Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 184 (2003) 670–678
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Note

An influence matrix particle–particle particle-mesh algorithm with exact particle–particle correction

J.H. Walther

Institute of Computational Science, ETH Zürich, Hirschengraben 84, CH-8092 Zürich, Switzerland

Received 19 December 2001; received in revised form 26 September 2002; accepted 12 October 2002

Keywords: Particle–particle particle-mesh algorithm; Particle methods; *N*-body problem; Coulomb interaction; Electrostatic interaction; Influence matrix

1. Introduction

The classical N-body problem consisting of the interaction of N computational particles, is the key computational issue in a number of diverse fields of science including chemistry, fluid dynamics, and astrophysics. For particles interacting through long range potentials the direct evaluation of their mutual interaction nominally scales as $\mathcal{O}(N^2)$. In order to reduce this computational cost a simple truncation of the interaction potential appears to be a viable approach for homogeneous (neutral) systems [1], reducing the operational count to $\mathcal{O}(N)$. While this approximation may suffice for the modelling of the electrostatic interaction in molecular dynamics (MD) simulations, the assumption is insufficient for problems in fluid dynamics and astrophysics. In these cases, and for MD simulations requiring higher accuracy than warranted by a simple truncation, fast methods have been devised such as the Barnes–Hut [3] and the fast multipole method [7], as well as hybrid mesh based algorithms including the particle-mesh (PM) and the particle–particle particle-mesh (PPPM) algorithm originally proposed by Hockney and Eastwood [9].

While fast multipole methods offer an operational cost of $\mathcal{O}(N)$ and an exact enforcement of the free-space boundary condition, hybrid particle-mesh algorithms with an operational cost of $\mathcal{O}(N\log N)$ or $\mathcal{O}(N)$ are often found to be computationally superior for problems in simple geometries and for periodic systems. The particle-mesh algorithm is based on the observation that the three-dimensional, free-space Green's function to the Laplace operator (∇^2) is $1/4\pi r$, and they attain their efficiency by employing fast Fourier transforms or fast iterative solvers for the solution of the field equation on a mesh. The density field is constructed on the mesh from the strength of the particles using a smooth projection, which also serves to interpolate the computed force field back onto the particles.

For particle systems involving non-smooth density fields the exact force field will contain sub-grid scales not resolved by the PM algorithm, and an explicit particle–particle (PP) correction term ($\mathbf{f}^{\mathrm{cc}}$) is required to

E-mail address: walther@inf.ethz.ch.

0021-9991/02/$ - see front matter
PII: S0021-9991(02)00035-9

resolve these scales resulting in the PPPM algorithm. The sub-grid scales are generally anisotropic as they depend on the relative position of the particles on the mesh. The PPPM algorithm presented by Hockney and Eastwood [9] reduces this anisotropy by applying an optimized Green's function in Fourier space, taking into account the projection steps and the differential operators involved in the particle-mesh algorithm. Moreover, the modification and use of the Fourier components of the Green's function allows free-space boundary conditions [6,8], an advantage which is not commonly recognized. However, the required access to Fourier space limits the choice of field solver, which may be an important factor for an efficient implementation on parallel computer architectures [4].

Alternatively, one may obtain an isotropic sub-grid scale by providing sufficient smoothing during the projection of the charge density. This methodology was employed by Beckers et al. [4] in a parallel, iterative, finite difference based PPPM algorithm. In a two step projection procedure, the particle charge is first projected onto the mesh using linear interpolation, and in a second step redistributed using a Gaussian projection with a wide kernel including some 500 grid points. The cost of this second step is reduced by replacing the redistribution with an elliptic smoother, effectively solving the heat equation to match the width of the Gaussian kernel. The associated sub-grid scales are assumed to be isotropic and given by the exact free-space integration of the Gaussian charge distribution. The rms errors in the force and potential energy was found to be less than 1% for a relative cutoff of $r_c/h = 6$, where r_c is the spherical cutoff distance of the particle–particle correction, and h denotes the mesh spacing.

Another free-space PPPM algorithm that allows finite difference based solvers [1] was proposed by Theuns [14] for application in astrophysics. In this method, the resolved, anisotropic force field is estimated from the projected density field by evaluating the sub-grid scales in real space. The algorithm proceeds by subtracting this estimate for the resolved field for particles in close proximity, leaving a particle–particle correction term of the form r^{-1}. The errors in the force and potential were found to be approximately 6%.

The present PPPM algorithm is similar to the method of Theuns [14] but uses an accurate influence matrix technique to allow specific cancellation of the anisotropic sub-grid scales without requiring the access to Fourier space and hence allows the use of other fast field solvers. The linearity of the problem secures an exact particle–particle correction term, improving the overall accuracy of the method compared to [14], and at the same computational cost.

The remainder of this Note is organized as follows: the governing equations are outlined in Section 2. The particle-mesh and the original Hockney and Eastwood PPPM algorithm are described in Sections 3.1 and 3.2. The proposed influence matrix PPPM algorithm is described in Section 3.3, and the results and conclusion are presented in Sections 4 and 5, respectively.

2. Governing equations

In the following, we shall consider problems in electrostatics with particles interacting through the three-dimensional Coulomb potential

$$\phi(\mathbf{x}_i;\mathbf{x}_j) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \tag{1}$$

where $\mathbf{x}_i$ and q_i denote the position and charge of the ith particle, and ϵ_0 is the permittivity of vacuum. Since the three-dimensional Green's function to the Laplacian is $1/4\pi r$, the electrostatic potential ($\Phi(\mathbf{x}_i) = q_i\phi(\mathbf{x}_i)$) is governed by the Poisson equation

[1] A fast FFT solver was applied in [14].

$$\nabla^2 \Phi = -\frac{\rho}{\epsilon_0}, \tag{2}$$

where ρ is the charge density

$$\rho(\mathbf{x}) = \int\int\int \delta(\mathbf{x} - \mathbf{x}_i) q_i \, \mathrm{d}\mathbf{x}, \tag{3}$$

and $\delta(\mathbf{x})$ is the Dirac delta function.

The electrostatic field ($\mathbf{E}$) is computed from the potential as

$$\mathbf{E} = -\nabla \Phi, \tag{4}$$

and the electrostatic force acting on the particles as $\mathbf{f}_i = q_i \mathbf{E}(\mathbf{x}_i)$.

3. The particle–particle particle-mesh algorithm

The PPPM algorithm comprises of two main parts: a PM step which allows solution of the smooth component of the force field, and a particle–particle (PP) correction term, which allows accurate treatment of the non-smooth part of the density field.

3.1. The particle-mesh algorithm

The particle-mesh algorithm proceeds by constructing the charge density field on the mesh from the charge carried by the particles using a smooth projection

$$\tilde{\rho}(\mathbf{x}_m) = \frac{1}{h^3} \sum_i^N W(\mathbf{x}_i - \mathbf{x}_m) q_i, \tag{5}$$

where the subscript m refers to mesh quantities, $\tilde{\rho}$ is the smooth density field, and $W(\mathbf{x})$ is the projection kernel. Commonly used kernels are moment conserving splines such as the linear Cloud-In-Cell (CIC), the Triangular Cloud Shape (TCS), and higher order kernels [5,10]. The electrostatic field ($\tilde{\mathbf{E}}_m$) is computed from Eq. (4) on the mesh using finite differences or direct differentiation of the Fourier components when a Fourier solver is employed. The resolved electrostatic force ($\widetilde{\mathbf{f}}_i$) acting on the particles is finally obtained by interpolation

$$\tilde{\mathbf{f}}_i = q_i \sum_m^M W(\mathbf{x}_i - \mathbf{x}_m) \, \tilde{\mathbf{E}}_m, \tag{6}$$

where M is the number of mesh points involved in the projection step.

The solution of Eqs. (2), (5), and (6) forms the basis of the particle-mesh algorithm, which requires that Eq. (5) is a close approximation to Eq. (3). This is generally true for collisionless systems, i.e., for systems where the mean particle spacing is greater than the mesh spacing, or for systems having smooth density fields. For system involving collisions or discontinuous density fields the true force field, ($\mathbf{f}_i$) contains sub-grid scales ($\tilde{\tilde{\mathbf{f}}}_i$) such that

$$\mathbf{f}_i = \tilde{\mathbf{f}}_i + \tilde{\tilde{\mathbf{f}}}_i. \tag{7}$$

For a specific particle configuration the magnitude and distribution of the sub-grid scales will generally depend on the smoothness and the moment conserving properties of the projection kernel. The loss of

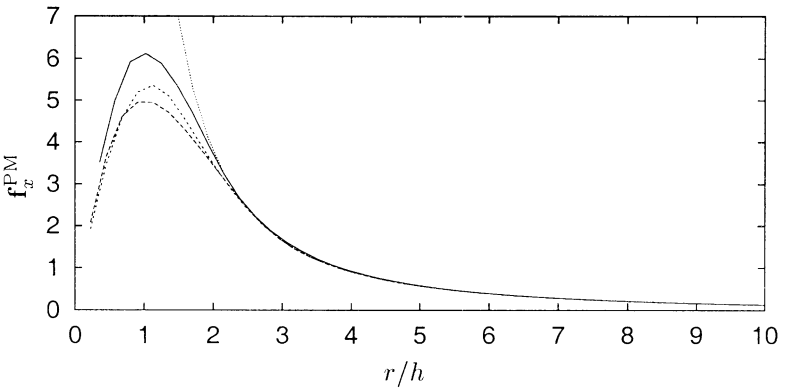

Fig. 1. The particle-mesh resolved periodic force field ($\tilde{\mathbf{f}}_i$) induced by a unit charge placed at different relative positions of the central mesh cell of a 32^3 mesh. The x component of the force field is measured along the x–y diagonal, and the solution is obtained using the TCS kernel and fourth order finite differences. Particle locations: —, $(0,0)$; – –, $(\frac{1}{2}h,0)$; - - -, $(\frac{1}{2}h,\frac{1}{2}h)$; $\cdots$, the exact force field is computed directly using $100 \times 100 \times 100$ image particles.

moments incurred in the projection step, and the accuracy and directional sensitivity of the differential operators applied on the mesh are furthermore responsible for the anisotropy of the resolved scales as shown in Fig. 1. The resolved electrostatic force acting between two particles does not only depend on their mutual distance, but also on their relative position on the mesh.

3.2. The Hockney and Eastwood particle–particle correction

The particle–particle particle-mesh algorithm proposed by Hockney and Eastwood [9] is based on an integral form of the Poisson equation (Eq. (2))

$$\Phi(\mathbf{x}) = \int\int\int G(\mathbf{x}-\mathbf{y})\rho(\mathbf{y})\,\mathrm{d}\mathbf{y} = G * \rho. \tag{8}$$

The convolution (Eq. (8)) is evaluated in Fourier space

$$\hat{\Phi} = \hat{G}\hat{\rho}, \tag{9}$$

and employs an optimized Green's function ($G = G_{\mathrm{opt}}$) to enforce a prescribed, isotropic, and hence radially symmetric sub-grid scale. Thus,

$$\tilde{\tilde{\mathbf{f}}}(\mathbf{x}_i;\mathbf{x}_j) \approx \tilde{\tilde{\mathbf{f}}}(|\mathbf{x}_i-\mathbf{x}_j|), \tag{10}$$

and

$$\tilde{\mathbf{f}}(\mathbf{x}_i;\mathbf{x}_j) \approx \tilde{\mathbf{f}}(|\mathbf{x}_i-\mathbf{x}_j|) \equiv \mathbf{R}(|\mathbf{x}_i-\mathbf{x}_j|), \tag{11}$$

where $\mathbf{R}(|\mathbf{x}_i-\mathbf{x}_j|)$ is a prescribed function, and $\tilde{\tilde{\mathbf{f}}}(\mathbf{x}_i;\mathbf{x}_j)$ and $\tilde{\mathbf{f}}(\mathbf{x}_i;\mathbf{x}_j)$ denote the sub-grid and resolved fields at $\mathbf{x}_i$ induced by a unit charge located at $\mathbf{x}_j$, respectively. Commonly used sub-grid scale functions are $\tilde{\tilde{f}}(r) = r^{-1}\mathrm{erfc}(ra^{-1})$ and $\tilde{\tilde{f}}(r) = (r^{-1}-a^{-1})H(a-r)$, where a is an adjustable parameter, and H is the Heaviside function. The optimal Green's function is determined by combining Eqs. (5), (9), (4), and (6), and minimizing the rms error between $\tilde{\mathbf{f}}(\mathbf{x}_i;\mathbf{x}_j)$ and $\mathbf{R}(|\mathbf{x}_i-\mathbf{x}_j|)$ with respect to the discrete Green's function (G_{opt}) see Hockney and Eastwood [9, p. 273]. The prescribed resolved field ($\mathbf{R}$), and hence the sub-grid scale term is computed during an initialization step of the algorithm and tabulated for efficient calculation. Thus,

the particle–particle correction amounts to adding the isotropic sub-grid scale during the PP step of the algorithm

$$\mathbf{f}_i = \tilde{\mathbf{f}}_i + \sum_j \tilde{\tilde{\mathbf{f}}}(|\mathbf{x}_i - \mathbf{x}_j|), \quad \text{for } |\mathbf{x}_i - \mathbf{x}_j| < r_c, \tag{12}$$

where r_c is the cutoff radius.

3.3. An influence matrix particle–particle correction

The present influence matrix particle–particle particle-mesh algorithm is similar to the method of local correction by Anderson [2] and to the method of Theuns [14] by providing an estimate for the resolved field ($\tilde{\mathbf{E}}$) on the mesh. In [14], this estimate is compute in real space from the charge density ($\tilde{\rho}$) on the mesh

$$\tilde{\mathbf{E}}(\mathbf{x}_m) \approx \frac{h^3}{4\pi\epsilon_0} \sum_i^L \tilde{\rho}_i \frac{\mathbf{x}_i - \mathbf{x}_m}{|\mathbf{x}_i - \mathbf{x}_m|^3}, \tag{13}$$

where $\mathbf{x}_i$ and $\mathbf{x}_m$ denote the mesh points, and L is the number of mesh points involved in the estimate. The present method replaces Eq. (13) with an accurate influence matrix technique, representing an exact account for the approximations and anisotropy incurred by the differential operators on the mesh. Thus, during the projection step of the algorithm, the resolved electrostatic field ($\tilde{\mathbf{E}}$) induced by a collection of particles contained in a grid cell (cf. Fig. 2) is pre-computed using the influence matrix

$$\widetilde{\boldsymbol{\delta E}} = \mathbf{C}\widetilde{\boldsymbol{\delta\rho}}, \tag{14}$$

where C_{ij} are the components of the influence matrix describing the electrostatic field $\widetilde{\boldsymbol{\delta E}}_i$ at the ith grid point induced by the charge density $\widetilde{\delta\rho}_j$ at the jth grid point. The influence matrix is a $M \times (3 \times L)$ matrix, where M is the number of mesh points involved in the projection (Eq. (5)), and L is the number of grid

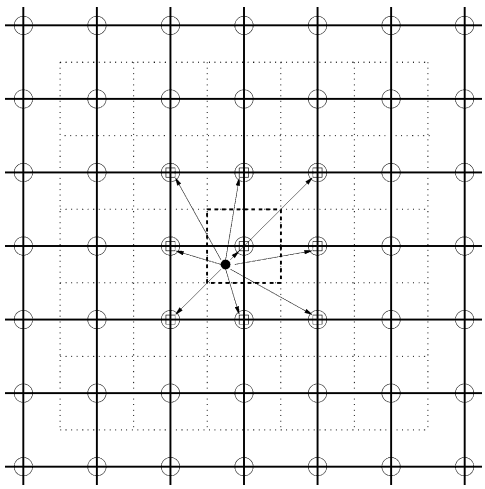

Fig. 2. Two-dimensional schematic of the PPPMi algorithm for the TCS projection kernel and two neighbouring cells ($K = 2$) in the particle–particle correction. The particles (●) contained within the dashed box contribute to the charge distribution of the (3×3) mesh points marked with a box (□). The mesh points included in the influence matrix are marked with a circle (○). The dotted lines indicate the staggered mesh and ordering required by the TCS projection kernel.

points included in the particle–particle correction. The factor of three results from the three components of the **E**-field. The particle–particle correction is subjection to a Cartesian cutoff, and for K neighbouring grid boxes included in the cutoff, the TCS projection kernel result in $L = (2K+3)^3$ cf. Fig. 2. The estimated resolved (PM) electrostatic field ($\tilde{\mathbf{E}}$) is interpolated onto the particles in the neighbouring cells using Eq. (6) and subtracted to cancel the contribution added by the PM part of the algorithm. The subsequent PP step involves an exact r^{-1} correction term (hence $\mathbf{f}_i^{\mathrm{cc}} = (4\pi\epsilon_0)^{-1}\mathbf{x}/|\mathbf{x}|^3$) for free-space problems and the r^{-1} term including the periodic images for periodic systems. For the periodic system, the explicit inclusion of the images is computationally expensive and are avoided by applying the influence matrix corresponding to the free-space problem. Hence, only the free-space component of the resolved force field is annihilated, preserving the periodic images.

The influence matrix is computed as part of the initialization step of the algorithm by placing M test particles of unit charge at different locations within a single grid cell. The density field is projected using Eq. (5), and the Poisson equation (Eq. (2)) is solved using the fast Poisson solver applied in the PM part of the algorithm. The electrostatic field is computed from Eq. (4) and forms the right-hand side of

$$\mathbf{C}\left\{\widetilde{\delta\rho}_j\right\}_k = \left\{\widetilde{\delta\mathbf{E}}_i\right\}_k, \tag{15}$$

where $\left\{\widetilde{\delta\rho}_j\right\}_k$ is a $M \times M$ matrix containing the charge density at the jth grid point as created by the kth test particle, and $\left\{\widetilde{\boldsymbol{\delta E}}_i\right\}_k$ is a $M \times (3 \times L)$ matrix containing the corresponding electrostatic field at the ith grid point. The influence matrix (**C**) is computed solving the linear system of equations (Eq. (15)). The actual position of the test particles was found to have a negligible influence on the influence matrix provided the test points are distinct.

The present influence matrix particle–particle particle-mesh algorithm (PPPMi) involves the following steps:

1. Projection of the charge density field:
   (a) Sort the particles in groups of particles that will assign charge density to identical grid points.
   (b) For each of these groups:
      (i) Perform the projection to obtain the contribution to the charge density ($\delta\rho$) to the surrounding grid points

$$\widetilde{\delta\rho}(\mathbf{x}_m) = \frac{1}{h^3}\sum_i^P W(\mathbf{x}_i - \mathbf{x}_m)q_i, \tag{16}$$

where P is the number of particles in the group.

      (ii) Compute the induced electrostatic field at the neighbouring grid cells from the charge density using the influence matrix ($\widetilde{\delta\mathbf{E}} = \mathbf{C}\widetilde{\delta\rho}$).
      (iii) For each of the neighbouring cells, project $\widetilde{\delta\mathbf{E}}$ to the particles and subtract this contribution

$$\mathbf{f}_i^a = q_i\sum_m^M W(\mathbf{x}_i - \mathbf{x}_m)\widetilde{\delta\mathbf{E}}_m. \tag{17}$$

      (iv) Compute the density field ($\widetilde{\rho}$) from the contributions $\widetilde{\delta\rho}$.
2. Solve the Poisson equation ($\nabla^2\Phi = -\tilde{\rho}/\epsilon_0$) on the mesh.
3. Compute the resolved electrostatic field ($\tilde{\mathbf{E}} = -\nabla\Phi$) on the mesh.
4. Project the electrostatic field onto the particles and compute the resolved particle forces

$$\tilde{\mathbf{f}}_i = q_i\sum_m^M W(\mathbf{x}_i - \mathbf{x}_m)\widetilde{\mathbf{E}}_m. \tag{18}$$

5. Compute the particle–particle correction between particles in neighbouring groups

$$\mathbf{f}_i^c = \sum_j^Q \mathbf{f}^{cc}(|\mathbf{x}_i - \mathbf{x}_j|), \tag{19}$$

where Q denotes the number of particles contained in the groups.

6. Compute the total particle force as:

$$\mathbf{f}_i = \tilde{\mathbf{f}}_i - \mathbf{f}_i^a + \mathbf{f}_i^c. \tag{20}$$

The present PPPMi algorithm allows efficient implementation on distributed memory computer architectures due to the unrestricted choice of Poisson solver. The computational efficiency of the algorithm can furthermore be improved at the expense of increased memory usage, by storing the induced electrostatic field ($\widetilde{\delta E}$) separately for each of the particle groups. Hence, the repeated projection step (1(b)iii) conducted for each of the $(2K+1)^3$ neighbours can be performed after the main projection of the density field (see also [11]).

4. Results

To study the accuracy of the proposed PPPMi algorithm, we consider two particles of unit charge placed in a cubic, periodic domain (Fig. 1). The electrostatic force acting between the particles is computed as function of the interparticle spacing as one of the particles is displaced from the other along the x–y diagonal of the domain. The employed field solver [13] uses second order finite differences, but fourth order accuracy is obtained by the method of deferred corrections [12]. The solution is computed for a mesh resolution of 32^3 and 64^3, respectively, applying the TCS projection kernel and a relative cutoff (r_c/h) of 2 and 3. The PPPM force is compared with the estimated exact force using $100 \times 100 \times 100$ images. The resolved force field with explicit annihilation of the anisotropic sub-grid scales ($\tilde{\mathbf{f}}_i - \mathbf{f}_i^a$) and the particle–particle correction term ($\mathbf{f}_i^c$) are shown in Fig. 3 for a relative cutoff of 3 corresponding to $r/h \approx 5.0$. The corresponding error in the force is shown in Fig. 4. For a particle spacing below the cutoff, the error is negligible and within the accuracy of the estimated exact solution. The maximum error occurs at the cutoff distance, with errors of 1.1% and 0.3% for a relative cutoff of 2 and 3, respectively. The results indicate that

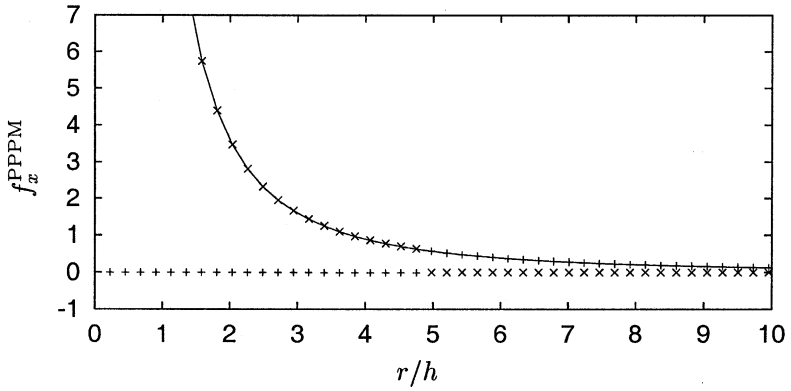

Fig. 3. The PPPMi force field for two particle in a periodic domain (Fig. 1). The force contributions are: + + +, the particle mesh force field with cancellation of the sub-grid scale terms ($\tilde{\mathbf{f}}_i - \mathbf{f}_i^a$); × × ×, the particle–particle correction term ($\mathbf{f}_i^c$); —, the exact force field.

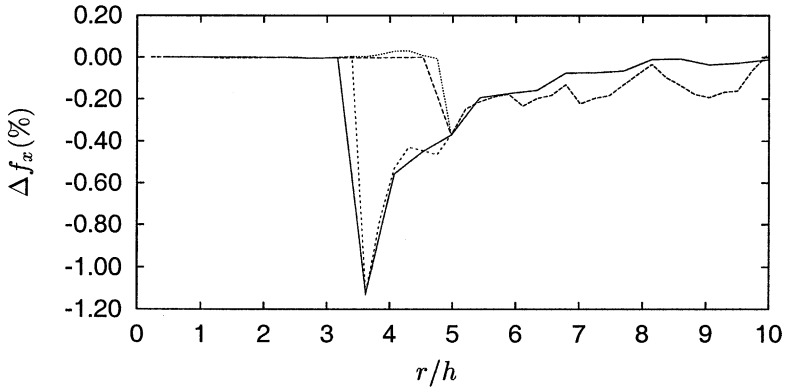

Fig. 4. The error (Δf_x) in the f_x component of the PPPMi force for different mesh resolutions and relative cutoff radii for the particle configuration shown in Fig. 1. —: 32^3 mesh points, $r_c/h = 2$, – – –: 32^3 mesh points, $r_c/h = 3$; - - -: 64^3 mesh points, $r_c/h = 2$; · · ·: 64^3 mesh points, $r_c/h = 3$.

the maximum error is insensitive to the mesh resolution for constant relative cutoffs, but with a reduced mean error as the mesh is refined.

Finally, the convergence of the proposed algorithm is demonstrated for a system of 10^3 particles uniformly distributed within the computational domain and assigned a uniform, random charge distribution in the range $[-\frac{1}{2}, \frac{1}{2},]$. The root mean square error $\Delta f_{\text{rms}} = f_{\text{mean}}^{-1}(1/N \sum_i^N |\mathbf{f}_i - \mathbf{f}_{\text{exact}}|^2)^{1/2}$ and the maximum error $\Delta f_{\text{max}} = f_{\text{mean}}^{-1} \max_i |\mathbf{f}_i - \mathbf{f}_{\text{exact}}|$, f_{mean} denoting the mean force, are shown in Table 1 for mesh resolutions (N_g) of 16^3, 32^3, and 64^3, respectively. The particle–particle correction extends from $K = 2$ to 8 neighbours. Both second and fourth order finite difference solutions are considered, and the error is computed from the solution obtained on a 128^3 mesh using 16 neighbours. Second and fourth order convergence is recovered from the test cases (cf. Table 1) with a mean error of 2% for the second order solution computed on a 16^3 mesh using $K = 2$ neighbours, and 0.005% for the fourth order solution obtained on a 64^3 mesh using $K = 8$ neighbours.

Table 1
Convergence study of the root mean square error (Δf_{mean}), and the maximum error (Δf_{max}) of the electrostatic force acting between 1000 randomly distributed charged particles in a cubic and periodic domain

N_g	K	FD	Δf_{mean} (%)	Δf_{max} (%)
16	2	4	0.868	5.527
16	2	2	2.132	15.608
16	3	4	0.076	0.947
32	3	4	0.115	0.980
32	4	4	0.039	0.270
32	4	2	0.738	8.319
64	3	4	0.178	1.693
64	4	4	0.067	0.480
64	4	2	1.161	8.143
64	8	4	0.005	0.066
64	8	2	0.230	2.317
128	16	4	–	–

The particle charge is sampled from a uniform distribution, and the solution is obtained using second and fourth order finite difference (FD) on a 16^3, 32^3, and 64^3 mesh resolution (N_g), respectively. The TCS projection kernel is used throughout and the results are compared with the solution obtained on a 128^3 mesh using 16 neighbours in the particle–particle correction ($K = 16$).

5. Conclusion

An improved particle–particle particle-mesh algorithm has been presented for the fast solution of N-body problems. The proposed algorithm uses an influence matrix technique to annihilate the anisotropic sub-grid scales incurred by the truncation of the moments by the projection step and by the truncation errors of the finite differences typically used in fast iterative Poisson solvers. The subsequent particle–particle correction step involves the exact r^{-1} term. The algorithm allows the use of "black-box" fast Poisson solvers, hence offering greater implementational flexibility than the original PPPM algorithm.

The methodology has been extended to problems in fluid mechanics using the immersed boundary technique for the particle (vortex) method, and will be presented in a separate paper [11]. The PPPMi algorithm furthermore allows consistent treatment of higher order singularity distributions commonly used in boundary element methods.

Acknowledgements

I would like to thank Professor Petros Koumoutsakos and Thomas Werder for their valuable comments on an earlier version of this paper, and Professor Andrew Pohorille for initiating this work. The support from the Research Commission at ETH Zürich, and from the CTR Summer Program 2000 Stanford University is greatly appreciated.

References

[1] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Clarendon Press Oxford, Oxford, 1987.
[2] C.R. Anderson, A method of local corrections for computing the velocity field due to a distribution of vortex blobs, J. Comput. Phys. 62 (1) (1986) 111–123.
[3] J. Barnes, P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, Nature 324 (4) (1986) 446–449.
[4] J.V.L. Beckers, C.P. Lowe, S.W. De Leeuw, An iterative PPPM method for simulating Coulombic systems on distributed memory parallel computers, Mol. Sim. 20 (6) (1998) 369–383.
[5] G.-H. Cottet, P. Koumoutsakos, Vortex Methods—Theory and Practice, Cambridge University Press, New York, 2000.
[6] J.W. Eastwood, D.R.K. Brownrigg, Remarks on the solution of Poisson's equation for isolated systems, J. Comput. Phys. 32 (1979) 24–38.
[7] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[8] R.W. Hockney, The potential calculation and some applications, Methods Comput. Phys. 9 (1970) 136–210.
[9] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, second ed., IOP, 1988.
[10] J.J. Monaghan, Particle methods for hydrodynamics, Comp. Phys. Repts. 3 (1985) 71–123.
[11] J.H. Walther, G. Morgenthal, An immersed interface method for the vortex-in-cell algorithm, J. Turbul. 3 (2002) 1–9.
[12] V. Pereyra, On improving the approximate solution of a functional equation by deferred corrections, Numer. Math. 8 (1966) 376–391.
[13] Green Mountain Software, Crayfishpak, User's guide, P.O. Box 86124, Madeira Beach, Florida 33738, USA, 1990.
[14] T. Theuns, Parallel P3M with exact calculation of short range forces, Comp. Phys. Commun. 78 (1994) 238–246.