Journal of Computational Physics 229 (2010) 947–957

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Short Note

Controlling self-force errors at refinement boundaries for AMR-PIC

Phillip Colella [a,*], Peter C. Norgaard [b]

[a] Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA
[b] Mechanical and Aerospace Engineering Department, Princeton University Princeton, NJ 08540, USA

ARTICLE INFO

Article history:
Received 9 July 2008
Received in revised form 24 June 2009
Accepted 1 July 2009
Available online 14 July 2009

Keywords:
Particle-in-cell methods
Adaptive mesh refinement
Self-forces

ABSTRACT

We analyze the source of the self-force errors in the node-centered adaptive-mesh-refinement particle-in-cell (AMR-PIC) algorithm and propose a method for reducing those self-forces. Our approach is based on a method of charge deposition due to Mayo [A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM Journal of Numerical Analysis 21(2) (1984) 285–299] that can reduce the self-force error to any specified degree of accuracy.

1. Introduction

Particle-in-cell (PIC) methods [4] are often used to solve the Vlasov–Poisson equations for collisionless kinetic problems arising in plasma physics and astrophysics. In this approach, the distribution function is approximated using a particle representation, which is viewed mathematically as a sum of smoothed delta functions in phase space. The forces on particles are computed using a finite-difference or spectral approximation to the Poisson equation. Traditionally, the grids used for the Poisson calculation are uniform rectangular grids. In the absence of irregular geometries, FFT methods are used to solve Poisson's equation for the potential induced by charges interpolated from the particles to the grid points. The gradient is computed from that potential on the grid and interpolated to the particle locations. In the case where the boundary includes complex conducting surfaces, finite-difference approximations on rectangular grids are used, combined with cut-cell methods for the irregular Dirichlet boundaries and multigrid iteration for the Poisson solver.

For many problems, the replacement of a uniform rectangular grid by structured adaptive mesh refinement (AMR) is an attractive option [9,8]. Particles are often localized to a small subset of the physical domain and the use of an AMR grid that maintains a fixed number of particles per cell will improve both the efficiency and accuracy of the method. However, it has been noted [9] that such AMR-PIC algorithms can suffer from the presence of large contributions to the force induced by a particle on itself in the neighborhood of refinement boundaries. In contrast, it is straightforward to design uniform-grid versions of PIC that do not suffer from spurious self-forces. Most of the AMR-PIC algorithms currently in use either ignore the problem, or use Poisson solvers that impose a non-standard form for the matching conditions at coarse-fine boundaries that has the potential for a significant loss of accuracy.

In this paper, we analyze the source of the self-force errors in the AMR-PIC algorithm in [9] and propose a method for reducing those errors. Our approach is based on a method of charge deposition due to Mayo [5] that, in the present setting, can reduce the self-force error to any specified degree of accuracy. In practice, we only use this charge deposition algorithm for particles near refinement boundaries, where the errors are largest.

* Corresponding author. Tel.: +1 510 486 5412; fax: +1 510 495 2505.
E-mail addresses: pcolella@lbl.gov (P. Colella), norgaard@princeton.edu (P.C. Norgaard).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.004

2. Uniform-grid PIC

We want to compute an approximation to the force field induced by a charge distribution in free space.

$$\varphi(\mathbf{x}) = \int_{\mathbb{R}^D} G(|\mathbf{x}-\mathbf{y}|)\rho(\mathbf{y})d\mathbf{y}, \quad \vec{E} = -\nabla\varphi, \tag{1}$$

$$G(r) = -\frac{1}{2\pi}log(r) \text{ if } D = 2, \quad G(r) = -\frac{1}{4\pi r} \text{ if } D = 3.$$

A particle representation of this charge distribution is specified in terms of a collection of particle positions and charges, $\{(\mathbf{x}^k, q^k)\}$, $\rho(\mathbf{x}) \approx \sum_k q^k \delta(\mathbf{x}-\mathbf{x}^k)$. The particle-in-cell method for computing $\vec{E}^k \approx \vec{E}(\mathbf{x}^k)$ is given as follows.

(1) Interpolate the particle charges onto the grid.

$$\rho^h_{\mathbf{i}} = \sum_k \frac{q^k}{h^D}\Psi\left(\mathbf{i} - \frac{\mathbf{x}^k}{h}\right), \tag{2}$$

where $\mathbf{i} \in \mathbb{Z}^D$ and h is the mesh spacing.

(2) Compute grid-based approximation to the convolution integral in (1) and evaluate the field gradient on the grid using symmetric finite differences.

$$\varphi^h_{\mathbf{i}} = \sum_{\mathbf{j}} \rho^h_{\mathbf{j}} G^h_{\mathbf{i},\mathbf{j}}, G^h_{\mathbf{i},\mathbf{j}} \approx G(|\mathbf{i}h - \mathbf{j}h|), \tag{3}$$

$$E^h_{d,\mathbf{i}} = \sum_{\mathbf{s}} a_{\mathbf{s}}(\varphi^h_{\mathbf{i}+\mathbf{s}} - \varphi^h_{\mathbf{i}-\mathbf{s}}). \tag{4}$$

The approximation to the Green's function G^h is typically computed using a fast Poisson solver, with the free-space boundary conditions represented using FFTs and Hockney domain-doubling, or with the James–Lackner algorithm, that represents free-space boundary conditions by computing an appropriate Dirichlet boundary conditions using boundary convolutions (for a discussion of free-space boundary conditions, see [7]).

(3) Interpolate field gradient to particle locations

$$\vec{E}^k = \sum_{\mathbf{i}} \Psi(\frac{\mathbf{x}^k}{h} - \mathbf{i})\vec{E}_{\mathbf{i}} \tag{5}$$

Here, the function Ψ used to interpolate between particles and grids is assumed to be an even function of its arguments. A simple choice is given by

$$\Psi(\mathbf{z}) = \prod_{d=1}^{D} max(1 - |z_d|, 0) \tag{6}$$

The self-force induced by a single particle $(\mathbf{x}^0, q^0)$ can be easily computed from this algorithm.

$$E^0_d = \frac{q^0}{h^D}\sum_{\mathbf{s}} a_{\mathbf{s}} \sum_{\mathbf{i},\mathbf{j}} \Psi\left(\mathbf{i} - \frac{\mathbf{x}^0}{h}\right)\Psi\left(\frac{\mathbf{x}^0}{h} - \mathbf{j}\right)\left(G^h_{\mathbf{i}+\mathbf{s},\mathbf{j}} - G^h_{\mathbf{i}-\mathbf{s},\mathbf{j}}\right) \tag{7}$$

$$= \frac{q^0}{h^D}\sum_{\mathbf{s}} a_{\mathbf{s}} \sum_{\mathbf{i},\mathbf{j}} \left(\Psi\left(\mathbf{i} - \mathbf{s} - \frac{\mathbf{x}^0}{h}\right)\Psi\left(\frac{\mathbf{x}^0}{h} - \mathbf{j}\right)G^h_{\mathbf{i},\mathbf{j}}\right) - \left(\sum_{\mathbf{i},\mathbf{j}} \left(\Psi\left(\mathbf{i} - \frac{\mathbf{x}^0}{h}\right)\Psi\left(\frac{\mathbf{x}^0}{h} - (\mathbf{j}-\mathbf{s})\right)G^h_{\mathbf{i}-\mathbf{s},\mathbf{j}-\mathbf{s}}\right)\right) \tag{8}$$

Clearly, this sum vanishes if $G^h_{\mathbf{i},\mathbf{j}} = G^h_{\mathbf{i}-\mathbf{s},\mathbf{j}-\mathbf{s}}$ i.e. if G^h is translation-invariant. This is an exact relation for discretizations of G obtained from the Hockney construction and is true to a high degree of accuracy for finite-difference discretizations using James–Lackner.

Finally, one can define uniform-grid PIC methods for other boundary conditions by replacing the free-space approximate Green's function in (1) by one that satisfies the requisite boundary conditions, with the potential φ^h computed using fast solvers. In that case, there is a nontrivial self-force induced by the effect of the boundaries on the field. Nonetheless, there is a well-defined notion of the error in the force induced by a single particle on itself. On uniform-grids, the error in the self-force of a single particle for such boundary conditions is $O(h^p)$, where p is the order of accuracy of the discretization of the Laplacian.

3. An AMR-PIC algorithm

We define our locally-refined grid in terms of a nested hierarchy of unions of rectangles, following the approach in [6]. We define a hierarchy of mesh spacings $h_l, l = 1 \ldots l_{max}$, with $h_l = r_l h_{l+1}, r_l > 1$ an integer. Then Ω^l is a union of discrete node-centered rectangles in $\mathbb{R}^D$.

$$\Omega^l \subset \{\mathbf{i}h_l : \mathbf{i} \in \mathbb{Z}^D\} \tag{9}$$

$$\Omega^l \supset \mathcal{P}(\Omega^{l+1}), \quad \mathcal{P}(\mathbf{i}h_{l+1}) = \left\lfloor \frac{\mathbf{i}}{r_l} \right\rfloor h_l. \tag{10}$$

We also define the points in our hierarchy where the finite-difference approximation to the Laplacian that we will use is defined.

$$\Omega^l_{valid} = \{\mathbf{p} \in \Omega^l : \mathbf{p} \pm \mathbf{e}^d h_l \in \Omega^l, d = 1 \ldots D\} - \Omega^{l+1}_{valid}, \tag{11}$$

$$\Omega_{valid} = \bigcup_l \Omega^l_{valid}, \tag{12}$$

where $\mathbf{e}^d$ denotes the unit vector in the d direction. Finally, we assume that the various levels in the hierarchy satisfy the following proper nesting conditions, in order to simplify the bookkeeping in our discretization methods and to make certain that the values required for the finite-difference approximations are available in Ω_{valid}.

- Grids on level l cover the geometric region in space identical to that covered by a subset of Ω^{l-1}.

$$\mathcal{I}(\mathcal{P}(\Omega^l)) = \Omega^l \tag{13}$$

$$\mathcal{I}(\Sigma) \equiv \{(\mathbf{i}r_l + \mathbf{s})h_{l+1} : \mathbf{i} \in \Sigma, 0 \leqslant s_d \leqslant r_l - 1\}, \quad \Sigma \subset \Omega^l \tag{14}$$

- All points on the boundary of Ω^l are separated from any point on the boundary of Ω^{l+1} by at least a distance $2h_l$ in every coordinate direction.

We define the generalization of the $2D + 1$-point discretization of the Laplacian on this locally-refined grid hierarchy to be

$$\varphi^a : \Omega_{valid} \to \mathbb{R}, \tag{15}$$

$$\Delta^a(\varphi^a)_{\mathbf{p}} = \frac{1}{(h_l)^2}\left(2D\varphi^a_{\mathbf{p}} - \sum_{\pm}\sum_{d=1}^{D} \tilde{\varphi}^a_{\mathbf{p}\pm\mathbf{e}^d h_l}\right), \quad \mathbf{p} \in \Omega^l_{valid}, \tag{16}$$

where $\tilde{\varphi}^a$ is an extension of φ^a to Ω^l obtained by interpolating values at the points $\Omega^l - \Omega_{valid}$; for details, see [6].

To generalize the uniform-grid PIC algorithm described above to AMR grids, we want to think of the AMR solution algorithm in terms of convolution with a discrete Green's function and find a charge deposition/field interpolation method for which the self-force expression corresponding to (8) vanishes. Since the AMR grids consist of unions of rectangular patches, a natural choice would be to apply the uniform-grid formula corresponding to the local rectangular grid spacing at the location of the particle. This fails for two reasons. The first is that not all of the nodes surrounding a rectangular grid cell are in Ω_{valid} (Fig. 1). To accomodate that problem, we can deposit the charge on the grid corresponding to the finest level such that all the corners of the cell containing the particle are in Ω_{valid} and then use only the potential at that level of refinement for interpolating the electric field. In that case, we obtain the expression for the self-force given in (8). However, this is not sufficient, because the discrete Green's function corresponding to the AMR discretization of the Laplacian given above is not translation-invariant, so that the cancellation in (8) does not occur. For grid points that are sufficiently far away from refinement boundaries, the discrete Green's function fails to be translation-invariant by terms that are $O(h^2)$. Thus for particles located in such regions, we obtain a level of self-force that converges to zero with the mesh spacing and in practice yields an acceptable level of error. However, for points near to refinement boundaries, the discrete Green's function fails to be translation-invariant by terms that are $O(1)$ relative to the mesh spacing, leading to the large self-force contributions described in [9].

Our remedy for this problem is to modify the charge deposition algorithm in the neighborhood of refinement boundaries, using the approach in [5]. In this approach, the charge at the grid points surrounding the cell are all convolved with a discrete kernel corresponding to the AMR Poisson operator described above applied to the analytic Green's function evaluated at grid points. The kernel is truncated to be zero within a few mesh spacings of the particle, since the Green's function is harmonic away from the charge. The resulting discrete Green's function approximates the analytic Green's function and as such is more nearly translation-invariant, leading to the cancellations required in (8) to reduce the self-force to acceptable levels.

The AMR-PIC algorithm based on this approach can be specified as follows:

(1) Interpolate the particle charges onto the grid. Find the smallest h_l such that $\mathbf{x}^k \in [\mathbf{p}_0, \mathbf{p}_0 + \mathbf{u}h_l]$, $\mathbf{p}_0 \in \Omega_{valid}$($\mathbf{u}$ is the vector whose entries are all 1's) and

$$\mathcal{S}^k = \{\mathbf{p}_0 + \mathbf{v}h_l : v_d = 0, 1\} \subset \Omega_{valid} \tag{17}$$

$$\{\mathbf{s} \pm \mathbf{e}^d h_l : \mathbf{s} = \mathcal{S}^k, d = 1 \ldots D\} \subset \Omega_{valid} \tag{18}$$

These conditions are guaranteed to hold for some level l because of our proper nesting conditions. We define

$$R^k_{\mathbf{p}} = \frac{q^k}{h_l^D}\Psi\left(\frac{\mathbf{p} - \mathbf{x}^k}{h_l}\right), \quad \mathbf{p} \in \mathcal{S}^k \tag{19}$$

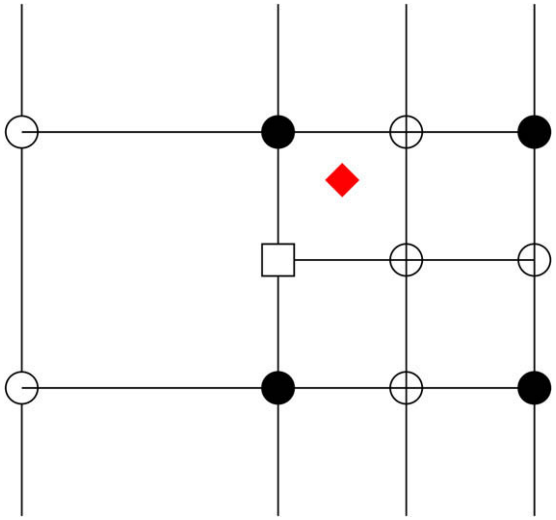

Fig. 1. AMR-PIC near a grid refinement boundary. The particle location is indicated by the red diamond, and the points in Ω_{valid} are indicated by circles (both filled and open). The grid location indicated by the square in this figure is computed by interpolation from the points in Ω_{valid}, rather than by solving a difference approximation to Poisson's equation at that point. In particular, there is no right-hand side defined at that point. Consequently, we distribute the charge from the particle to the four grid points in Ω_{valid} on the coarse grid adjacent to the particle, indicated by the filled circles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and $R^k_{\mathbf{p}} = 0$ otherwise, where Ψ is given by (6). We then define

$$\rho^{a,k}_{\mathbf{p}} = \sum_{\mathbf{q}\in\Omega_{valid}} w^k_{\mathbf{p},\mathbf{q}} R^k_{\mathbf{q}}, \rho^a_{\mathbf{p}} = \sum_k \rho^{a,k}_{\mathbf{p}}, \quad \mathbf{p} \in \Omega_{valid} \tag{20}$$

If the particle position $\mathbf{x}^k$ is sufficiently far away from any refinement boundary, we set $w^k_{\mathbf{p},\mathbf{q}} = \delta_{\mathbf{pq}}$, i.e. $\rho^{a,k}_{\mathbf{p}} = R^k_{\mathbf{p}}$. (we will give a precise description of the condition below). Otherwise,

$$w^k_{\mathbf{p},\mathbf{q}} = \left(\Delta^a G^{\mathbf{p}}\right)_{\mathbf{q}} \text{ if } |(\Delta^a G^{\mathbf{p}})_{\mathbf{q}}| \geqslant \eta \max_{\mathbf{q}'} |(\Delta^a G^{\mathbf{p}})_{\mathbf{q}'}| \tag{21}$$

and $w^k_{\mathbf{p},\mathbf{q}} = 0$ otherwise. Here, $G^{\mathbf{p}}_{\mathbf{q}} = G(\max(|\mathbf{p}-\mathbf{q}|, h_l))$ and $\eta < 1$ is an adjustable parameter that controls the level of self-force.

(2) Compute the grid-based approximation to the convolution integral in (1) and evaluate the field gradient on the grid using symmetric finite differences. We solve

$$\Delta^a \varphi^a = \rho^a \text{ on } \Omega_{valid} \tag{22}$$

using some form of infinite-domain boundary conditions at the coarsest level. We then compute the gradient field using second-order finite differences.

$$E^a_{d,\mathbf{p}} = -\frac{1}{2h_l}\left(\varphi^a_{\mathbf{p}+\mathbf{e}^d h_l} - \varphi^a_{\mathbf{p}-\mathbf{e}^d h_l}\right) \tag{23}$$

(3) Interpolate field gradient to particle locations.

$$\vec{E}^k = \sum_{\mathbf{p}\in\mathcal{S}^k} \vec{E}^a_{\mathbf{p}} \Psi\left(\frac{\mathbf{x}^k - \mathbf{p}}{h_l}\right) \tag{24}$$

By (18), we need the values of $\vec{E}^a_{\mathbf{p}}$ only at points where the stencil is contained in Ω_{valid}.

Finally, we must specify the rule by which we use this more elaborate charge deposition algorithm. If $\mathbf{x}^k$ is contained in the region in space defined by Ω^l that is also not covered by the region defined by Ω^{l+1}, then we use (21) provided that

$$dist(\mathbf{x}^k, \partial\Omega^l) < C_{low} h_l \quad \text{or} \quad dist(\mathbf{x}^k, \partial\Omega^{l+1}) < C_{high} h_l, \tag{25}$$

where the distance function is defined using the max norm.

In the special case of a single particle, the self-force on the particle is given by the same formula as in the uniform-grid case.

$$E_d^0 = \frac{q^0}{h_l^D}\frac{1}{2h_l}\left(\sum_{\mathbf{p},\mathbf{q}} \Psi\left(\frac{\mathbf{p}-\mathbf{e}^d h_l - \mathbf{x}^0}{h}\right)\Psi\left(\frac{\mathbf{q}-\mathbf{x}^0}{h}\right)G_{\mathbf{q}}^{a,\mathbf{p}} - \sum_{\mathbf{p},\mathbf{q}} \Psi\left(\frac{\mathbf{q}-\mathbf{e}^d h_l - \mathbf{x}^0}{h_l}\right)\Psi\left(\frac{\mathbf{p}-\mathbf{x}^0}{h_l}\right)G_{\mathbf{q}-\mathbf{e}^d h_l}^{a,\mathbf{p}-\mathbf{e}^d h_l}\right), \tag{26}$$

where the discrete Green's function G^a is given as the solution to

$$\left(\Delta^a G^{a,\mathbf{p}}\right)_{\mathbf{q}} = w_{\mathbf{p},\mathbf{q}}^0. \tag{27}$$

The points at which the summands in (26) are nonzero are all contained in $\Omega_{valid} \bigcap \Omega^l$, so that the derivation is the same as in the uniform-grid case. If $\eta = 0$ in (21), then $G^{a,\mathbf{p}} = G^{\mathbf{p}}$ the grid so that $G_{\mathbf{q}-\mathbf{e}^d h_l}^{a,\mathbf{p}-\mathbf{e}^d h_l} = G_{\mathbf{q}}^{a,\mathbf{p}}$ and the self-force goes to zero. Since the Green's function is harmonic away from the charge, Δ^a applied to the Green's function evaluated at grid points is exactly the truncation error of the discretized operator, which decays as a function of the distance of the charge. Thus choosing a small, but nonzero η in (21) leads to an potential that differs by a small amount from the Green's function evaluated at grid points, which leads to a reduced self-force. On the other hand, we find that, for choices of η that provide a significant reduction of the self-force, there are actually only a small number of nonzero weights in the neighborhood of $\mathbf{p}$, making the cost of computing the convolution in (20) small.

The present algorithm differs from the AMR-PIC algorithm discussed in [9], in that the latter sets $\rho_{\mathbf{p}}^{a,k} = R_{\mathbf{p}}^k$. The corresponding discrete Green's function has large deviations from translation invariance, leading to large self-forces. Our approach for reducing this error leads to an algorithm for charge deposition is more expensive than that given in [9]. However, the self-forces fall off rapidly with the distance of the particle from refinement boundaries, so that the simpler charge deposition algorithm used there can be applied away from those boundaries, with the more complicated one described here used only for a small subset of the particles. In addition, the convolution coefficients $w_{\mathbf{p},\mathbf{q}}^k$ are the same for all the particles whose bins $\mathcal{S}^k$ are the same. Thus the calculation can be organized to sum the initial deposition (19) for all the particles in the same bin and then apply the convolution (20) to the sum. Since there are typically at least several particles per bin, this makes the increase in the number of operations over the standard charge deposition algorithm even smaller.

4. Results

We compare the accuracy of the algorithm presented here with that of a baseline AMR-PIC algorithm on a series of single-particle tests [9]. In all of these tests, we compute the solution using algorithm described in [9] as well as with the algorithm described here. The AMR Poisson discretization is given by that in [6] for two AMR levels, with a factor of 2 refinement between the coarse and fine level. In all but the last example, the coarse grid covers the entire domain, which is the unit square $[0,1]^2$, with the fine gird covering the subdomain $[.25,.75]^2$. Thus the number of coarse grid points in each direction is equal to the number of fine grid points in each direction and we will denote both by $N+1$. For the case of free-space boundary conditions, we compute a Dirichlet boundary condition on the domain boundary using the analytic Green's function for the particle. The value for η in all of the calculations performed here is .3, with $C_{low} = 3, C_{high} = 1$.

In the first series of tests we examine the self-force on a single particle with unit charge in free space generated by the old and new algorithms, computed as a function of the location of the particle. In Figs. 2 and 3, we show a scatter plot of the force at the particle locations corresponding to grid points, as a function of the distance to the refinement boundary. In this case, the self-force is entirely due to the charge deposition, since there is no interpolation error. Away from the refinement boundaries, we are using the same charge deposition algorithm (19), so that the answers agree in that region. In addition, at any fixed distance away from the boundary, the self-force error is approaching zero quadratically in the mesh spacing, while in the neighborhood of the boundary both algorithms are showing a self-force error that is $O(1)$ relative to the mesh spacing. However, the present algorithm reduces that $O(1)$ error by more than two orders of magnitude. We emphasize that we can reduce the self-force as much as we wish by reducing further the value of the parameter η , at the expense of increasing the cost of depositing the charge.

Next, we look at the error in the far field of the expression

$$\int_{\partial B} \vec{E}\cdot\hat{\mathbf{n}}dA \tag{28}$$

which, according to Gauss' law, should be equal to the total charge contained in the domain B. The possibility for large errors in this expression are specific to the node-centered method used here to discretize Poisson's equation. In contrast, the method in [8] does not generate large errors for charges well-separated from ∂B since the discretization satisfies a discrete form of Gauss' law. In Figs. 4 and 5 we show a scatter plot of the errors in (28) for particles located at grid points, where the domain B is the box $[.125,.875]^2$ and the integral is replaced by a second-order accurate quadrature using the values for $\vec{E}$

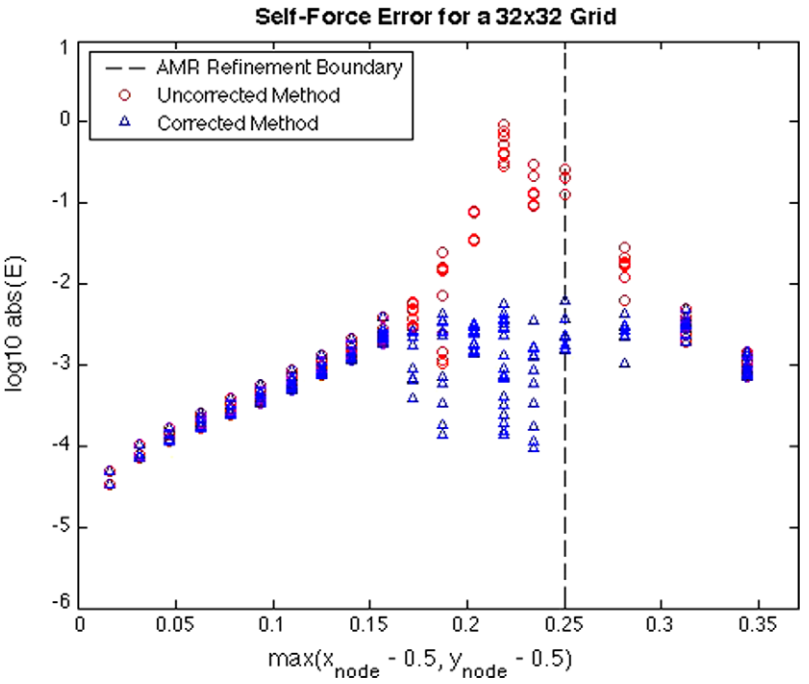

Fig. 2. Plot of self-force at grid points for a single particle placed at a point in Ω_{valid}, as a function of the distance of the point to the refinement boundary, $N = 32$. The values for the baseline method are plotted as red triangles, and the values for the new method plotted as blue circles. Here and in what follows, the refinement boundary is indicated by a dotted line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

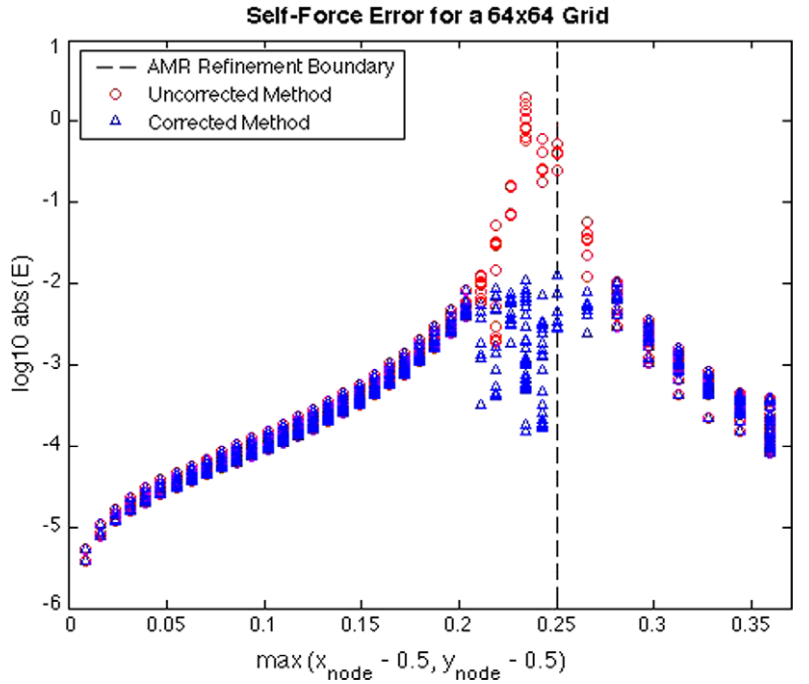

Fig. 3. Same as Fig. 2, but with $N = 64$.

computed at grid points by the algorithm. As before, we see a reduction by two orders of magnitude in the error for particles near the refinement boundary and quadratic convergence in the mesh spacing for particles a fixed distance away from the boundary.

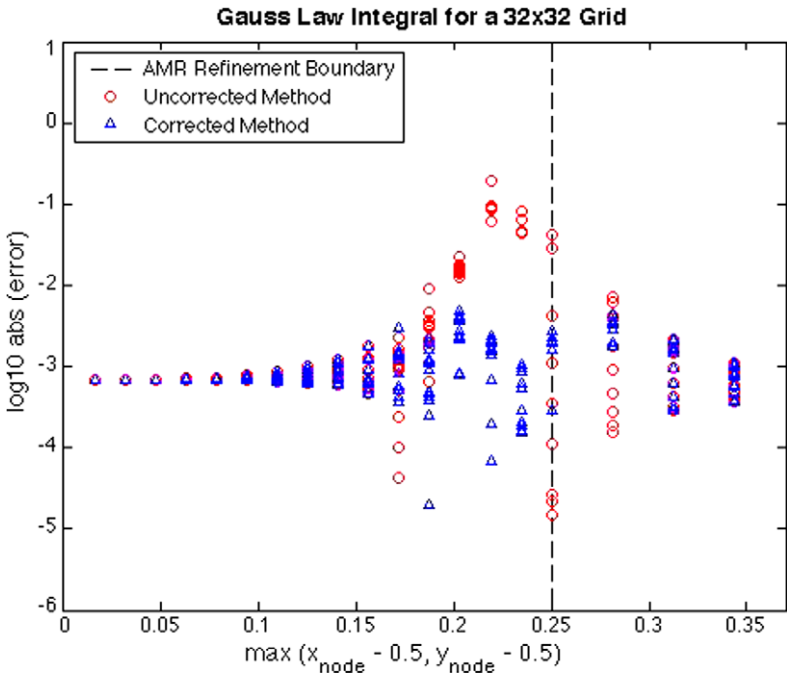

Fig. 4. Plot of error in (28) for a single particle placed at a point in Ω_{valid}, as a function of the distance of the point to the refinement boundary, $N = 32$. The values for the baseline method are plotted as red triangles, and the values for the present method plotted as blue circles. The refinement boundary is indicated by a dotted line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

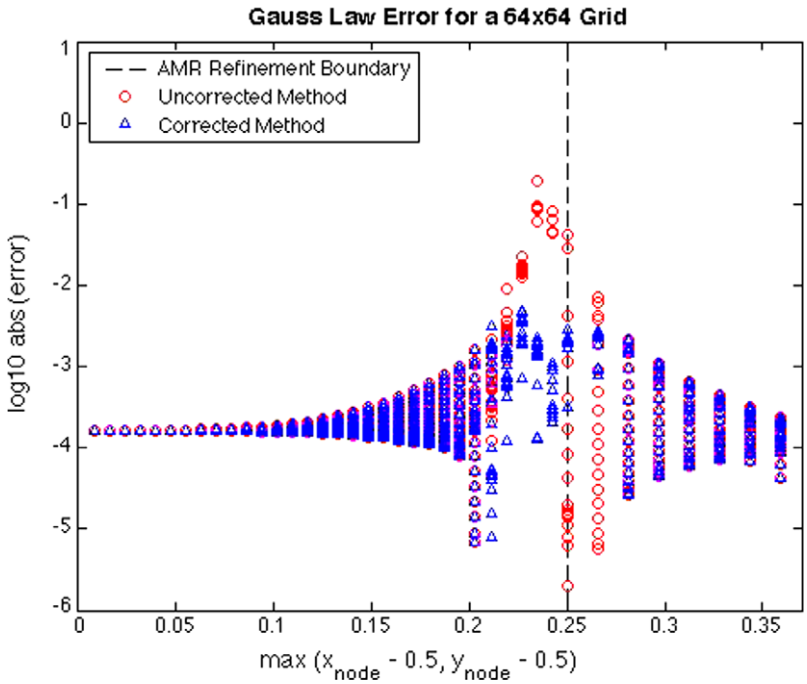

Fig. 5. Same as Fig. 4, but with $N = 64$.

In Figs. 6 and 7 we compare the self-force errors for particles not located on grid points. Again, we see a reduction in the error by two orders of magnitude, thus demonstrating that the reduction in the error is preserved by the interpolation step. To obtain this result, it is essential to impose the conditions (17) and (18). Otherwise, the error in interpolating the potential to non-valid points introduces large self-force errors.

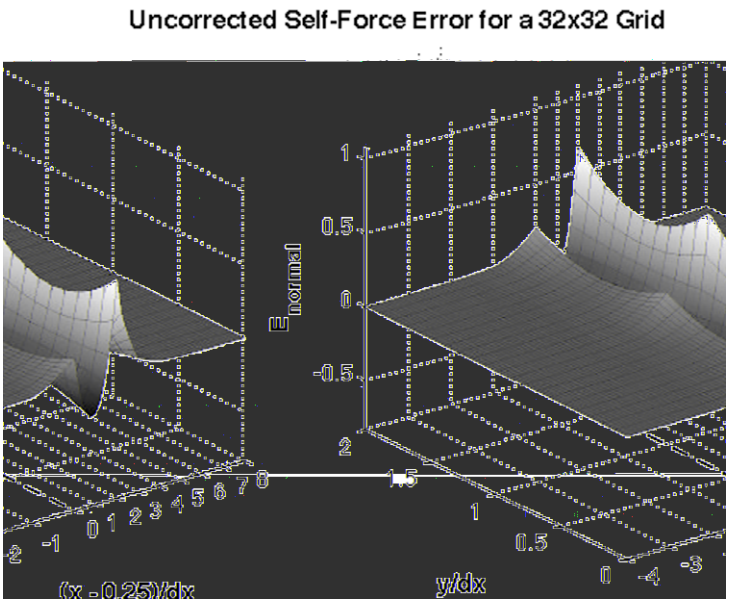

Fig. 6. Plot of self-force on single particles, as a function of particle location, near the refinement boundary. Baseline method, $N = 32$.

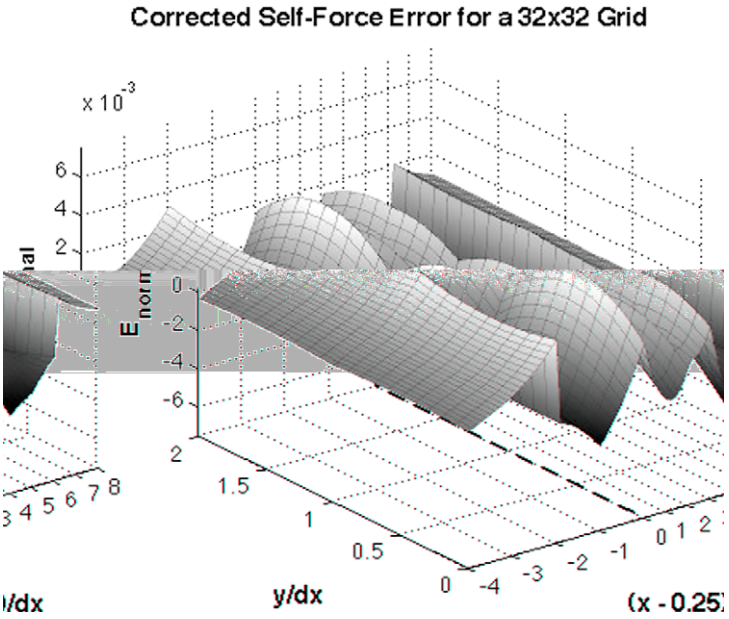

Fig. 7. Plot of self-force on single particles, as a function of particle location, near the refinement boundary. New method, $N = 32$. Note the change in the scale of the dependent variable.

In Figs. 8 and 9 we show the calculation of the particle trajectories for a particle in conducting box starting at rest from the point $(.3, 0)$. The conducting walls induce a force on the particle, causing it to move towards the wall. When the particle reaches the wall, we apply an elastic reflection boundary condition. The particle dynamics are computed using a standard leapfrog method for integrating the ODEs, with the electric field computed using either the baseline method or the present method. Our results for the baseline method reproduce those in [9], with the large self-force error causing the particle to stay on one side of the refinement boundary. The reduction in the self-force error in the present method causes the particle to accelerate correctly as it crosses the refinement boundary. In this case, the conducting boundary induces a self-force on the particle, causing it to move. However, the error in the self-force in the baseline method due to the refinement boundary is so great as to bring the particle to a halt at that boundary. By using the present algorithm, we can reduce the self-force so that the dynamics of the single particle are correct.

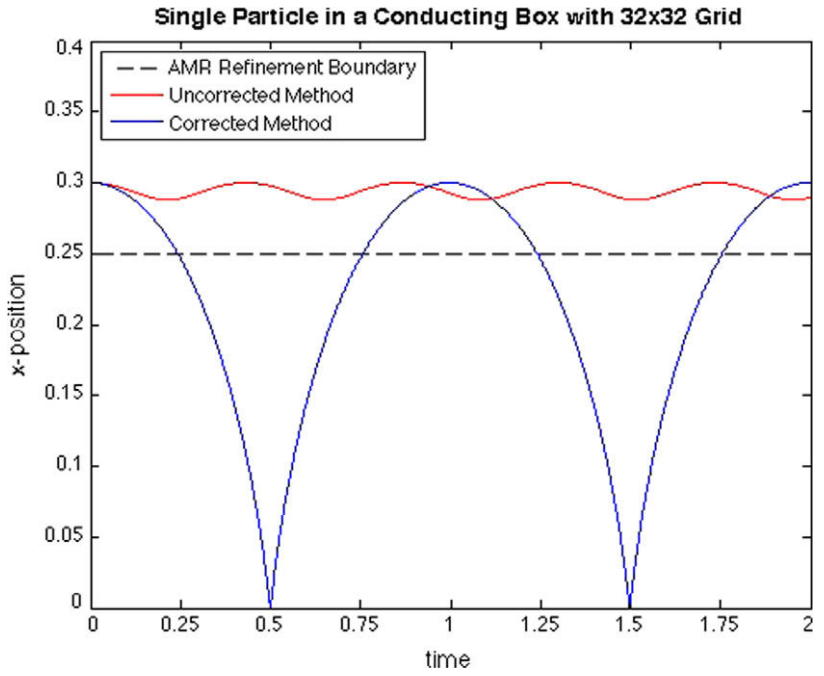

Fig. 8. Plot of the x-component of a single particle trajectory as a function of time, for $N = 32$. Red is the baseline method, blue is the new algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

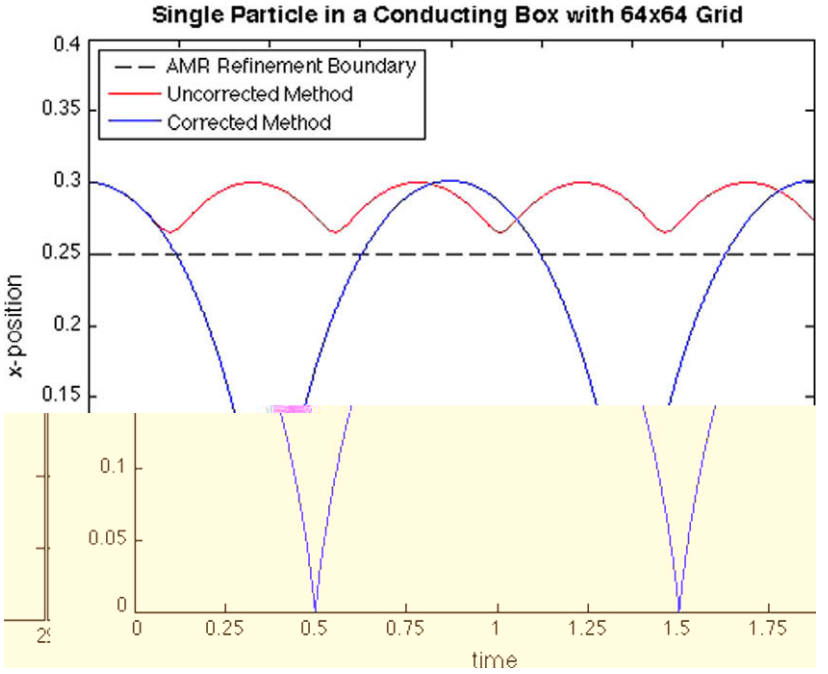

Fig. 9. Same as Fig. 8, but with $N = 64$.

Finally, in Fig. 10, we show the results of a calculation involving two particles. In this case, the base level grid is 64×64, with the region $\{x < .5\}$ is refined by a factor of two. The particles are placed initially at the points $(.3, .5)$ and $(.7, .5)$. The exact solution for this problem consists of the particles oscillating symmetrically about the point $(.5, .5)$ along the line $\{y = .5\}$. The results show that the present method more nearly preserves the symmetry of the exact solution than the baseline algorithm.

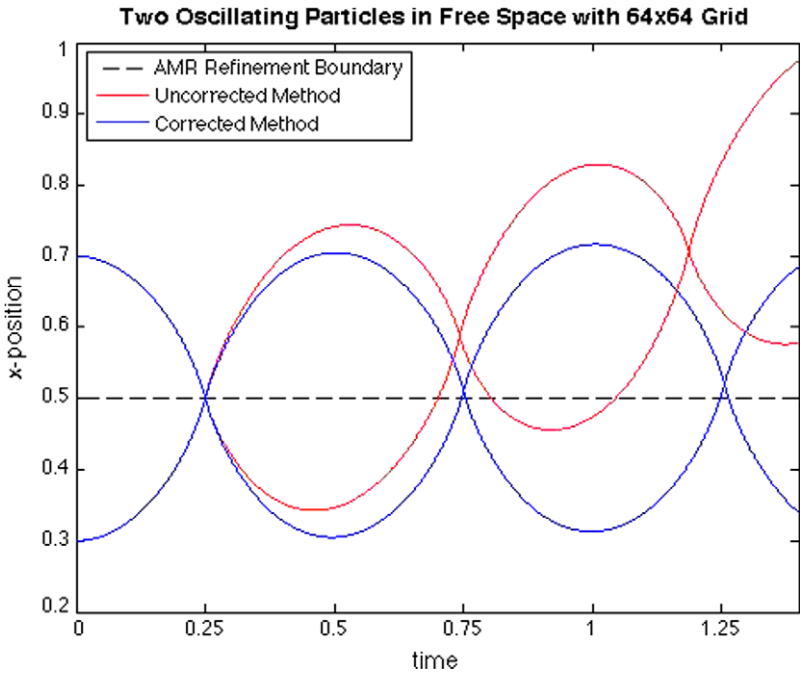

Fig. 10. Plot of the particle trajectories of the two-particle problem as a function of time, for $N = 64$. Red is the baseline method, blue is the new algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Conclusions

The method presented here addresses the issues raised in [9] regarding the use of AMR-PIC for Vlasov–Poisson. Perhaps the most important result is the control of the error in the far field represented here by the error in Gauss' law. The errors in the far field are global, with contributions from all of the particles near refinement boundaries, it is much more important that we have a mechanism for controlling such errors. The present method provides such a mechanism.

The results here provide a starting point for the design of a general method for plasma physics and astrophysics simulations. Some of the issues still to be addressed include the determination of robust choices for the numerical parameters, such as η and the distance to the refinement boundaries at which one changes charge deposition methods. For example, it may well be the case that these parameters will need to change as a function of the local mesh spacing. Another issue is the use of these ideas the algorithm in [8], which is based on a finite-volume discretization of the Laplacian. In that case, the stencil for interpolating the forces to particles cannot be forced to automatically have the cancellations required near refinement boundaries as was done here, and another approach, such as one based on the ideas in [2] will need to be employed. It is also possible to use AMR Mehrstellen discretizations [1,3] of the Laplacian. When these operators are applied to the Green's function as in (21), the magnitude of the weights decays more rapidly as a function of the distance from the charge than is the case for the standard $(2D + 1)$ point operator [2]. This leads to smaller stencils for the convolution operator to obtain a given level of accuracy.

Acknowledgments

We would like to acknowledge many helpful discussions with Alex Friedman at LLNL and Jean-Luc Vay at LBNL and in particular for suggesting several of the test problems used here for evaluating the self-force errors. Work at LBNL was supported by the US Department of Energy Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231. PCN was supported by the US Department of Energy Computational Sciences Graduate Fellowship Program under grant number DE-FG02-97ER25308.

References

[1] A.S. Almgren, T. Buttke, P. Colella, A fast adaptive vortex method in three dimensions, Journal of Computational Physics 113 (1994) 177–200.
[2] C.R. Anderson, A method of local corrections for computing the velocity field due to a distribution of vortex blobs, Journal of Computational Physics 62 (1986) 111–123.
[3] M. Barad, P. Colella, A fourth-order accurate local refinement method for Poisson's equation, Journal of Computational Physics 209 (2005) 1–18.
[4] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, McGraw-Hill, 1981.
[5] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM Journal of Numerical Analysis 21 (2) (1984) 285–299.

[6] P.W. McCorquodale, P. Colella, D.P. Grote, J.-L. Vay, A node-centered local refinement algorithm for Poisson's equation in complex geometries, Journal of Computational Physics 201 (2004) 34–60.
[7] P.W. McCorquodale, P. Colella, G. Balls, S.B. Baden, A local corrections algorithm for solving Poisson's equation in three dimensions, Communications in Applied Mathematics and Computational Science 2 (1) (2007) 57–81.
[8] F. Miniati, P. Colella, Block-structured adaptive mesh and time refinement for hybrid, hyperbolic + N-body systems, Journal of Computational Physics 227 (1) (2007) 400–430.
[9] J.L. Vay, P. Colella, P. McCorquodale, B. Van Straalen, A. Friedman, D.P. Grote, Mesh refinement for particle-in-cell plasma simulations: applications to and benefits for heavy ion fusion, Laser and Particle Beams 20 (4) (2002) 569–575.