleaning technique proposed in [15].

3.4.1. Global Poisson correction

In the classic projection, one expresses the computed field, $\boldsymbol{E}^*$, as

$$\boldsymbol{E}^* = \boldsymbol{E} + \nabla\phi,$$

where $\nabla \cdot \boldsymbol{E} = \rho$. This represents a Helmholtz decomposition of the computed field, $\boldsymbol{E}^*$, into a charge conserving component and a gradient, $\nabla\phi$.

This immediately yields

$$\nabla^2\phi = \nabla \cdot \boldsymbol{E}^* - \rho \quad \text{with } \phi = 0 \text{ on } \partial\Omega, \tag{19}$$

where the boundary conditions come from $\boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{n} \times \nabla\phi = 0$, i.e., $\phi =$ constant along the boundary. As this constant has no importance, we are free to choose homogeneous conditions.

Once ϕ is found, the electric field is corrected as

$$\boldsymbol{E} = \boldsymbol{E}^* - \nabla\phi, \tag{20}$$

where $\boldsymbol{E}$ now is the corrected electric field.

Consistent with the Maxwell solver the Poisson equation is solved with a local discontinuous element scheme at every Runge–Kutta stage. We rewrite the Poisson equation into a system of two first-order equations,

$$\nabla \cdot \boldsymbol{q} = f, \quad \boldsymbol{q} = \nabla\phi. \tag{21}$$

This is discretized exactly as Maxwell's equations, leading to

$$\hat{S}\cdot\hat{\boldsymbol{q}} = \hat{M}\hat{\boldsymbol{f}} + \hat{F}\hat{\boldsymbol{n}}\cdot[\hat{\boldsymbol{q}} - \hat{\boldsymbol{q}}^*],$$
$$\hat{M}\hat{\boldsymbol{q}} = \hat{S}\hat{\phi} - \hat{F}\hat{\boldsymbol{n}}[\hat{\phi} - \hat{\phi}^*].$$

Note that $\hat{\boldsymbol{q}}$ is a purely local variable. Using a standard approach [16] we use the stabilized central fluxes

$$\hat{\phi}^* = \frac{1}{2}(\hat{\phi}^- + \hat{\phi}^+), \quad \hat{\boldsymbol{q}}^* = \frac{1}{2}(\hat{\boldsymbol{q}}^- + \hat{\boldsymbol{q}}^+) - \tau[\hat{\phi}],$$

where $\tau \propto n^2/h$ is a stabilization parameter, scaling with the order, n, and grid size, h. Note that other choices of the numerical fluxes are possible and may lead to improved behavior [16]. It should also be noted that this approach may lead to a reduced accuracy in $\boldsymbol{E}$ as ϕ is computed to $\mathcal{O}(h^{n+1})$ and, thus, $\nabla\phi$ to $\mathcal{O}(h^n)$ [16]. For low order elements this is likely to affect the overall accuracy of the scheme.

The Poisson solver has the additional advantage that it can be used to model low speed plasma physics. In this case the Maxwell equations are not solved and only the Poisson solver is used for updating the fields. Implementation of the Poisson solver for high-speed problems, however, leads to unphysical instantaneous effects, that can affect the global solution unfavorably.

3.4.2. Hyperbolic cleaning – the χ-method

As an alternative to the projection technique, we consider divergence control by a hyperbolic cleaning method (we shall also refer to it as the χ-method) described in [15]. In this approach a correction potential is introduced into the Maxwell equation (1) as a Lagrangian multiplier. In the strictly hyperbolic formulation [15], Maxwell's equations, Eq. (1), are altered to

$$\begin{aligned} &\frac{\partial \boldsymbol{q}}{\partial t} + \nabla\cdot\tilde{\boldsymbol{F}} = \boldsymbol{J}, \\ &\frac{\partial \phi}{\partial t} = \chi(\rho - \nabla\cdot\boldsymbol{E}) - \varepsilon\phi, \end{aligned} \tag{22}$$

with the modified fluxes of $\tilde{\boldsymbol{F}} = [\tilde{F}_x, \tilde{F}_y], \tilde{F}_x = [\chi\phi, -B_z, -E_y]^{\mathrm{T}}$, $\tilde{F}_y = [B_z, \chi\phi, E_x]^{\mathrm{T}}$. Here ε is a damping constant. Eq. (22) is a strictly hyperbolic system of equations, with four characteristic velocities, $\lambda_{1,4} = \chi$, $-\chi$ and $\lambda_{2,3} = -c$, c, where $c = 1$ is the normalized speed of light. Compared to the uncorrected Maxwell equations, two characteristics, propagating information at speed, χ, are added, and one characteristic with zero velocity is omitted, hence eliminating the DC component of the system.

The χ characteristics effectively reduce the divergence error by propagating it away at velocity χ. Taking $\chi \gg c$ implies that the divergence error be swept out of the domain very rapidly, effectively imposing Gauss' law. As χ approaches infinity, one recovers the original equation (1).

The fully explicit formulation of Eq. (22) allows for the space discretization to be unaltered from Eq. (10) with the understanding that $\hat{\boldsymbol{q}}$, $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{J}}$ are adjusted for Eq. (22). The only significant change is the numerical flux of Eq. (11) which changes to

$$\boldsymbol{n}\cdot[\boldsymbol{F} - \boldsymbol{F}^*] = \begin{cases} \boldsymbol{n}\times(\gamma\boldsymbol{n}\times[\boldsymbol{E}] - [\boldsymbol{B}]) + \chi\boldsymbol{n}(\boldsymbol{n}\cdot[\boldsymbol{E}] - [\phi]), \\ \boldsymbol{n}\times(\gamma\boldsymbol{n}\times[\boldsymbol{B}] + [\boldsymbol{E}]), \\ \chi(\boldsymbol{n}\cdot[\boldsymbol{E}] - [\phi]). \end{cases} \tag{23}$$

One can prove accuracy and stability of the modified system following the approach in [10].

3.4.3. Comparison of Poisson correction and Lagrangian multiplier techniques

The Poisson projection correction requires a global solve, which appears expensive at first. In contrast, the χ-method is local and seemingly cheaper. However, the explicit time step in the χ-method is proportional to $1/\chi$ making the equations stiffer and more expensive for an increased χ. So, while increasing χ improves the physical significance of the modified equations (22), it also makes them more expensive.

Simple operation counts will show that the χ-method require 1/3 extra work per time step as compared to the simple Maxwell's equations, originating from the need to differentiate ϕ. On the other hand, a Maxwell's

time step with the Poisson correction applied at every Runge–Kutta stage is ~3–4 times slower than the χ-method, when using a sparse direct solver for the Poisson solve. Thus, even with a relatively small (physically inaccurate) value of $\chi > 3$–4, the χ-method may be slower than the Poisson correction approach. On the other hand, the loss of accuracy in the latter method may become a problem when computing at low order and/or in an *hp*-environment.

The ease by which the χ-method can be parallelized is highly attractive and clearly advantageous over the Poisson solver. However, to make the χ-method sufficiently robust and physically correct, we find that high values of χ are needed, e.g., $\chi > 10$. In this case an implicit field solver will be needed, reintroducing the need for global solves. Another consideration is that in case of PIC, the computationally expensive two-way particle weighing may well dominate all other elements. In such a case, it is preferable that the field solver requires a minimum number of times where particles are weighed over a given simulation time, i.e., a large time step. In conclusion, it is not at all clear which method is to be favored. We find that having both at the disposal is perhaps the best approach and we shall continue to consider both techniques.

3.5. Particles and boundary conditions

Particle boundary conditions are needed for the tracking of the particles and the weighing of the particles to the grid. This section discusses these boundary conditions separately.

3.5.1. Boundary conditions for particle tracking

A particle can react inelastic, fully elastic, and partially elastic with a boundary. The inelastic boundary condition simply removes the particle from the computational domain. The elastic boundary conditions require knowledge of the angle of the particle path with the wall and the distance of the particle from the wall so that the reflected path can be determined. We use a levelset [17,18] to represent the distance normal to the boundaries and its accompanying direction from which the particle's angle and distance to the wall can be deduced.

To compute the levelset, γ, we seek the steady state solution to

$$\frac{\partial \gamma}{\partial \tau} + \boldsymbol{w} \cdot \nabla \gamma, = \operatorname{sgn}(\gamma_0) + \mu \Delta \gamma. \tag{24}$$

Here γ represents a distance function and $\boldsymbol{w} = \operatorname{sgn}(\gamma_0)\frac{\nabla \gamma}{|\nabla \gamma|}$ signifies the signed normal. τ is an artificial time and $\operatorname{sgn}(x)$ is the classic sign-function. Here, γ_0 is the known geometrical boundary and specifies the initial condition for γ. μ is a diffusion coefficient that is ideally zero. The left side of Eq. (24) is a Hamilton–Jacobi equation whose characteristics have unity velocity and a normal direction to the geometrical boundary. The right-hand side introduces a minor stabilizing diffusion.

The numerical solution of (24) follows the method outlined above. Spatial derivatives are determined as

$$\nabla \hat{\gamma} = \hat{S}\hat{\gamma} - \hat{F}\boldsymbol{n}[\hat{\gamma} - \hat{\gamma}^*], \tag{25}$$

where $\hat{S}$ and $\hat{F}$ are defined in Eq. (9). The numerical flux, $\hat{\gamma}^*$, is a simple central flux. Applying Eq. (25) twice gives second derivatives. The levelset equation is integrated with a Runge–Kutta scheme until steady state is reached. This is done in a preprocessing stage and values for γ and $\boldsymbol{w}$ are stored at every grid point.

The elastic boundary condition uses the two last particle coordinates, $\boldsymbol{x}_{\mathrm{p}}^{(1)}$ and $\boldsymbol{x}_{\mathrm{p}}^{(2)}$, where at stage (1) the particle is still within the computational bounds and at stage (2) the particle has crossed the boundary. With the local interpolation of γ and $\boldsymbol{w}$ to $\boldsymbol{x}_{\mathrm{p}}^{(1)}$ the direction and distance of the particle relative to the wall is known. The reflected particle coordinate, $\boldsymbol{x}_{\mathrm{p}}^{(3)}$, may be determined as sketched in Fig. 3. Stages (1) and (2) determine the distance and direction of the particle track as $s = |\boldsymbol{x}_{\mathrm{p}}^{(1)} - \boldsymbol{x}_{\mathrm{p}}^{(2)}|$ and $\boldsymbol{n}_{\mathrm{p}} = (\boldsymbol{x}_{\mathrm{p}}^{(1)} - \boldsymbol{x}_{\mathrm{p}}^{(2)})/s$, respectively. From $\boldsymbol{n}_{\mathrm{p}} = [n_{\mathrm{p},x}, n_{\mathrm{p},y}]^{\mathrm{T}}$ and $\boldsymbol{w} = [w_x, w_y]^{\mathrm{T}}$ we obtain a normal vector, $\boldsymbol{m} = [m_x, m_y]^{\mathrm{T}}$, with the components

$$m_x = n_{\mathrm{p},y} w_x - n_{\mathrm{p},x} w_y, \quad m_y = n_{\mathrm{p},x} w_x + n_{\mathrm{p},y} w_y,$$

that can be used to extract the angle, α. The distance s_3 follows from $s_1 = |\gamma/m_y|$, $s_2 = s - s_1$ and $s_3 = s_2 m_y$. The reflected particle coordinate is computed with $\boldsymbol{x}_{\mathrm{p}}^{(3)} = \boldsymbol{x}_{\mathrm{p}}^{(2)} - \zeta \boldsymbol{w} s_3$. Here, ζ determines the elasticity of the reflection. If $\zeta = 2$ the reflection is fully elastic. Determination of the particle velocity at stage (3) follows a similar procedure.

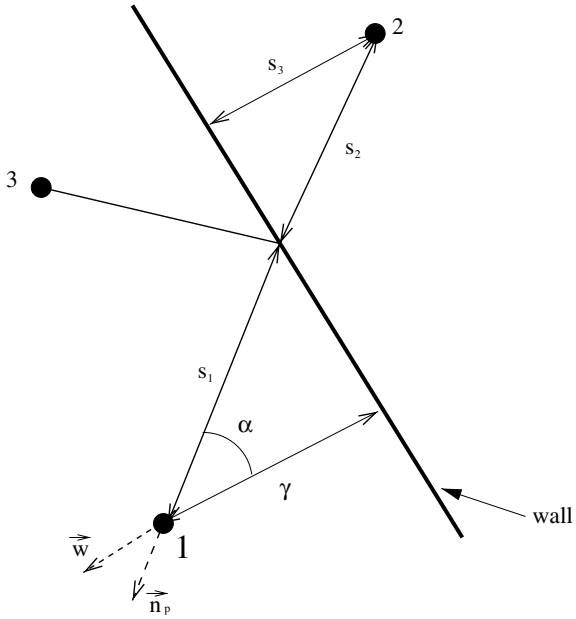


Fig. 3. Schematic for determination of the particle coordinate after a fully elastic collision with the wall.

With the explicit Runge–Kutta time integration, Eq. (12), only the particle coordinate at stage (2) is known. To obtain the particle coordinate at stage (1) we linearly track the particle back with the approximate local time step of a Runge–Kutta stage assuming constant particle velocity. If the RK4 stage is larger than one, $s > 1$, the residuals $\boldsymbol{w}_i$ and $\boldsymbol{q}^{(i-1)}$ in Eq. (12) are recomputed assuming constant particle velocity and a linear path from stage (2) to (3). Using different time steps, like the simulation time step, or not recomputing the residuals may lead to instability.

3.5.2. Boundary conditions for particle weighing

We assign particle clouds to the grid using a constant radius assignment function. If a particle cloud is near a boundary its weighing area crosses the boundary and part of the assignment function will not be projected to the grid which affects ρ and $\boldsymbol{J}$. Depending on the type of boundary encountered, a correction is required to model the physical boundary condition. For example, a conducting wall places a mirror particle with opposite charge, $q = -q$ on the other side of the wall. The mirror distance is readily determined from the levelset γ and $\boldsymbol{w}$.

3.6. Filtering

Filtering is used to reduce noise in fields and enhance robustness of the algorithm. Low diffusion spectral methods can benefit from it to the point where filtering stabilizes the simulation. We apply filtering in two instances. If the influence area of the particle $R/\mathrm{d}x < 1$, ρ and $\boldsymbol{J}$ are filtered for noise reduction. Secondly, we enhance the Poisson solver in geometries with sharp corners. The filtering is applied on the right side of Gauss's law equation (2).

We have used a standard exponential filter, adding a little dissipation to the scheme as possible. The details of the filter, its impact and implementation as a matrix operator are discussed in [19]. As we shall see shortly, the use of this filter is not needed in many applications and, as some of the tests show, it is effect is minimal beyond making the general computational approach more robust.

4. Numerical examples

In the following, we shall present a number of tests, first of a simple character to simply validate several components of the scheme and, subsequently, to model a variety of basic, yet essential plasma phenomena.

4.1. Tests of the scheme components

We shall begin by validating the particle tracking, the reflecting boundary condition, and the charge conservation methods.

Computation of the Larmor radius and $\boldsymbol{E}\times\boldsymbol{B}$ drift confirm the fourth order accuracy of the scheme (not shown here). We take unity fields, $q/m = 1$ and initialize with unity particle velocity. Comparison of the computed and exact particle velocity for the Larmor radius in Fig. 4(a) illustrates the slightly dissipative nature of the RK4 scheme. The $\boldsymbol{E}\times\boldsymbol{B}$ drift (Fig. 4(b)) shows a slight growth in error over time as can be expected [10].

We validate the fully elastic particle boundary condition by releasing a circular array of particles in a circular geometry with an outward velocity normal to the wall. The grid consist of straight sided elements. Fig. 5 shows the initial circular particle array and accompanying velocity vector before and after interaction with the wall. Ideally, the particle array after reflection should be a circular array, but, as can be seen from the figure, the straight elements reflect the particle not exactly normal to a circle. Near the corners there is a coagulation of particles and the velocity vector does not point to the center of the circle. A finer grid or a high-order boundary-fitted grid would reduce this effect.

The dipole simulation presented in [15] is reproduced to assess the accuracy of the Poisson and hyperbolic divergence cleaning techniques. The simulation is performed on a 16 by 16 square domain with 572 elements. One hundred particles are released in a circular array with radius 1.28 in the center of the domain. A constant magnetic field, $B_z = 1$, drives the particles with $q/m = -0.195$, $q = -3.86\times10^{-5}$ in a circle with velocity, $|\boldsymbol{v}_{\mathrm{p}}| = 0.25$. The influence radius of the particle cloud is $R = 0.90$. Fig. 6 confirms that the hyperbolic cleaning removes excess divergence from the domain faster in time for larger χ. The Poisson projection method (which

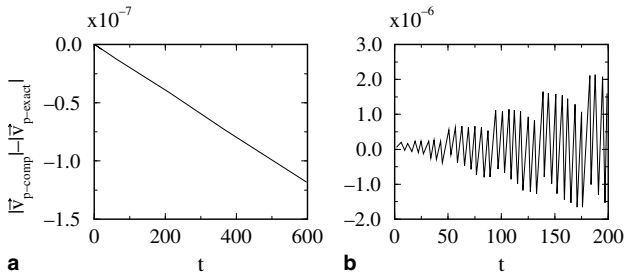


Fig. 4. Comparison of the computed particle velocity, $|\boldsymbol{v}_{\mathrm{p\text{-}comp}}|$, with the exact particle velocity, $|\boldsymbol{v}_{\mathrm{p\text{-}exact}}|$, versus time for (a) the Larmor radius and (b) $\boldsymbol{E}\times\boldsymbol{B}$ drift.

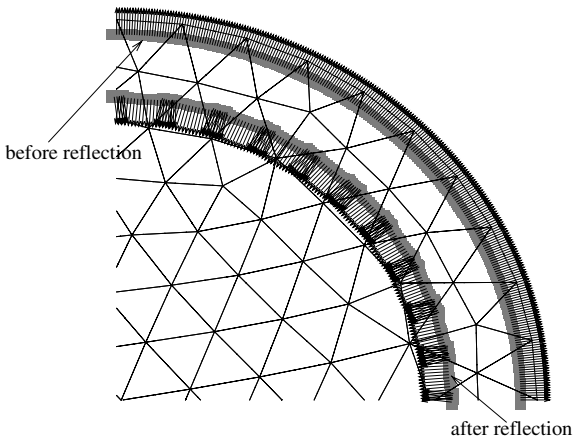


Fig. 5. Particles plotted with their velocity vector before and after interaction with the outer boundary of a circular unstructured grid (first quadrant is shown).

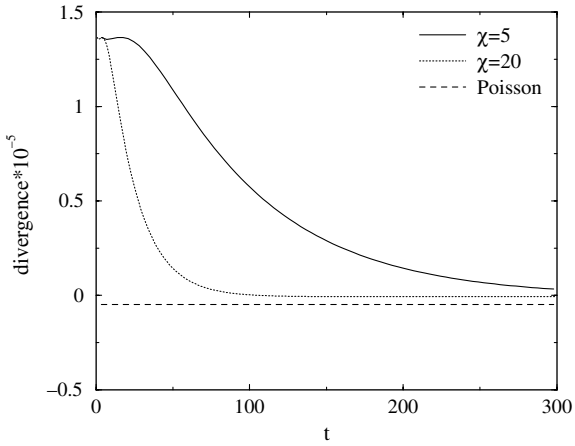


Fig. 6. Comparison of divergence for a dipole computation with the Poisson correction and the Lagrangian multiplier method with $\chi = 5$ and 20.

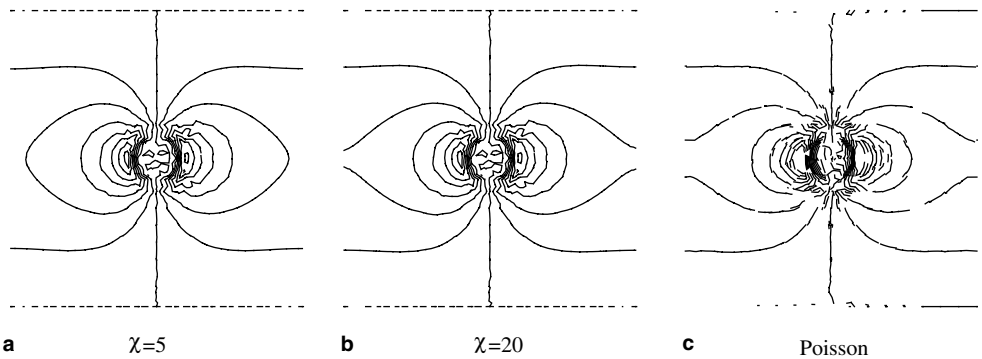


Fig. 7. Comparison of the electrical field, E_x, for a dipole computation with the Poisson correction and the Lagrangian multiplier method with $\chi = 5$ and 20.

has no time dependency on the divergence cleaning for this case) is affected by noise in the charge density more severely than the hyperbolic cleaning, resulting in the slightly larger divergence at steady state ($t > 100$ for $\chi = 20$). A comparison of the contour lines of the electric field in the x-direction, E_x (Fig. 7), shows that the projection method (Fig. 7(c)) smears E_x most. With increasing χ, the results from the hyperbolic cleaning compare better with those obtained by the projection method which can be expected to be less robust for marginally resolved problems.

A further test is to consider the question of self-forcing, i.e., whether a particle is being pushed by its own field. In temporal splitting schemes used in classical FDTD PIC this self-force is averaged away. The scheme presented here has no splitting and should theoretically not suffer from a self force. We test this hypothesis by releasing one particle with zero velocity, $q = 1$ and $q/m = 1$ in the center of a unit square domain with 228 4th order elements and periodic boundary conditions. The particle cloud has radius $R = 0.5$ and $\alpha = 10$. The fields are initially zero. Fig. 8 shows that the particle is initially marginally displaced due to errors in the initial particle assignment. At later times the particle remains at it position with a slight oscillation confirming the absence of any essential self force.

4.2. Finite grid instability

The finite grid heating and resulting instability manifests itself if the Debye length λ_D is underresolved and is caused by aliasing errors in the determination of the non-linear current density. The result is an unphysical

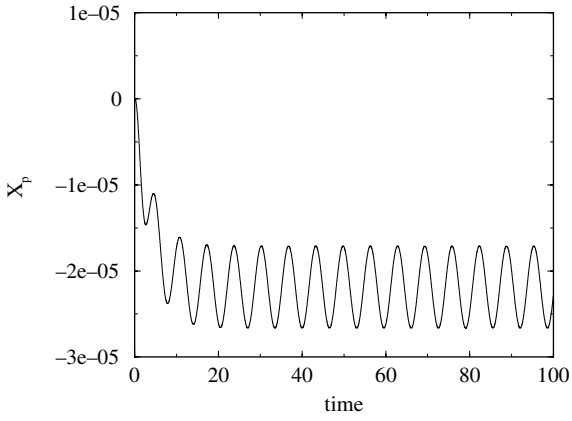


Fig. 8. Particle coordinate in x direction, X_p, plotted versus time for a particle initially at rest in a zero electromagnetic field.

total energy increase, which has been known to trouble classic particle-in-cell methods, in particular for problems with high-density plasmas like for example laser–matter interactions. Theoretically, a smoother weighing function and high-order schemes should suffer less from such errors and thus exhibit a reduced grid heating.

We will test the finite grid instability by simulating an isotropic plasma in a periodic domain with an underresolved Debye length. The severity of the finite grid instability is characterized by simulating for a fixed time and monitoring the total energy increase. A comparison is made with OOPIC [20], a second-order structured finite difference electromagnetic PIC code.

We take the electron number density $n_e = 10^{21}$ cm^{-3} and the electron temperature $T_e = 90$ eV. The resulting Debye length is $\lambda_D = 2.23$ nm. The thermal velocity of the electrons is $v_{the} = 3.97 \times 10^6$ m/s. The side of the computational square has a reference length of $L_f = 10^{-7}$ m. The simulations are run for a time of $T_{final} = 3.3 \times 10^{-4}$ s.

The OOPIC simulations use equidistant grids with 25×25, 50×50, and 100×100 cells and 8 particles per cell. For the coarsest grid the cell size is 18 times the Debye length, i.e., one can expect a significant grid heating. The increase of the total energy in time in Fig. 9 confirms the significant (exponential) growth as well as the reduced grid heating with improved resolution. At the final time the total energy has increased with a factor of 17, 3.7 and 1.5 for the 25×25, the 50×50, and the 100×100 grid, respectively.

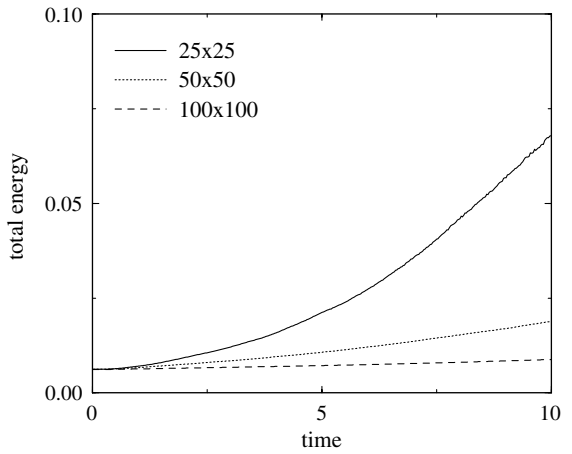


Fig. 9. Comparison of total energy trend for OOPIC simulations of a homogeneous plasma with a 25×25, 50×50, and 100×100 cell grids.

For the high-order PIC method we study the effect of various resolution parameters. As a base we choose a grid with 200 triangles and a fifth order approximation. The grid has 50 grid points on the side of the computational domain. 70×70 particles are released in the domain with a non-dimensional radius $R = 0.1$, which is equal to $L_f/10$ in dimensional units. The power in (18) is $\alpha = 10$. This base case has a resolution comparable to the 50×50 OOPIC case presented above.

Fig. 10 summarizes the effect of various resolution parameters. All cases show a small (<5%) initial drop in total energy, a result of the particles randomizing their coordinates from the initial equidistant release positions. At later times the finite grid instability increases the total energy. For the base case the total energy has increased with a mere 2%. Fig. 10(a) shows that the Poisson divergence cleaner introduces more grid heating than the hyperbolic cleaning approach, possibly caused by its global nature and increased sensitivity to noise.

Changing the value of χ does not have a significant effect on the grid heating, because the constant $\boldsymbol{E}$ for this isotropic plasma simulation is not affected much by the χ correction. Moreover, the finite grid instability is driven by $\boldsymbol{J}$, which is also barely affected by χ in this isotropic case.

Grid (h) refinement (Fig. 10(b)) improves the Debye length resolution and therefore decreases the grid heating. A halved h improves grid heating by an order. Doubling the number of particles in the x and the y direction (Fig. 10(c)) reduces the grid heating by 50%, i.e., a linear effect. It is also observed that with fewer particles the initial total energy reduction is larger. Increasing R (Fig. 10(d)) reduces aliasing and thus grid heating. Changing R from 0.05 to 0.075 and 0.1 reduces grid heating by 1 and 2 orders of magnitude, respectively. Smoothening the particles shape (Fig. 10(e)) by reducing α from 10 to 5 decreases grid heating slightly. Reducing α further to 1 yields a non-smooth linear shape function that doubles the grid heating.

From the above we can conclude that the high-order method can achieve a significant smaller grid heating (in the order of a few percent) compared to OOPIC (a minimum of 50%) for similar resolution. The high-order method is quite sensitive to h and R adjustments. These parameters can be varied independently of each other making the high-order PIC flexible, as opposed to classic PIC methods which couple the two and have a more predictable and moderate grid heating dependency on h refinement. As a final note we should mention that the absence of a total energy increase does not mean that the results are accurate. For example, capturing particle dynamics may require a smaller R or more particles than required to control grid heating.

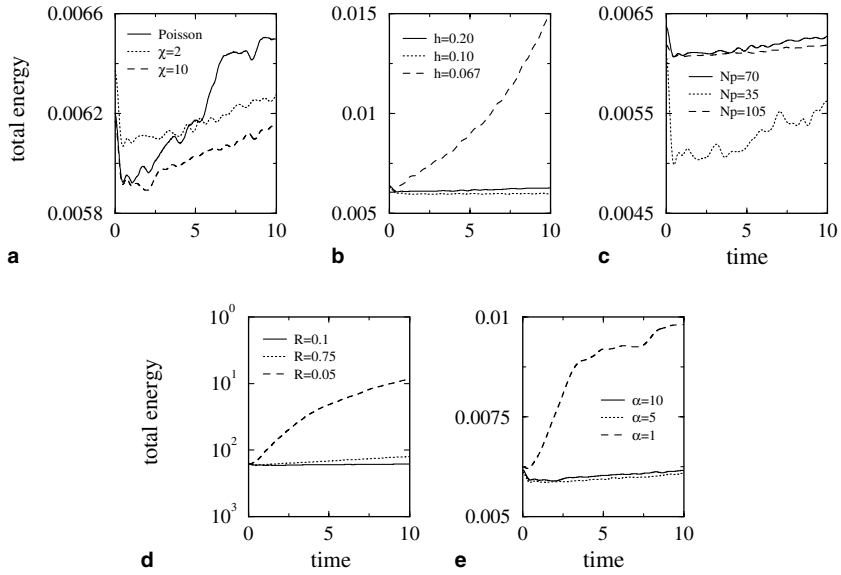


Fig. 10. Total energy plotted versus time for several grid resolution parameters for high-order PIC simulations of a homogeneous plasma. The solid lines signify the base case. (a)–(e) plot the effect of the divergence cleaning method, grid spacing h, number of particles $N_p \times N_p$, particle cloud radius R, and power α of the particle shape function (18), respectively.

This flexibility is a strength which should be explored in the modeling of complex phenomena, although it remains a challenge to estimate when to choose the parameters, i.e., particle size and form, in an optimal manner. This is complicated further by the fact that the role of the particles is both to represent to spatial charge and model the distribution in velocity space.

4.3. One-dimensional plasma tests

As a first dynamic test, we simulate essentially 1D plasma cases, including a plasma wave, a 2-stream instability, and linear Landau damping. The results are compared with the XES1 code [1] which is a 1D electrostatic solver that solves a Poisson equation using a spectral Fourier method to obtain the electric field. To compare with our method, we solve only for the electric field and set the current densities to zero.

For all three cases, the computational domain has a length of 2π in the x-direction of the plasma wave propagation. In the y-direction the grid has length 1.5 and is meshed with approximately two triangles so as to simulate a 1D setting, i.e., a full two-dimensional solver is used in this case. The total number of elements is 62. In the y-direction, 25 particles are distributed equidistantly to emulate a 1D setting. We use a particle. equation (18), with $\alpha = 10$. No filtering is applied in any of the cases below. We set $N = 64$ for the XES1 Fourier spectral method and use no compensation or filtering [1].

4.3.1. Plasma wave

320 particles are equidistantly distributed with a superimposed one-dimensional sine deviation

$$x = x_{\text{eq}} + A\sin(kx_{\text{eq}}), \tag{26}$$

where the amplitude of the deviation is $A = 0.001$ and the wavenumber $k = 2$. The cloud influence area is $R = 0.5$. Physical parameters for the particle are $q = 0.001177$ and $q/m = 1.0$. Fig. 11 shows that the total energy (a), the kinetic energy(b) and the potential energy of the plasma wave computed with the DG Poisson solver are in excellent agreement with XES1. XES1 predicts a slightly lesser total energy, but both methods preserve energy equally well.

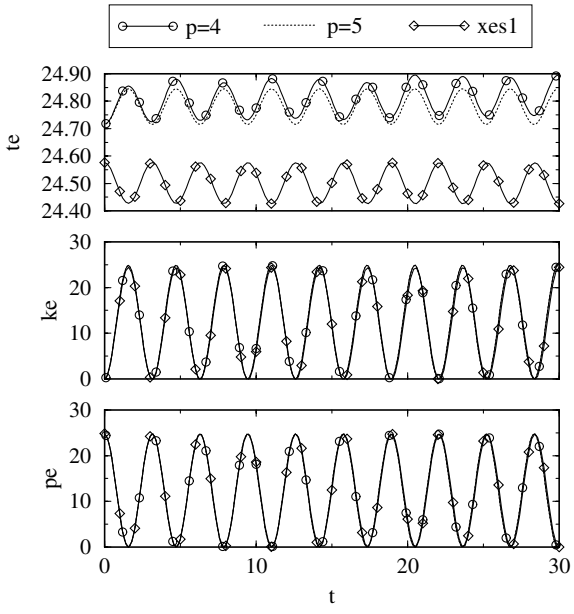


Fig. 11. A comparison of the total energy, te, kinetic energy, ke, and potential energy, pe, plotted versus time of electrostatic PIC simulations with the current method and XES1 for a plasma wave.

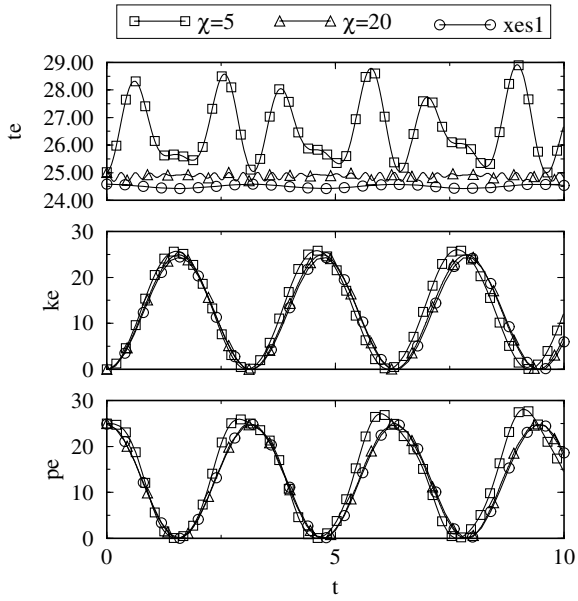


Fig. 12. A comparison of the total energy, te, kinetic energy, ke, and potential energy, pe, plotted versus time of electrostatic PIC simulations with the current method and XES1 for a plasma wave.

Fig. 12 shows that using the hyperbolic cleaning approach with the field solver conserves energy equally well as the spectral Poisson solver for $\chi = 20$. For $\chi = 5$ the total energy fluctuation (which should theoretically be zero) increases dramatically as compared to $\chi = 20$ and we observe a dispersive effect on the plasma wave when χ is too small. The improved agreement for increasing χ is expected as this changes the χ-method linearly towards the more realistic governing Maxwell's equations. The kinetic and potential energy results show equal dependencies on χ.

4.3.2. 2-stream instability

256 particles with $R = 0.5$ are released according to Eq. (26) with $A = 0.0001$, $k = 2$ and a unit velocity. Another 256 are released with $A = -0.0001$ and unit velocity in the opposite direction. Fig. 13 confirms that

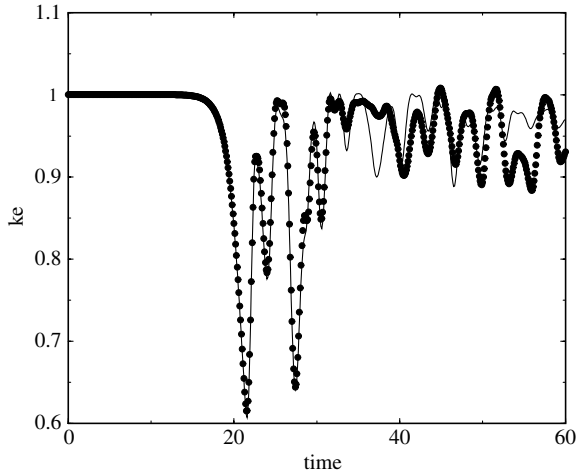


Fig. 13. A comparison of the kinetic energy, ke, plotted versus time of electrostatic PIC simulations with the current method and XES1 for a 2-stream instability.

the computed results (using $n = 4$ elements) and XES1 predict the appearance of the instability in the sudden drop of the kinetic energy at equal times. Excellent comparison up to $t \sim 35$ is found after which ($t > 35$) 2D effects lead to a different particle heating and different quantitative kinetic energy. Both methods show a total energy fluctuation of less than 1% indicating equally accurate energy conservation.

4.3.3. Landau damping

We simulate Landau damping with 1k and 10k particles with $R = 0.4$ released according to (26) with $A = 0.1$. The initial velocity is Maxwellian with a thermal velocity of $v_{\text{th}} = 0.4$.

Fig. 14(a) shows that both XES1 and the current method are unable to predict the Landau damping for more than 3 periods. The current method seems to deteriorate a little less. For 10k particles (Fig. 14) no substantial deterioration of the wave is observed for $t < 15$. The current methods' minima are, however, closer to zero indicating a more accurate approximation of the phenomenon.

4.4. Two-dimensional plasma tests

In the following, we shall also present a few fully electromagnetic two-dimensional tests to confirm that the general approach also works in such cases.

4.4.1. Weibel instability

This section presents results of PIC simulations of the Weibel instability presented in [21]. We compare the finite difference time domain (FDTD) method [1], and the high-order PIC method.

The Weibel instability simulations are performed on a unit square with periodic boundary conditions. We consider a quasi-neutral plasma with a thermal velocity ratio of 5 of the velocity in x, $u_{\text{the}} = 0.25$ and y, $v_{\text{the}} = 0.05$ direction. The plasma frequency is 15 times the length of the square, i.e., $\omega_{\text{pe}} = 15$ resulting in $\frac{q}{m} = -(15\pi)^2$ with the electron charge density set to $\rho = -1$.

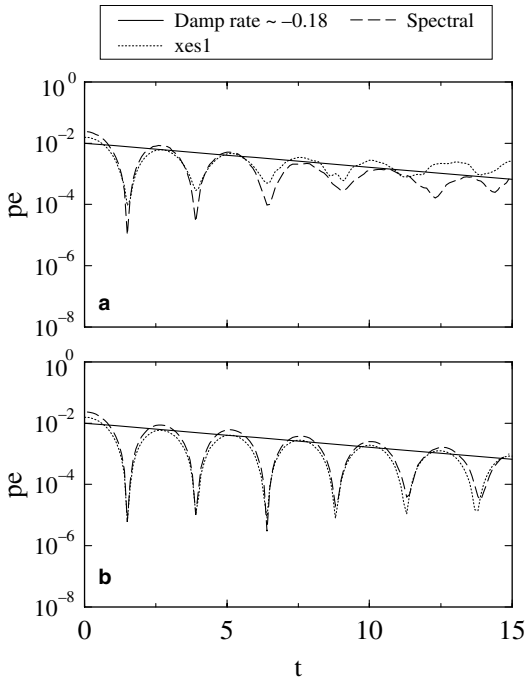


Fig. 14. A comparison of the potential energy, pe, plotted versus time of electrostatic PIC simulations with the current method and XES1 for Landau damping with (a) 1k particles and (b) 10k particles.

With these settings, magnetic waves develop with a dominant frequency in the y-direction. The wave number decreases in time as the thermal velocities approach the equilibrium state.

A study with the FDTD method indicates that a 256×256 grid with 36 particles per cell yields a reasonably converged solution. The results of this simulation are used in the remainder of this section for comparison.

The high-order simulations are performed on a grid with 200 triangles using a fifth order scheme. We track $N_p \times N_p$ particles in this domain for two time units. In Figs. 15 and 16, we compare various plasma energy

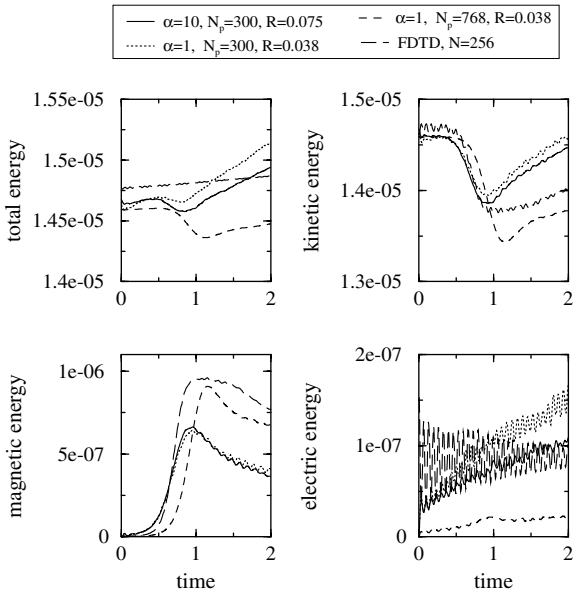


Fig. 15. Comparison of various plasma energy components for high-order simulations with LMM divergence cleaning at $\chi = 10$ with $\alpha = 1$ and 10 and $N_p = 300$ and 768 to FDTD PIC. For $\alpha = 10$ and 1, the particle cloud radius $R = 0.075$ and 0.038, respectively.

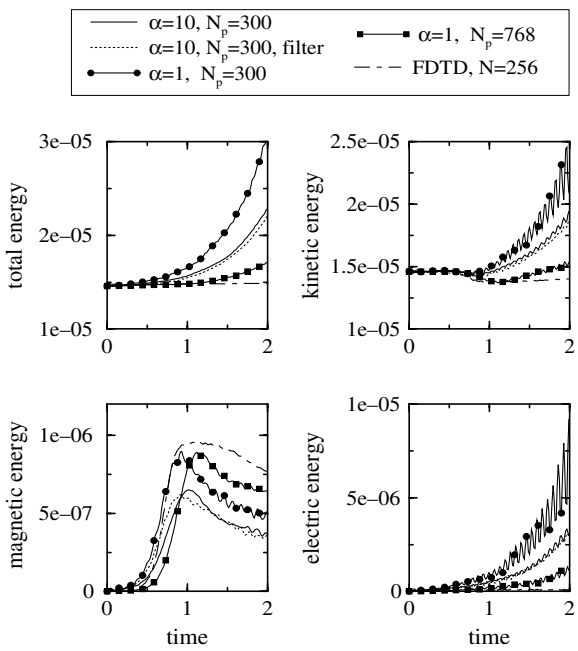


Fig. 16. Comparison of various plasma energy components for high-order simulations with Boris divergence cleaning with $\alpha = 1$ and 10, $N_p = 300$ and 768 and with filtering to FDTD PIC. For $\alpha = 10$ and 1, the particle cloud radius $R = 0.075$ and 0.038, respectively.

components of simulations, obtained using both Poisson and hyperbolic divergence ($\chi = 10$) cleaning, with the results of the FDTD simulations. The hyperbolic cleaning method conserves total energy better (less than 3% deviation) as compared to the global cleaning method (10–50% increase) for similar resolutions. The poor energy conservation in the latter approach translates into a poor comparison to FDTD method of the other energy components.

The FDTD based method requires significantly more grid points to obtain total energy conservation comparable to those obtained with the hyperbolic cleaning technique. Both show comparable kinetic energy trends, i.e., first a decrease followed by a slow increase. The hyperbolic cleaning method predicts a slightly smaller peak value in the magnetic energy as compared to the FDTD method, but the trend is comparable. The hyperbolic cleaning approach appears superior to PIC with global Poisson based cleaning as well as to FDTD in reducing noise as witnessed by the electrical energy trend.

Surprisingly, we observe that decreasing α in Eq. (18) for the hyperbolic cleaning approach (Fig. 15) has a minimal effect on the energy trends. Note that for $\alpha = 1$ we take R half the value as compared to $\alpha = 10$. The reason is that at $\alpha = 10$ most of the deposition function is located within the half radius near the origin. For $\alpha = 1$ the deposition in this region is approximated by a linear function. The simplified function and reduced influence region lead to a factor three speed up.

Fig. 16 shows that decreasing α with the Poisson based divergence cleaning leads to poorer energy conservation as one would expect (larger aliasing error) resulting in worsened energy trends. This is not the case for the χ cleaning. It is not clear why this difference is so obvious. Applying a weak filtering on the Poisson equation source term reduces grid heating, however it does not necessarily improve the results as witnessed in the H_z energy trend. Increasing the number of particles improves the result as expected through a reduction of the grid heating and an improved comparison to the FDTD result.

Figs. 17 and 18 compare the H_z and E_x energy spectra at $t = 2$. We see that most of the energy is stored in the region $0 < |\boldsymbol{k}| < 7$. At larger $|\boldsymbol{k}|$ the H_z energy spectrum of the FDTD simulation shows an increase caused by the inability of the finite difference method to capture high wave numbers effectively as well as enforcing energy conservation exactly, leading to a pileup of high-frequency energy. The high-order simulations show a drop in the spectrum caused by the diffusion of the upwind numerical flux, Eq. (11). The E_x energy spectrum

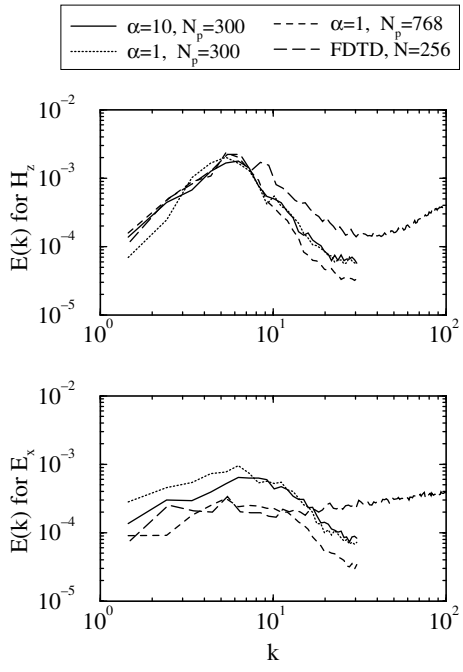


Fig. 17. Comparison of H_z and E_x energy spectra for for high-order simulations with LMM divergence cleaning at $\chi = 10$ with $\alpha = 1$ and 10 and $N_p = 300$ and 768 to FDTD PIC. For $\alpha = 10$ and 1, the particle cloud radius $R = 0.075$ and 0.038, respectively.

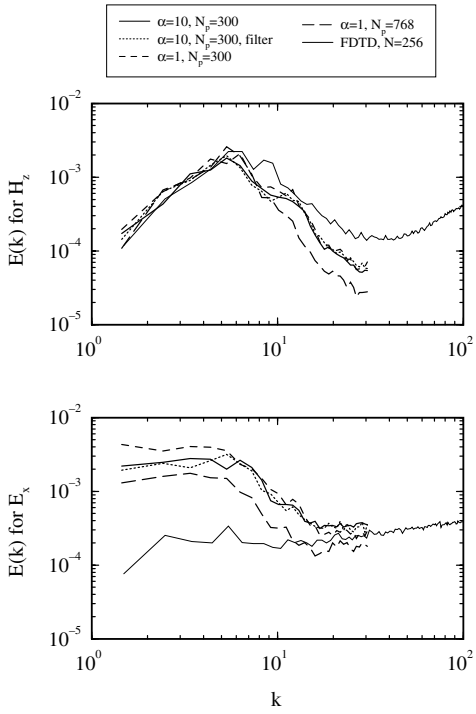


Fig. 18. Comparison of H_z and E_x energy spectra for high-order simulations with Boris divergence cleaning with $\alpha = 1$ and 10, $N_p = 300$ and 768 and with filtering to FDTD PIC. For $\alpha = 10$ and 1, the particle cloud radius $R = 0.075$ and 0.038, respectively.

shows no decrease with $|\boldsymbol{k}|$ for FDTD, but drop off for high-order simulations because of the slight high-frequency dissipation by the upwind flux. The H_z spectra are not affected much by α and a moderate filter. Increasing the number of particles increases the drop off at high wave numbers. Decreasing α and decreasing N_p introduces more energy in the low wavenumber part of the E_x spectrum. The filter has little effect on the E_x spectrum as well. The E_x spectrum of the high-order method with hyperbolic divergence cleaning compares better to the FDTD result than the results obtained with the projection based divergence cleaner.

The thermal u and v velocity at time $t = 2$, tabulated in Table 1, shows that the results obtained with both divergence cleaning techniques converge towards the FDTD result with increased resolution. From this table one concludes that the results for $\chi = 10$ compare best with the FDTD result. Increasing χ from 2 to 10 has quite an effect on the velocities indicating that the less physical $\chi = 2$ simulations should not be considered, consistent with the results in Section 4.3.1.

Table 1
Comparison of thermal velocities for high-order and FDTD PIC at $t = 2$

Scheme	α	R	N_p	u_{the}	v_{the}	$u_{the} - v_{the}$
$\chi = 2$	10	0.075	300	0.198	0.151	0.047
$\chi = 2$	1	0.038	300	0.202	0.148	0.054
$\chi = 2$	1	0.038	768	0.204	0.137	0.067
$\chi = 10$	10	0.075	300	0.208	0.145	0.062
$\chi = 10$	1	0.038	300	0.207	0.147	0.060
$\chi = 10$	1	0.038	768	0.205	0.138	0.067
Poisson	10	0.075	300	0.227	0.187	0.040
Poisson	1	0.038	300	0.237	0.194	0.043
Poisson	1	0.038	768	0.212	0.154	0.058
FDTD ($N = 256$)				0.206	0.140	0.066

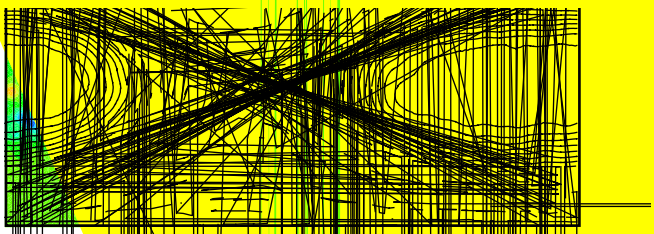


Fig. 19. Contours of out-of-plane current and magnetic field lines.

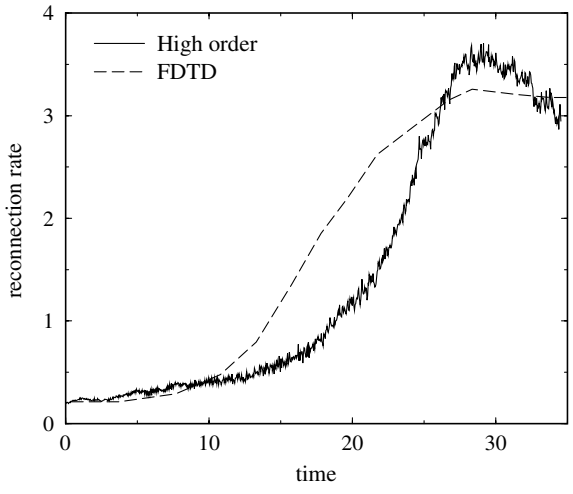


Fig. 20. The reconnected flux plotted versus time, compared with results obtained with CELESTE3D.

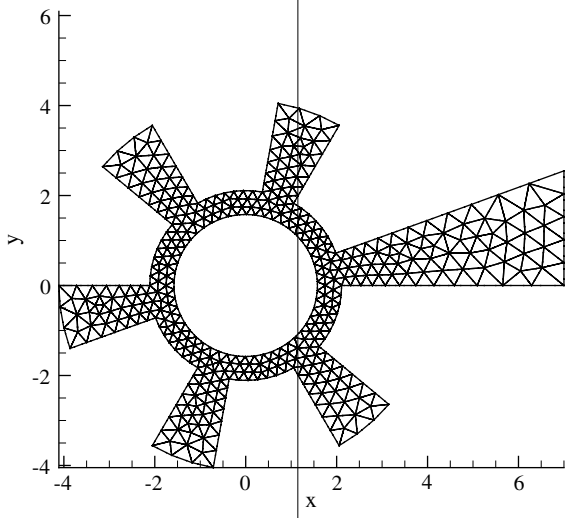
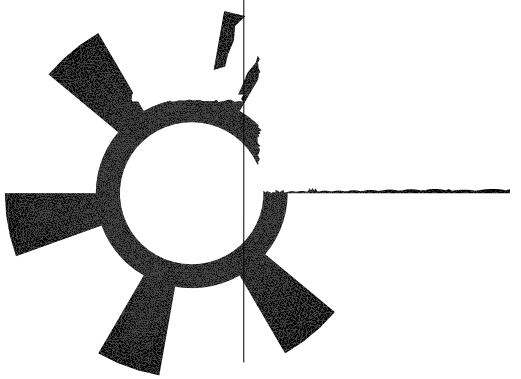


Fig. 21. Unstructured grid and dimensions used for the A6-magnetron flow simulation.



After initial effects have disappeared the A6 magnetron settles down in a 2π-mode, its normal operative mode, as shown in the particle snapshot of Fig. 22.

5. Concluding remarks and future directions

We present the first stage in the development of a new high-order particle-in-cell algorithm. The core of the algorithm is based on a high-order discontinuous Galerkin Maxwell field solver on unstructured grids. The main advantages of the method lie in the higher efficiency of high-order methods to deal with high-frequency physics and the geometric flexibility of the boundary-fitted unstructured grid. Furthermore, the DG formulation has inherent properties, e.g. natural dissipation control and dispersion properties to avoid numerical Cherenkov radiation, which suggests it is well suited as a core component of a PIC method.

The algorithm requires full order interpolation to determine the field at the particle position. A monomial polynomial basis ensures the interpolation is fast. The cell-location algorithm takes advantage of the inverse of the isoparametric mapping of physical coordinates to a master element.

The coupling of the particle grid to the Eulerian field grid uses simple smooth functions. This is shown to reduce noise and effectively enable control of finite grid instabilities. A constant influence area for each particle avoids compressible particles and reduces noise in the charge and current density at the possible cost of an increased number of weighing elements per particle. For problems with large scale separation in the geometry, one should consider an approach to enable particles of different sizes, e.g., an *h*-type particle adaptivity. We are currently exploring ways of achieving this without impacting charge conservation.

With the pre-computation of a levelset distance function, particles can interact elastically with complex geometric boundaries.

Divergence control is performed either through a classic projection scheme or a purely hyperbolic cleaning approach. The former requires the solution of a Poisson equation which is implemented with a discontinuous Galerkin method consistent with the field solver. The hyperbolic cleaning method is local, fully hyperbolic, and easily implemented in the framework of the Maxwell field solver. However, the equations become increasingly stiff when improving the physical representation making the method less effective. The computations show that values of χ exceeding 10 is needed to ensure a robust approach, making this technique more expensive than the projection method for comparable accuracy as long as an explicit time-stepping approach is used.

The computational results for a number of different plasma physics benchmarks and test cases confirm the ability to model these very basic phenomena while offering full geometric flexibility and the potential for *hp*-type adaptivity.

That said, however, many issues remain to be addressed in a satisfactory manner. In particular, the use of advanced implicit–explicit time-stepping methods to enable the use of the hyperbolic cleaning method efficiently at high values of χ seems a natural extension. Furthermore, guidelines for the trade offs between particle size and smoothness must be developed, e.g., for problems dominated by kinetic effects one should clearly be careful with using only a few large particles.

The development of particle clouds that are able to scale according to the size of the underlying grid as well as the fundamental physics make for a true *hp*-adaptive particle-in-cell method for the modeling of large scale plasma dynamic problems in complex geometries. We hope to be able to continue to report on such progress in the near future.

Acknowledgments

The authors thank Dr. T. Davis from the University of Florida at Gainesville for provision of sparse matrix-multiply functions and UMFPACK. We appreciate the freeware XES1 from the plasma and theory simulation group at Berkeley.

We also thank Dr. Fariba Fahroo and Dr. Arje Nachman, both from AFOSR, for initial and sustained encouragement and Dr. Keith Cartwright (AFRL/Kirtland AFB) and Dr. David Bruhwiler (TechX Corp) for help with identifying suitable test cases and continued support and encouragement.

This work was partially supported by NSF Career Award DMS0132967 and by AFOSR Award FA9550-05-1-0477.

Appendix A. Fast interpolation for particle mover

Interpolation that is consistent with the scheme in this paper use nodal points, $\boldsymbol{\xi}$, from Hesthaven [11] on the master element, I (rather than a general D). In this approach the smooth function $q(\boldsymbol{\xi})$, i.e., the electromagnetic field, is represented as

$$q(\boldsymbol{\xi}) = \sum_{j=1}^{N} q_j L_j(\boldsymbol{\xi}), \tag{27}$$

where $L_j(\boldsymbol{\xi})$ is the genuine two-dimensional multivariate Lagrange interpolation polynomial, $L_j(\boldsymbol{\xi}) \in P_n^2$, where

$$P_n^2 = \mathrm{span}\{\xi^i \eta^j; i, j \geqslant 0; i + j \leqslant n\}, \tag{28}$$

based on $N_n^2 = N$ nodal points, $\boldsymbol{\xi}_i$, given in the interior as well as on the boundary of I. For the interpolation to be complete, we must require

$$N = \frac{(n+1)(n+2)}{2}.$$

For the actual construction of the interpolation polynomials, let us introduce the complete polynomial basis, $p_i(\boldsymbol{\xi}) \in P_n^2$, and express the interpolation property as

$$\forall i : f(\boldsymbol{\xi}_i) = \sum_{j=0}^{N} \hat{f}_j p_j(\boldsymbol{\xi}_i) \Rightarrow V\hat{\boldsymbol{f}} = \vec{\boldsymbol{f}}, \tag{29}$$

where $\hat{\boldsymbol{f}} = [\hat{f}_0, \ldots, \hat{f}_N]^{\mathrm{T}}$ is the vector of expansion coefficients, $\boldsymbol{f} = [f(\boldsymbol{\xi}_0), \ldots, f(\boldsymbol{\xi}_N)]^{\mathrm{T}}$ is the grid vector and $V_{ij} = p_j(\boldsymbol{\xi}_i)$ is the multi-dimensional Vandermonde matrix. Clearly, for the interpolation to exist V must be nonsingular. Under the assumption of existence and uniqueness of the interpolation polynomial, we can express (29) as,

$$\forall i : f(\boldsymbol{\xi}) = \sum_{j=0}^{N} f(\boldsymbol{\xi}_i) L_j(\boldsymbol{\xi}_i). \tag{30}$$

Combining (29) and (30) implies that,

$$\boldsymbol{L} = (V^{\mathrm{T}})^{-1}\boldsymbol{p}. \tag{31}$$

The properties of V, e.g., its conditioning depends exclusively on the structure of the nodal set, $\boldsymbol{\xi}_j$, and on the way in which we choose to represent the basis, i.e., $p_i(\boldsymbol{\xi})$. While the former is chosen to ensure well-behaved Lagrange interpolation polynomials, we have significant freedom in the specification p_i. This freedom of choice in p_i makes it possible to choose between interpolation accuracy and speed. For creation of the interpolation and differentiation of the matrix, an orthonormal Jacobi polynomial basis results in a well conditioned Vandermonde matrix for an acceptable polynomial order range and provides good accuracy.

For particle interpolation where p in the RHS of (31) changes for every particle and every time step, the evaluation of p with an orthonormal Jacobi polynomial basis is too expensive. A faster alternative is the multivariate monomial basis, i.e., $p_i(\boldsymbol{\xi}_i) = \xi^i \eta^j$. For moderate polynomials order ($n < 4 - 6$) this basis conditions V acceptably and the computational efficiency is significantly improved as compared to the Jacobi basis. For higher polynomial order the condition number of grows exponentially with n for the monomial basis making it unsuitable for large n.

References

[1] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, NY, 1985.
[2] S.H. Gold, G.S. Nusinovich, Review of high-power microwave source research, Rev. Sci. Instrum. 68 (11) (1997) 3945–3973.
[3] J. Villasenor, O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69 (1992) 306–316.
[4] J.W. Eastwood, W. Arter, N.J. Brealey, R.W. Hockney, Body-fitted electromagnetic PIC software for use on parallel computers, Comput. Phys. Commun. 87 (1995) 155–178.
[5] T. Umeda, Y. Omura, T. Tominaga, H. Matsumoto, A new charge conservation method in electromagnetic particle-in-cell simulations, Comput. Phys. Commun. 156 (2003) 73–85.
[6] H.X. Vu, J.U. Brackbill, CELESTE1D: an implicit, fully kinetic model for low-frequency, electromagnetic plasma simulation, Comput. Phys. Commun. 69 (1992) 253.
[7] R.W. Hockney, S.P. Goel, J.W. Eastwood, Quiet high-resolution computer models of a plasma, J. Comput. Phys. 14 (1974) 148–158.
[8] A.D. Greenwood, K.L. Cartwright, J.W. Luginsland, E.A. Baca, On the elimination of numerical Cherenkov radiation in PIC simulations, J. Comput. Phys. 201 (2004) 665–684.
[9] J.S. Hesthaven, T. Warburton, High-order nodal discontinuous Galerkin methods for the Maxwell eigenvalue problem, R. Soc. London Ser. A 362 (2004) 493–524.
[10] J.S. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell's equations, J. Comput. Phys. 181 (2002) 186–221.

[11] J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, SIAM J. Numer. Anal. 35 (1998) 655–676.
[12] A.H. Mohammadian, V. Shankar, W.F. Hall, Computation of electromagnetic scattering and radiation using a time-domain finite-volume discretization procedure, Comput. Phys. Commun. 68 (1991) 175–196.
[13] M.H. Carpenter, C.A. Kennedy, A fourth-order 2N-storage Runge–Kutta scheme, NASA TM 109112, June, 1994.
[14] G.B. Jacobs, D.A. Kopriva, F. Mashayek, A particle-tracking algorithm for the multidomain staggered-grid spectral method, AIAA Paper 2001-0630, 2001.
[15] C.D. Munz, P. Omnes, R. Schneider, E. Sonnendrukker, U. Voss, Divergence correction techniques for Maxwell solvers based on a hyperbolic model, J. Comput. Phys. 161 (2000) 484–511.
[16] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.
[17] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, 1999.
[18] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfacesApplied Mathematical Sciences, vol. 153, Springer, Berlin, 2003.
[19] D. Gottlieb, J.S. Hesthaven, Spectral methods for hyperbolic problems, J. Comput. Appl. Math. 128 (1–2) (2001) 83–131.
[20] OOPIC Pro, Tech-X Corporation. Available from: <www.txcorp.com>.
[21] R.L. Morse, C.W. Nielson, Numerical simulation of the Weibel instability in one and two dimensions, Phys. Fluids 14 (4) (1971) 830–840.
[22] J. Birn, J.K. Drake, M.A. Shay, B.N. Rogers, R.E.M.H. Denton, M. Kuznetsova, Z. Ma, A. Bhattacharjee, A. Otto, P.L. Pritchett, Geospace Environmental Modeling (GEM) magnetic reconnection challenge, J. Geophys. Res. 106 (11) (2001) 3715–3719.
[23] G. Lapenta, private communication, 2005.
[24] A. Palevsky, G. Bekefi, Microwave emission from pulsed, relativistic e-beam diodes. II. The multiresonator magnetron, Phys. Fluids 22 (May) (1978) 986–996.
[25] R.W. Lemke, T.C. Genoni, T.A. Spencer, Three-dimensional particle-in-cell simulation study of a relativistic magnetron, Phys. Plasmas 6 (2) (1999) 603–613.