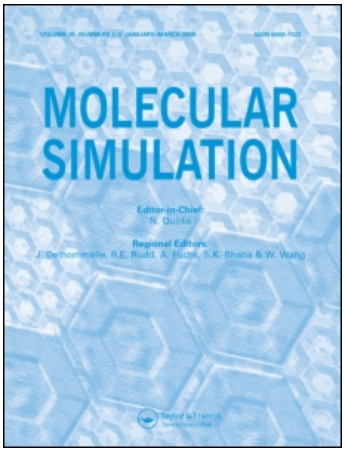
This article was downloaded by:
On: *14 January 2011*
Access details: *Access Details: Free Access*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

An Iterative PPPM Method for Simulating Coulombic Systems on Distributed Memory Parallel Computers

J. V. L. Beckers[a]; C. P. Lowe[a]; S. W. De Leeuw[a]
[a] Department of Applied Physics, Delft University of Technology, Delft, The Netherlands

To cite this Article Beckers, J. V. L. , Lowe, C. P. and De Leeuw, S. W.(1998) 'An Iterative PPPM Method for Simulating Coulombic Systems on Distributed Memory Parallel Computers', Molecular Simulation, 20: 6, 369 — 383
To link to this Article: DOI: 10.1080/08927029808022044
URL: http://dx.doi.org/10.1080/08927029808022044

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Molecular Simulation, 1998, Vol. 20, pp. 369–383
Reprints available directly from the publisher
Photocopying permitted by license only

© 1998 OPA (Overseas Publishers Association)
Amsterdam B.V. Published under license
under the Gordon and Breach Science
Publishers imprint.
Printed in India.

AN ITERATIVE PPPM METHOD FOR SIMULATING COULOMBIC SYSTEMS ON DISTRIBUTED MEMORY PARALLEL COMPUTERS

J. V. L. BECKERS, C. P. LOWE and S. W. DE LEEUW

Delft University of Technology, Department of Applied Physics, Lorentzweg 1, 2628 CJ, Delft, The Netherlands

(Received April 1997; accepted August 1997)

Downloaded At: 19:08 14 January 2011

We describe results obtained from a new implementation of Hockney's Particle-Particle Particle-Mesh (PPPM) method for evaluation of Coulomb energies and forces in simulations of charged particles. Rather than taking the usual approach, solving Poisson's equation by means of a Fourier transformation, we use an iterative Poisson solver. In a molecular dynamics (MD) simulation the solution from the previous time-step provides a good starting point for the next solution. This reduces the number of iterations per time-step to acceptable values. The iterative scheme has a complexity $\mathcal{O}(N)$, and, in contrast with the Fourier transform based approach, it is easily implemented on a parallel architecture with a minimum of communication overhead.

We examine the origin of the errors in the algorithm and find that reasonable accuracies in the Coulomb interaction can best be attained by making the charge density profile as smooth as possible. This involves spreading the particle charges over a large number of grid points. Assigning these charges then becomes the most time consuming part of the algorithm. We show how we can then gain a considerable saving in computing time by employing a diffusion equation as a charge spreading mechanism.

The effect of employing the algorithm with an accuracy less than that typically tolerated in an Ewald summation is studied by computing, from an MD simulation of silica, quantities that are sensitive to the long range part of the Coulomb interaction. These results are compared to full Ewald sum reference simulations and found to be within the statistical error.

Keywords: Particle-particle particle-mesh; Coulomb interaction; electrostatic interaction; parallel algorithms; molecular simulation

1. INTRODUCTION

By truncating the interaction potential and using neighbour lists, the computational effort required to perform a molecular dynamics simulation

usually scales linearly with the number of particles N. However, because of the long range of the Coulomb potential, such a truncation in a system which contains electrostatic charges can lead to incorrect behaviour [1, 2]. We therefore need to employ an efficient method to evaluate the full untruncated Coulomb interaction and we want it to scale linearly with N, to make it feasible to study large systems.

The most widely used approach is Ewald's method, where the long-ranged part of the interaction is computed in Fourier space. For most small systems (up to 10^3 atoms) this method is very satisfactory, but for larger systems we run into problems. Even an optimized implementation of the Fourier part scales as $\mathcal{O}(N^{3/2})$ [3], and for N larger than 10^4 the method becomes too time-consuming. For parallel implementations, the Fourier part requires a costly global communication over all processors, causing communication overhead.

Several methods have been developed in an attempt to improve on the Ewald scaling. The Fast Multipole method (FMM) [4] has complexity $\mathcal{O}(N)$, and can be implemented in parallel. The FMM implementations need to be thoroughly optimised [5, 6] in order to reach competitive performance. Writing an optimised FMM code is elaborate, especially for non-cubic cells, which is probably why the FMM is less frequently used. There also remains some debate over the size of system required before FMM becomes competitive with Ewald's summation [7, 8].

The particle Mesh Ewald method (PME) is a fast Fourier transform (FFT) version of Ewald's method and was shown to be competitive in the regime of $N = 10^4 - 10^5$ [9, 10]. However, the speed of the PME depends on the efficiency of the FFT routine which on some distributed memory architectures is not always optimal.

Here we examine the alternative approach of using the particle-particle particle mesh method (PPPM) developed by Hockney *et al.* [11]. Our aim is to formulate this algorithm in such a way that it can be implemented as efficiently as possible on parallel, distributed memory, computers. The only part of the original algorithm which does not lend itself to parallel implementation is the Fourier transform method used to solve Poisson's equation on a grid. There are, however, other techniques for solving Poisson's equation which are more suitable for parallel implementation, for instance iterative relaxation methods. These are not normally competitive with Fourier transformation but we describe here how, within the context of an MD simulation, they can be made so.

If we can achieve this then there is an additional advantage to using relaxation methods as opposed to Fourier transformation. They are not

Downloaded At: 19:08 14 January 2011

necessarily bound to periodic systems. Non-periodic boundary conditions can easily be implemented by restricting the values of the electrostatic potential at certain grid points. For example a vacuum or some constant dielectric boundary condition could be represented by a plane of grid points at constant potential or potential gradient. In contrast, Fourier based methods can only be used for periodic systems and for so-called "tin-foil" boundary conditions, an ideal metal surface producing mirror images to every charge in the system [13, 14].

Our other aim here is to examine the sources of error in the PPPM algorithm. In contrast to the other methods outlined above, there is no single parameter which determines the degree of error. Rather, in the PPPM, it is a combination of factors. Considering the source of errors leads us to suggest a modified version of the charge assignment part of the algorithm, which we find makes it feasible to simulate systems with sufficient accuracy to reproduce the correct physical behaviour. This point we establish by comparing simulations of silica using an accurate Ewald's summation and our PPPM algorithm and examining quantities sensitive to the long-ranged part of the Coulombic force. The iterative PPPM method at which we arrive has low communication overhead, making it highly suited for computing systems that strongly depend on scalability, and is competitive with Ewald's summation for moderate system sizes of a few thousand particles.

Downloaded At: 19:08 14 January 2011

2. METHOD

2.1. The PPPM Method

We begin with a brief summary of the PPPM method. This was originally developed by Hockney and Eastwood in the early 70's, and used with small variations ever since [6, 10]. The Coulomb potential is split up in a short range direct interaction part (PP) and a contribution from a mesh (PM).

$$E_i^{\text{Coulomb}} = \sum_{j<i} E_{ij}^{\text{direct}} + E_i^{\text{mesh}} \tag{1}$$

The advantage of this splitting is that the interaction between particles which are close together is still computed essentially directly (and hence accurately). This is important when charged atoms are bonded to each other in a molecule.

The direct part of the Coulomb energy of a pair of particles i, j separated by $\boldsymbol{r}_{ij}$ is given by Coulomb's law minus a correction term:

$$E_{ij}^{\text{direct}}(\boldsymbol{r}_{ij}) = \begin{cases} q_i q_j \left(\frac{1-E^c(r_{ij})}{4\pi\varepsilon|r_{ij}|}\right) & \text{for } |r_{ij}| < R_e \\ 0 & \text{for } |r_{ij}| \geq R_e \end{cases} \tag{2}$$

where q_i is the charge of particle i and ε is the permittivity of vacuum. The correction $E^c(r)$ is the portion of the interaction already covered by the mesh potential, to avoid double-counting of interactions. The direct contribution should vanish at the direct interaction cut-off R_c, and for $|r_{ij}| \geq R_c$ there is only a mesh contribution. The exact form of $E^c(r)$ we shall come to later (see Eq. 12).

The mesh potential is obtained by assigning the particle charges to grid points and then solving Poisson's equation on that grid:

$$\nabla^2\phi(\boldsymbol{R}) = -\frac{1}{\varepsilon}\rho(\boldsymbol{R}) \tag{3}$$

where $\rho(\boldsymbol{R})$ and $\phi(\boldsymbol{R})$ are the charge density and electrostatic potential at grid point $\boldsymbol{R}$. The charge density $\rho(\boldsymbol{R})$ is defined as the grid point charge per grid cell volume:

$$\rho(\boldsymbol{R}) = \frac{q(\boldsymbol{R})}{h_x h_y h_z} \tag{4}$$

where h_x, h_y, h_z are the mesh spacings in dimensions x, y, z. The charges $q(\boldsymbol{R})$ are obtained by assigning the particle charges to the grid points in two steps. In the first step we use a linear charge assignment scheme [11]. Every particle charge q_i is distributed over its 8 surrounding grid points, and the charge on a grid point $\boldsymbol{R}$ is computed as:

$$q(\boldsymbol{R}) = \sum_{i=1}^{N} q_i W(\boldsymbol{r}_i - \boldsymbol{R}) \tag{5}$$

where $W = W_x W_y W_z$ is the weight of a particle at $\boldsymbol{r}_i$ on the grid point $\boldsymbol{R}$:

$$W_x(r_{i,x} - R_x) = \begin{cases} 1 - \frac{|r_{i,x} - R_x|}{h_x} & \text{if } |r_{i,x} - R_x| < h_x \\ 0 & \text{if } |r_{i,x} - R_x| \geq h_x. \end{cases} \tag{6}$$

A higher order (27 points) assignment scheme has been reported to give better accuracy [6]. We have not observed such an improvement in our

Downloaded At: 19:08 14 January 2011

implementation, probably because of our use of a second charge assignment step.

In the second charge assignment step the charges are spread out over a number of surrounding grid points (typically $\sim$ 500) to produce a smooth total charge distribution. A smooth charge density profile increases the accuracy in electrostatic forces and energies because the largest errors are made where the gradients in charge density $\rho(\boldsymbol{R})$ and electrostatic potential $\phi(\boldsymbol{R})$ are large.

The most straightforward implementation of this second step is to use a predefined assignment function $f(|R-R_0|)$. Every charge $q(\boldsymbol{R}_0)$ found in the previous step is redistributed over all grid points enclosed in a sphere of a radius R_{assign}, which we set equal to the direct interaction radius R_c.

$$q'(\boldsymbol{R}) = \begin{cases} q(\boldsymbol{R}_0) f(|R - R_0|) & \text{if } |R - R_0| < R_c \\ 0 & \text{if } |R - R_0| \geq R_c \end{cases} \tag{7}$$

Where $q'(\boldsymbol{R})$ is the charge assigned to a grid point at $\boldsymbol{R}$ from $q(\boldsymbol{R}_0)$. There are several suitable assignment functions [11]. The linear and Gaussian shaped $f(r)$ are:

$$f_{\text{linear}}(r) = 1 - \frac{r}{R_c} M^{-1} \tag{8}$$

$$f_{\text{Gauss}}(r) = \sigma^{-3}\pi^{2/3} \exp\left(-\frac{r^2}{\sigma^2}\right) M^{-1} \tag{9}$$

The M^{-1} normalises the sum of $f(r)$ over all grid points within R_e to 1. We found, of these two, the linear function gives best results, probably because it is the same form as the assignment function in the first step and does not suffer from any small discontinuities. However, the Gaussian distribution gave almost comparable results if σ is set at $R_c/3$. For reasons which we will describe later, we concentrate on the latter.

The method now proceeds by evaluating the electrostatic potential on the mesh (by solving Poisson's equation, Eq. 3). The particle energy and force are computed as weighted sums over the same grid points used in the first step of the charge assignment. By using the same weights W (Eq. 6) for the charge assignment and force evaluation we ensure conservation of momentum in the particle dynamics [11].

$$E_i^{\text{mesh}} = q_i \sum_{\boldsymbol{R}} W(\boldsymbol{r}_i - \boldsymbol{R})\, \phi(\boldsymbol{R}) - E_{\text{Gauss},i}^{\text{self}} \tag{10}$$

Downloaded At: 19:08 14 January 2011

We finally need to correct for the mesh potential that a particle is experiencing from its own charge distribution, the self-energy. This is a constant term per particle given by:

$$E^{\text{self}}_{\text{Gauss},i} = \frac{q_i^2}{2\pi^{3/2}\varepsilon\sigma} \tag{11}$$

The correction term $E^c(|r|)$ in Eq. 2 for the Gaussian charge distribution is given by:

$$E^c_{\text{Gauss}}(r) = erf\left(\frac{r}{\sigma}\right) \tag{12}$$

We note here that the above readily generalises to the case where the dimensions of the simulation box are non-cubic or even fluctuate. Application of the method to constant pressure molecular dynamics is in principle straightforward.

Downloaded At: 19:08 14 January 2011

2.2. Modifications of the Algorithm

One may a priori expect that the accuracy of the PPPM algorithm depends largely on two parameters – the mesh spacing and the cut-off radius. If the cut-off radius is fixed and the mesh spacing decreased then clearly we will have a better solution to Poisson's equation for that charge distribution. On the other hand, if the mesh spacing is kept constant and the cut-off radius is increased then the charge distribution becomes smoother, and the existing mesh should again give a more accurate solution. The magnitude of the long-range contribution will also become a smaller proportion of the total. In order to obtain high accuracy one would therefore like both a fine mesh and a charge assignment function with a cut-off R_c as large as possible. However, the second charge assignment rapidly involves a large number of operations. For a fixed mesh spacing it scales as NR_c^3 (the first step is invariably much faster so we do not consider it further). This second step in the charge assignment scheme rapidly becomes the most time consuming part of the algorithm if we try to increase R_c. Neither is this problem unique to PPPM. A similar observation was made by Peterson for PME [9].

To somewhat alleviate this problem we have implemented this second step in a different way. If, following the first stage of the charge assignment, we solve the diffusion equation for a certain number of time-steps, N_t, then each charge we assigned evolves into a Gaussian of width $\sigma = 2(DN_t)^{1/2}$, where D is the diffusion coefficient and we have set the size of the diffusion time-step

to 1. This is just the analytic solution of the diffusion equation starting from an initial delta function distribution. We can therefore assign a Gaussian charge distribution, but now the number of operations scales as $N_G R_c^2$, where N_G is the total number of gridponts.

The diffusion equation is implemented as (for cubic grids):

$$q(\boldsymbol{R})^{(t+1)} = q(\boldsymbol{R})^{(t)} + D\left[\sum_{NN} q(\boldsymbol{R}_{NN})^{(t)} - 69(\boldsymbol{R})^{(t)}\right] \tag{13}$$

where the $\boldsymbol{R}_{NN}$ are nearest neighbour grid points. This is a relatively crude diffusion scheme, but is adequate for our purposes here. There is an upper limit for the value of the diffusion constant D of (1/6) [12]. However, a certain minimal number of steps is required (20 or so) for the Gaussian to evolve, so, in practice, D is taken to be less than the maximal value (otherwise the cut-off radius actually becomes too large). If the direct interaction cut-off is set at 3σ, the discontinuity at the cut-off is very small and an acceptable level of accuracy in forces and energies is obtained (see Section 3.3).

Using this scheme we have found it possible to use much larger cut-off radii than would otherwise have been practical. The importance of this we address later. In general the diffusion scheme was found to be about four times faster than the direct assignment scheme, and result in a factor of three gain in CPU time, for the parameters used in our MD simulations.

Our second modification to the algorithm is to use a simple iterative successive over-relaxation technique (SOR) as opposed to the Fourier transformation method originally used. As we noted before, this approach is not normally competitive, but if it could be made so, it would offer some benefits. This can be achieved by taking advantage of the information available in a typical molecular dynamics simulation, i.e., the solution from the previous time-step, to obtain a good starting point for the iteration. If we can provide a good enough initial guess so that the number of iterations required to converge is small (strictly such that solving Poisson's equation is not the most time consuming part of the algorithm) then we can conclude that the method is adequate.

The iteration scheme we use is a standard SOR type [12], using odd-even ordering of grid points. The potential $\phi(\boldsymbol{R})$ is updated according to:

$$\text{error}(\boldsymbol{R}) = \omega\left(\frac{1}{\varepsilon}\rho(\boldsymbol{R}) + \nabla^2\phi(\boldsymbol{R})^{\text{old}}\right) \tag{14}$$

Downloaded At: 19:08 14 January 2011

$$\phi(\boldsymbol{R})^{\text{new}} = \phi(\boldsymbol{R})^{\text{old}}\text{error}(\boldsymbol{R}) \tag{15}$$

until the absolute error per grid point is smaller than a threshold value. We found that a tolerance of 0.0001 V/Å^2 per grid point suffices, further iteration has no effect on the accuracy in particle forces, because space discretisation becomes the main source of errors.

The usual practice is to vary the overrelaxation parameter ω during the iteration using the Chesbyshev acceleration procedure. This guarantees that the residual error always decreases. However, if we use the solution from the previous MD time-step as a starting point for the iteration, then the distance from the solution is so small that the optimal overrelaxation parameter (which can be determined from the spectral radius of the mesh [12]), gives the best convergence.

The starting values for electrostatic potential $\phi(t)$ in the SOR iteration can be further improved by storing several solutions from previous time-steps and using them in a Taylor prediction:

$$\phi(t) = \phi(t - \Delta t) + (\phi(t - \Delta t) - \phi(t - 2\Delta t)) + \cdots \tag{16}$$

A better starting value for $\phi(t)$ will reduce the number of iterations required to reach a solution to within acceptable error bounds. However, the contribution of the Taylor terms to $\phi(t)$ decrease with the order of the term. Their significance drops to zero at the point where the accepted iteration error in $\phi(t)$ becomes larger than the Taylor term. We found that for our test simulations only the first order expansion gives a better prediction, and, even in this case, a test for significance was required.

If only the zeroth order prediction (i.e., the previous solution) is used, the fast moving particles can be accelerated by the remaining image of their own potential from the previous time-step. This effect tends to heat up the system and it can be suppressed by iterating to smaller error per grid point. More simply, the efect can be completely avoided by using a first order Taylor prediction.

3. RESULTS

The iterative PPPM algorithm was implemented in existing molecular dynamics code and run on Intel 860 and SGI R5000 processors, the latter being about four times faster. The systems under study are liquid and solid SiO_2 and we use the Vashishta [15] three body potential, where the atom charges are fixed at 1.6 e (silicon) and −0.8 e (oxygen).

Downloaded At: 19:08 14 January 2011

3.1. Scaling

For our test simulation of solid and liquid silica we found that between 3 and 5 iteration steps per MD time-step are needed using a first order Taylor expansion for the iteration starting point. This number of iterations depends slightly on temperature and length of the MD time-step but not on the system size. All other steps in the algorithm scale linear either with the number of particles or with the number of grid points. This implies that we have an overall order $\mathcal{O}(N)$ process. Figure 1 shows linear scaling with the number of particles for systems of vitreous silica (SiO_2) of equal density. Deviations from the ideal straight line are due to the fact that the space must be filled with an even number of grid points in each dimension, which causes small variations in mesh spacing. Note that solving Poisson's equation, using the modified SOR we described, is a small proportion of the total time.

The dashed line shows the $\mathcal{O}(N^{3/2})$ scaling of our Ewald implementation at comparable accuracy. It is estimated that further optimisation of the Ewald code could gain some CPU, up to a factor of 2, but it is clear that from 10^4 atoms the PPPM method is always faster.

Downloaded At: 19:08 14 January 2011

3.2. Parallel Speedup

Parallel implementation on distributed memory architectures of the iterative PPPM method is very straightforward. The usual domain decomposition of the computational box in subdomains can be mapped directly onto the mesh decomposition. Every processor computes the time evolution of a subdomain and the associated part of the mesh. Information from other processors is obtained by communication routines. For message passing

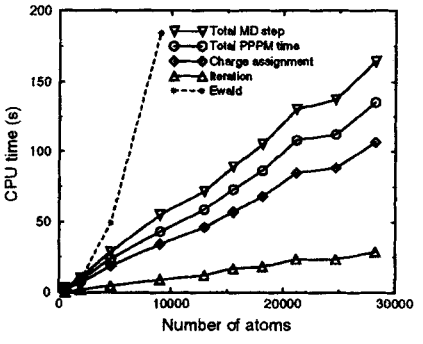


FIGURE 1 CPU time on single i860 processor as a function of number of particles.

implementations it is convenient to use halo's of overlapping grid points, which represent data on neighbouring processors. The halo's act as buffers at stages where data is sent over. The communication for PPPM involves a summation of charge densities after a charge assignment step and a copy of updated electrostatic potential after each half-sweep of the SOR scheme. For both communication steps the halo's need to be only a few grid points wide, which means that only neighbouring processors need to communicate and that the messages are small.

We have tested the PPPM method on a distributed memory linear processor array, DEMOS (Delft Molecular Dynamics Simulator), for configurations from one to eight boards and found close to ideal speedup. Figure 2 is a log-log plot of the CPU time per MD time-step as a function of the number of processors. The straight line gives the slope for ideal speedup.

3.3. Accuracy

The average error in the electrostatic force on a particle is defined as the root mean square deviation from a high accuracy Ewald[1] result: $\Delta f = ((1/N)\sum_i (\Delta f_i)^2)^{1/2}$ [9]. As we suggested, this error depends on both the resolution of the mesh and on the width of the charge assignment function. Figure 3 shows the force errors for a system of amorphous silica with atoms bearing charges -0.8 e and 1.6 e. The rms deviation in Coulomb energy per particle is shown in Figure 4. For both forces and energies the largest errors that occur are roughly 3 times larger in magnitude than the averages. The mesh spacing determines the accuracy of the potential gradients, directly

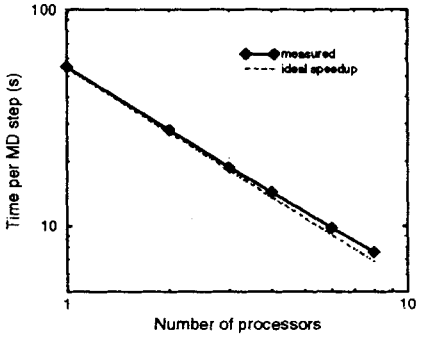


FIGURE 2 CPU time per MD step 1 to 8 i860 processors, system of 9000 charged atoms.

[1] Ewald parameters: $R_c = 6.0$Å, $K_c = 1.8$Å^{-1}, $\alpha_{\text{Ewald}} = 20$, giving $\Delta f < 0.002$ eV/Å.

Downloaded At: 19:08 14 January 2011

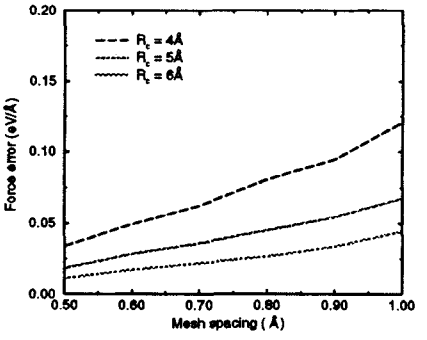


FIGURE 3 Average force accuracy dependence on mesh spacing and charge assignment radius, R_c.

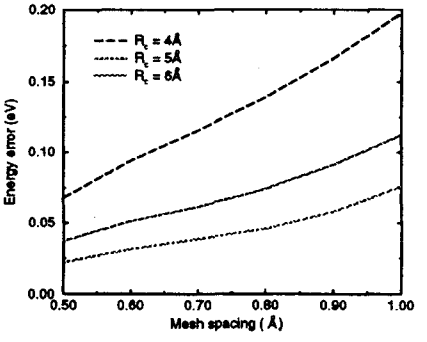


FIGURE 4 Average particle energy accuracy dependence on mesh spacing and charge assignment radius, R_c.

Downloaded At: 19:08 14 January 2011

related to the long range part of the Coulomb forces. At large distances the gradients are small, and a mesh spacing of 1 Å will give sufficient accuracy. The largest errors occur at short distances (< 5Å) where the gradients in charge and potential field are substantial. Increasing the width of the charge assignment smooths the charge distribution and includes more interactions in the direct contribution. As Figures 3 and 4 show, increasing the cut-off radius actually gives a proportionally bigger reduction in the errors than reducing the mesh spacing. It was for this reason that we concentrated earlier on the problem of assigning broad charge distributions. It is also interesting to note the following. The diffusional charge assignment scheme does not explicitly depend on the number of particles. If the number of particles is increased, but the number of grid points kept constant, the long range force algorithm takes essentially the same amount of CPU time. However, as the mesh spacing increases there is a drop in accuracy. At the

same time the cut-off radius also increases and the decrease in accuracy resulting from the increased mesh separation is more than compensated for by the improved accuracy coming from the increase in the cut-off radius. This means that, up until the point where calculating the direct contribution itself becomes significant, the number of particles in the simulation can be increased, and the accuracy remains at worst constant.

Figures 3 and 4 indicate that in the limit of $R_c \uparrow \infty$ and $h_x \downarrow 0$ the errors fall to zero and the method is exact. However, it is also clear that if higher accuracy is required (less than 0.01), then the method will only be competitive for very large systems. With this in mind, we have performed MD simulations to see whether, with this level of error, the effect of the long-range part of the electrostatic potential can still be accurately captured. In a typical simulation we used values $R_c \approx 6$Å and $h_x \approx 1$ Å, and the errors will be substantial. A Δf of 0.05 eV/Å corresponds to about 2% error in electrostatic forces, which is more than generally accepted in MD simulations [16]. In the next section we will examine whether the simulation results are affected by these errors.

Downloaded At: 19:08 14 January 2011

3.4. Simulations

We have tested the PPPM method on several systems of solid and liquid SiO_2 by comparing the results to reference runs using Ewald's method for evaluation of Coulomb interaction.[2]

At interaction cut-off 6.0Å and mesh spacing 1.0Å the PPPM method produced total energies deviating from the Ewald result by more than 46 eV (system of 576 atoms, Coulomb energy = −4300 eV). However, if we look at the *relative* total energies, or the difference in energies between several structures (molten, glass, quartz) we find an accuracy of 2.9 eV, which is less than 0.1% of the total Coulomb energy. This indicates that much of the error in total energy is actually a constant term and thereby irrelevant to most simulation results.

For the influence of errors in electrostatic forces we investigated several functions that are often used as output of MD simulation studies. Radial distribution, velocity autocorrelation, and structure factor were all reproduced well, but a simple accordance between PPPM and Ewald results is not a valid proof for the correctness of the PPPM method, because these functions might not depend strongly on the long range interaction. Indeed a simulation where we neglected the Fourier part of the Ewald sum, to leave

[2] Ewald parameters: $R_c = 6.0$Å, $K_c = 0.4$Å^{-1}, $\alpha_{\text{Ewald}} = 9.12$, giving $\Delta f <$ eV/Å.

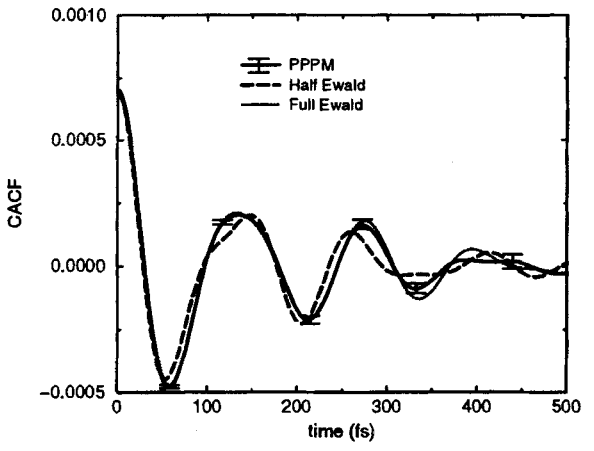


FIGURE 5 Current autocorrelation function [eV^2/fs^2], simulation using PPPM, half Ewald (screened cut-off), and full Ewald method for Coulomb interactions. The error bars on the PPPM curve represent statistical error (2σ, determined from multiple runs) for all curves. A simulation where we used a non-screened cut-off at 6Å resulted in very deviant behaviour, we do not show it here.

Downloaded At: 19:08 14 January 2011

only a screened cut-off potential ($\alpha_{\text{Ewald}} = 5.5$) in real space produced the results equally well.

A function that is more likely to depend on the long range part of the Coulomb interaction is the total current autocorrelation function (CACF) (definition in Appendix). We investigated the CACF for a system of solid silica at 2000 K using a potential proposed by Tsuneyuki [17], where the atomic charges are larger (2.4 e and −1.2 e) than in the Vashishta potential and a three body, term is absent. For this system the full Ewald CACF deviates clearly from the half Ewald result, where only the real space part is included as a screened cut-off at 5Å as shown in Figure 5. Apparently this CACF is very sensitive to the long range part of the Coulomb interaction, which makes it a proper test for our PPPM method.

It was found that the PPPM method with cut-off 6Å and mesh spacing 1Å reproduces the CACF to within statistical error from simulations using full Ewald sum. Although the PPPM errors are relatively large in absolute value, they occur non-systematically and there are no discontinuities in the potential so they should average out for a large part.

4. CONCLUSION

The iterative PPPM method can be made competitive with other well known methods used in MD simulations of Coulombic systems, especially if one

considers its linear scaling with the number of particles and relative ease of coding. For parallel implementations the iterative Poisson solver has a clear advantage over Fourier based methods, especially in environments that have a high cost on global communications. Interprocessor communication for the iterative PPPM is low and can be limited to neighbouring processors, which is very important for special purpose computers like DEMOS. We note however that Fourier based methods do have certain advantages. For instance assigning broad charge distributions in Fourier space is much less problematic. The iterative approach does, however, have the additional advantage of greater flexibility – it is not limited to periodic systems.

Molecular dynamics test simulations of silica using the PPPM method for the Coulomb interactions produced the correct physical behaviour even with an accuracy in forces which was relatively low. This indicates that the PPPM method captures the essential features of the Coluomb interaction, in particular those associated with the long range tail, and that the relatively high level of error has little systematic effect.

Downloaded At: 19:08 14 January 2011

Acknowledgements

This work has been supported by NWO, the Netherlands organisation of scientific research. The research of Dr. Lowe has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences.

References

[1] Brooks, C. L., Pettitt, B. M. and Karplus, M. (1985). Structural and energetic effects of truncated long ranged interactions in ionic and polar fluids. *J. Chem. Phys.*, **83**, 5897–5908.

[2] Perra, L., Essman, U. and Berkowitz, M. L. (1995). Effect of the treatment of long-range forces on the dynamics of ions in aqueous solutions. *J. Chem. Phys.*, **102**, 450–456.

[3] Perram, J. W., Petersen, H. G. and de Leeuw, S. W. (1988). An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Molecular Physics*, **65**, 875–889.

[4] Greengard, L. and Rokhlin, V. (1987). A fast algorithm for particle simulations. *J. Comp. Phys.*, **73**, 325–332.

[5] Petersen, H. G., Solvaeson, D., Perram, J. W. and Smith, E. R. (1994). The very fast multipole method. *J. Chem. Phys.*, **101**, 8870–8876.

[6] Pollock, E. L. and Glosli, J. (1996). Comments on PPPM, FMM and the Ewald method for large periodic Coulombic systems. *Comp. Phys. Comm.*, **95**, 93–110.

[7] Esselink, K. (1995). *Large-scale simulations of many-particle systems*. Ph.D. Thesis, University of Groningen, The Netherlands.

[8] Christiansen, D., Perram, J. W. and Petersen, H. G. (1993). On the fast multipole method for computing the energy of periodic assemblies of charged and dipolar particles. *J. Comp. Phys.*, **107**, 403–405.

[9] Petersen, H. G. (1995). Accuracy and efficiency of the particle mesh Ewald method. *J. Chem. Phys.*, **103**, 3668–3679.

[10] Darden, T., York, D. and Pedersen, L. (1993). The effect of long-range electrostatic interactions of macro-molecular crystals: A comparison of the Ewald and truncated list methods. *J. Chem. Phys.*, **99**, 8345–8348.
[11] Hockney, R. W. and Eastwood, J. W. (1981). *Computer Simulation Using Particles*, McGraw-Hill, New York.
[12] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes, chapter 19*. Cambridge University Press, 2nd edition.
[13] Hautman, J., Halley, J. W. and Rhee, Y.-J. (1989). Molecular dynamics simulation of water between two ideal classical metal walls. *J. Chem. Phys.*, **91**, 467–472.
[14] Luty, B. A. and van Gunsteren, W. F. (1996). Calculating electrostatic interactions using the particle-particle, particle-mesh method with non-periodic long-range interactions. *J. Phys. Chem.*, **100**, 2581–2587.
[15] Vashishta, P., Kalia, R. K. and Ebbjsö, I. (1990). Interaction potential for SiO_2: A molecular dynamics study of structural correlations. *Phys. Rev. B*, **41**, 12197–12209.
[16] Allen, M. P. and Tildesley, D. J. (1987). *Computer simulation of liquids*. Oxford University Press, New York.
[17] Tsuneyuki, S., Tsukada, M., Aoki, H. and Matsui, Y. (1988). First-principles interatomic potential of silica applied to molecular dynamics. *Phys. Rev. Lett.*, **61**, 869–872.

APPENDIX

Molecular Dynamics Parameters

All simulations were done in a constant volume box and temperature was kept around a preset value by soft scaling of particle velocities every 100 time steps. A velocity Verlet time integration scheme was used with a time-step of 1.0 fs. The system consists of 576 atoms in a box of $(21\ \text{Å})^3$, except where indicated otherwise.

Current Autocorrelation Function

The autocorrelation function of the electrostatic current $\boldsymbol{J}(t)$ is defined as [16]:

$$CACF(t) = N^{-2}\langle \boldsymbol{J}(t) \cdot \boldsymbol{J}(0) \rangle \tag{17}$$

$$\boldsymbol{J}(t) = \sum_{i=0}^{N} q_i \boldsymbol{v}_i(t) \tag{18}$$

During the run of 40,000 time-steps the $\boldsymbol{J}(0)$ is computed every 20 steps. The CACF is then computed for 400 stored values of $\boldsymbol{J}(-t)$ and subsequently $\boldsymbol{J}(0)$ is stored itself.

Downloaded At: 19:08 14 January 2011