ELSEVIER

Computer Physics Communications 95 (1996) 73–92

Computer Physics Communications

Ewald summation techniques in perspective: a survey

Abdulnour Y. Toukmaji, John A. Board Jr.

Department of Electrical and Computer Engineering, Duke University, Box 90291, Durham, NC 27708-0291, USA

Received 6 November 1995

Abstract

The simulation of large macromolecular systems has been and remains a challenging problem. There is a general presumption that simulations carried in periodic boundary conditions (PBC) are often the most appropriate to suppress boundary effects. To this end, Ewald summation has been employed to handle long ranged interactions in PBC. There has been a great deal of research targeted at improving the efficiency of Ewald summation, an $\mathcal{O}(N^2)$ algorithm in its traditional formulation, where N is the number of particles in the system. This paper reviews Ewald summation techniques by surveying conventional as well as state of the art efficient methods. Conventional methods, such as tabulation and approximation, are first re-examined along with an $\mathcal{O}(N^{3/2})$ method. Fourier-based approaches which have reduced the complexity to $\mathcal{O}(N\log(N))$ are presented. Multipole expansion techniques, suggested as an alternative to Ewald sums, are reviewed and compared to Fourier methods. The computational efficiency of these new methods facilitates longer, larger and more realistic simulations.

Keywords: Ewald summation; Fast multipole algorithm; Particle–mesh algorithms; Periodic boundary conditions; Molecular dynamics; Algorithms

1. Introduction

One of the challenges facing the molecular dynamics simulation (MD) community is the study of biologically important molecules especially in the presence of a solvent (typically water). Biological systems of interest (e.g., enzymes, proteins, DNA strands, membranes) range in size from a few tens to millions of atoms. When solvent molecules are added, system sizes of interest to MD range from about a thousand atoms on up, with a few tens of thousands of atoms being the largest sizes routinely studied today due to the computational requirements of the simulations. Periodic Boundary Conditions (PBC) have long been employed to minimize surface effects in a variety of calculations [11]. In infinite PBC, the simulation box is infinitely replicated in all directions to form a lattice. In practice, most MD simulations evaluate potentials using some cutoff scheme for computational efficiency. In these cutoff schemes, each particle interacts with the nearest images of the other $N-1$ particles (minimum-image convention), or only with those minimum images contained in a sphere of radius R_{cutoff} centered at the particle. The use of cutoff methods, however, has been shown to introduce significant errors and artificial behavior in a simulation [5,45,54]. To meet the objective of improving the quality and efficiency of MD simulations, it is important to develop algorithms that compute the N-body

PII S0010-4655(96)00016-1

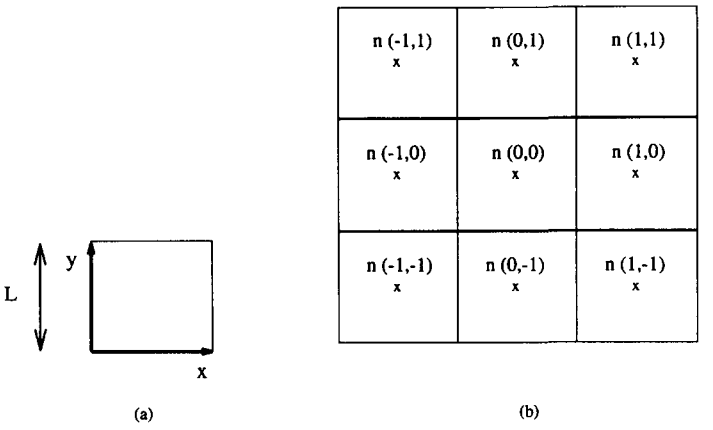

Fig. 1. In a 2D system (a) the unit cell coordinates and (b) a 3 × 3 periodic lattice built from unit cells.

problem for systems with PBC at a cost that is not much greater than that of cutoff schemes but with a better accuracy.

The total Coulomb energy of a system of N particles in a cubic box of size L and their infinite replicas in PBC is given by

$$U = \frac{1}{2}\sum_{\boldsymbol{n}}{}' \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{q_i q_j}{r_{ij,\boldsymbol{n}}}, \tag{1}$$

where q_i is the charge of particle i. The cell-coordinate vector is $\boldsymbol{n} = (n1, n2, n3) = n_1 L\boldsymbol{x} + n_2 L\boldsymbol{y} + n_3 L\boldsymbol{z}$, where $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are the cartesian coordinate unit vectors. The origin cell is located at $\boldsymbol{n} = (0,0,0)$ with image cells located at $L\boldsymbol{n}$ intervals in all three dimensions as $\boldsymbol{n}$ goes to infinity, see Fig. 1. The first sum is primed to indicate that terms with $i = j$ are omitted when $\boldsymbol{n} = \boldsymbol{0}$. The distance between a particle in the origin cell and another at an image cell is $r_{ij,\boldsymbol{n}} = |\, \boldsymbol{r}_{j_{\boldsymbol{n}}} - \boldsymbol{r}_i \,| = |\, \boldsymbol{r}_i - \boldsymbol{r}_j + \boldsymbol{n}L \,|$. The above sum is conditionally convergent, which means that the result depends on the order of summation [3]. Although other shapes are possible, the infinitely periodic system, by convention, is conceptually built in roughly spherical layers for proper convergence.

In most MD simulations, the long-range interactions (Coulomb interactions) are the most time consuming. This paper provides an account of the various methods used to reduce the overhead involved in computing the Coulomb interactions with PBC.

2. Ewald summation

2.1. Introduction

Ewald summation was introduced in 1921 [23] as a technique to sum the long-range interactions between particles and all their infinite periodic images efficiently. For brevity, *Ewald summation* and *Ewald sum* will be used interchangeably. Ewald recast the potential energy of Eq. (1), a single slowly and conditionally convergent series, into the sum of two rapidly converging series plus a constant term,

$$U_{\text{Ewald}} = U^r + U^m + U^o\,. \tag{2}$$

The Ewald sum is therefore written as the sum of these three parts, namely, the real (direct) space sum (U^r), the reciprocal (imaginary, or Fourier) sum (U^m), and the constant term (U^o), known as the self-term,

$$U^r = \frac{1}{2}\sum_{i,j}^{N'}\sum_{\boldsymbol{n}} q_i q_j \frac{\operatorname{erfc}(\alpha r_{ij,\boldsymbol{n}})}{r_{ij,\boldsymbol{n}}}, \tag{3}$$

$$U^m = \frac{1}{2\pi V}\sum_{i,j}^{N} q_i q_j \sum_{\boldsymbol{m}\neq\boldsymbol{0}} \frac{\exp(-(\pi\boldsymbol{m}/\alpha)^2 + 2\pi i\boldsymbol{m}\cdot(\boldsymbol{r}_i - \boldsymbol{r}_j))}{\boldsymbol{m}^2}, \tag{4}$$

$$U^o = \frac{-\alpha}{\sqrt{\pi}}\sum_{i=1}^{N} q_i^2 . \tag{5}$$

V is the volume of the simulation box, $\boldsymbol{m} = (l, j, k)$ is a reciprocal-space vector, and $\boldsymbol{n}$ was defined earlier. The self-term U^o is a correction term that cancels out the interaction of each of the introduced artificial counter-charges with itself as will be explained in Section 2.2. The complimentary error function decreases monotonically as x increases and is defined by $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - (2/\sqrt{\pi})\int_0^x e^{-u^2}\,du$. The theory of Ewald summation is described in more detail by Kittel [33] and Tosi [51].

De Leeuw et al. in [17] and Deem et al. in [16] pointed out that a dipole term that is a function of the medium surrounding the system needs to be added to the above sums. Systems that are surrounded by vacuum experience a dipolar layer on their surface which does not exist in systems surrounded by a conductor [3,16,17]. The dipole term includes the effects of the total dipole moment of the unit cell, the shape of the macroscopic lattice, and the dielectric constant of the surrounding medium.

2.2. A physical perspective

In a charge-neutral system, $\sum_{i=1}^{N} q_i = 0$, the Ewald sum method transforms the potential energy of Eq. (1) into a sum of two rapidly converging series in real and reciprocal space of this form,

$$\sum_{\boldsymbol{n}} \frac{1}{|\boldsymbol{n}|}F(\boldsymbol{n}) + \sum_{\boldsymbol{m}} \frac{1}{|\boldsymbol{m}|}(1 - F(\boldsymbol{m})) . \tag{6}$$

The rapid convergence of the two series stems from the fact that as $\boldsymbol{n} \to \infty$, the function $F(\boldsymbol{n})$ decays rapidly and hence the first series (real-space) converges rapidly. In addition, $(1 - F(\boldsymbol{m}))/|\boldsymbol{m}|$ in the second series (reciprocal-space) is a smooth function and hence its Fourier transform decays rapidly.

A physical interpretation of this decomposition of the lattice sum follows. Each point charge in the system is viewed as being surrounded by a Gaussian charge distribution of equal magnitude and *opposite* sign, Fig. 2, with charge density [3]

$$\rho_i(\boldsymbol{r}) = q_i\alpha^3 \exp(-\alpha^2 r^2)/\sqrt{\pi^3}, \tag{7}$$

where α is a positive parameter that determines the width of the distribution, and $\boldsymbol{r}$ is the position relative to the center of distribution. This introduced charge distribution screens the interaction between neighboring point-charges, effectively limiting them to a short range. Consequently, the sum over all charges and their images in real space converges rapidly. To counteract this induced Gaussian distribution, a second Gaussian charge distribution of the same sign and magnitude as the original distribution is added for each point charge. This time the sum is performed in the reciprocal space using Fourier transforms to solve the resulting Poisson's equation. It is worth pointing out that the choice of a charge distribution, conveniently taken to be Gaussian, is arbitrary. The Ewald sum has been cast with non-Gaussian charge distributions, see Ref. [30].

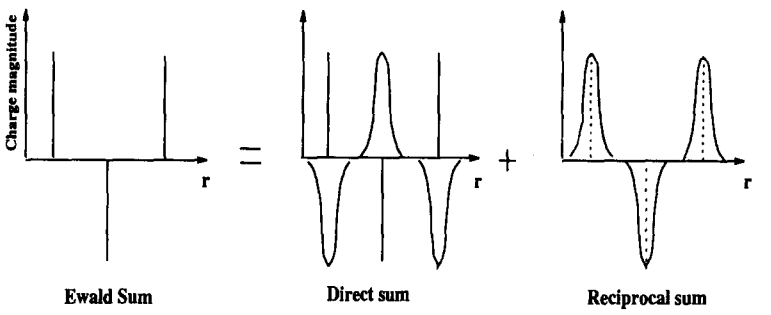

Fig. 2. The Ewald sum components of a one-dimensional point-charge system. The vertical lines are (+/−) unit charges, and the Gaussians are also normalized to unity.

2.3. Force calculation

In molecular dynamics (MD) simulations, force is the quantity needed for the simulation to progress. The force expressions can be easily obtained from the Ewald formulation, Eqs. (3) and (4), by direct differentiation in each coordinate $p = x, y, z$. The resulting equations are

$$F_p^r(i) = q_i \sum_{j=1, j\neq i}^{N} \sum_{\boldsymbol{n}} q_j \frac{(r_{ij,\boldsymbol{n}})_p}{r_{ij,\boldsymbol{n}}^3} \left\{ \operatorname{erfc}(\alpha r_{ij,\boldsymbol{n}}) + \frac{2\alpha}{\sqrt{\pi}} r_{ij,\boldsymbol{n}} \exp(-(\alpha r_{ij,\boldsymbol{n}})^2) \right\}, \tag{8}$$

$$F_p^m(i) = \frac{2q_i}{L} \sum_{j=1, j\neq i}^{N} \sum_{\boldsymbol{m}\neq \boldsymbol{0}} \frac{m_p}{\boldsymbol{m}^2} \exp\left(-\left(\frac{\pi \boldsymbol{m}}{\alpha L} \right)^2 \right) \sin\left(\frac{2\pi}{L} \boldsymbol{m} \cdot \boldsymbol{r}_{ij} \right). \tag{9}$$

The force exerted on particle i is the sum of the direct-space force, $F^r(i)$, and the reciprocal-space force, $F^m(i)$, for all components $p = x, y, z$. Note that the self-term U^o in Eq. (5) is constant and hence does not contribute to the forces in a simulation, however it adds a significant contribution to the potential.

2.4. Choosing Ewald summation parameters

Three parameters control the convergence of the sums in Eq. (2): $n_{\max}$, an integer which defines the range of the real-space sum and controls its maximum number of vectors (i.e. image cells), similarly $m_{\max}$, an integer defining the summation range in the reciprocal-space and its number of vectors, and α, the Ewald convergence parameter, which determines the relative rate of convergence between the real and reciprocal sums. Note that in Eq. (3), a large value of α, i.e. a narrow Gaussian distribution, makes the real-space sum converge faster, that is, as $\alpha \to \infty$, the $\operatorname{erfc}(\alpha x) \to 0$. This means that a small number of $\boldsymbol{n}$-vectors (i.e. $n_{\max}$ small) is sufficient for a rapid convergence. On the other hand, a small α in Eq. (4) causes the reciprocal-space sum to converge faster since as $\alpha \to 0$, the $\exp(-x/\alpha) \to 0$, i.e. a small $m_{\max}$ will suffice.

Traditionally for small systems, α is chosen large enough so that the real-space sum extends no further than the nearest neighbors of the original cell, while a choice of $m_{\max} = 5$ may be sufficient for reasonable convergence of the reciprocal sum. This choice of parameters leads to a reciprocal sum of the order N and a real sum of the order N^2 which is formidable for large systems. Nijbeor [38] showed that for $\alpha = \sqrt{\pi}/L$, the direct and reciprocal-space terms converge at the same rate. However, this is of limited use as emerging methods [14,53] are capable of performing the reciprocal sum more efficiently than the real sum. For large systems, $N > 10^4$, even the minimum-image convention becomes costly (in CPU cycles) and a fixed cutoff radius, $R_{\text{cutoff}} < L/2$, is generally used with a large α. Typical values of R_{cutoff} are between 8–12 Å, e.g. AMBER [12] uses 9 Å.

The choice of Ewald parameters should be based on several considerations:

(i) *System size N*: larger systems may require a larger α and/or R_{cutoff} to limit the number of pair-wise interactions such that the real-space sum converges faster;

(ii) *Accuracy desired*: choosing a larger R_{cutoff}, $n_{\max}$, or $m_{\max}$ will yield more accurate results, however it may be inefficient;

(iii) *CPU time consumed*: larger α means less work done in the real sum, which is traditionally the time consuming part; and

(iv) *Cutoff radius*: the smaller R_{cutoff}, the larger α needs to be for the real space sum to converge rapidly with a reasonable number of $\boldsymbol{n}$-vectors.

In practice, the reciprocal sum is calculated more efficiently than the real sum, hence, α is generally chosen to minimize the real sum and thus dictates the value of $m_{\max}$.

The choice of the Ewald sum parameters is system dependent and is subject to trade-offs between accuracy and speed which in turn is influenced by the algorithmic implementation. Rycerz and Jacobs [42] suggested choosing an α given by this equation, for systems with $N \leqslant 10\,000$,

$$\alpha = \tfrac{1}{2}\left(N^{1/3} + \frac{\beta L}{2R_{\text{cutoff}}}\right), \tag{10}$$

where β is a constant that depends on the system, e.g. 4.0 for Sodium Chloride. Smith [50] suggested using $\alpha \approx 3.5/R_{\text{cutoff}}$ while Perram [39] proposed choosing $\alpha \approx \sqrt{-\ln[\delta]}$ to ensure that the maximum term neglected is $O(\delta)$. The above estimates for α have been obtained by experimenting with the parameters such that the error is minimized. This approach may be feasible for small systems, but for large systems this method is very inefficient. Realizing this optimization problem, Kolafa and Perram [34] introduced simple formulas to estimate α, given a desired error in force (or acceleration) and other Ewald parameters. Other Ewald methods, e.g. Fourier-based methods examined in Section 4, rely on choosing a relatively large α. Since Fourier-based Ewald methods utilize fast Fourier transforms to evaluate the reciprocal sum, it is more efficient to "bias" the Ewald summation towards the reciprocal sum and limit the real-space sum within a small cutoff radius [14]. It should be noted that the potential energy is invariant to the choice of α.

3. Standard Ewald summation methods

3.1. Improved Ewald summation

The Ewald equations presented earlier can be simplified for efficient convergence [43]. One straightforward simplification is to transform the double loop in the reciprocal sum of Eq. (4) into a single sum which is rewritten as follows:

$$U^m = \frac{1}{2\pi V}\sum_{\boldsymbol{m}\neq\boldsymbol{0}} \frac{\exp(-(\pi\boldsymbol{m}/\alpha)^2)}{\boldsymbol{m}^2} \sum_{i,j}^{N} q_i q_j \{\exp(2\pi i \boldsymbol{m}\cdot(\boldsymbol{r}_i - \boldsymbol{r}_j))\}\,. \tag{11}$$

Expanding the term in { } and simplifying yields

$$U^m = \frac{1}{2\pi V}\sum_{\boldsymbol{m}\neq\boldsymbol{0}} \frac{\exp(-(\pi\boldsymbol{m}/\alpha)^2)}{\boldsymbol{m}^2} \sum_{i,j}^{N} q_i q_j \cos(2\pi\boldsymbol{m}\cdot\boldsymbol{r}_{ij})\,. \tag{12}$$

The above equation can be further simplified using a trigonometry identity,

$$U^m = \frac{1}{2\pi V} \sum_{m\neq 0} \frac{\exp(-(\pi m/\alpha)^2)}{m^2} \sum_{i,j}^{N} q_i q_j \{\cos(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_i)\cos(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_j) + \sin(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_i)\sin(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_j)\} \tag{13}$$

$$= \frac{1}{2\pi V} \sum_{m\neq 0} \frac{\exp(-(\pi m/\alpha)^2)}{m^2} \left\{ \left[\sum_{i=1}^{N} q_i \cos(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_i) \right]^2 + \left[\sum_{i=1}^{N} q_i \sin(2\pi \boldsymbol{m}\cdot \boldsymbol{r}_i) \right]^2 \right\}, \tag{14}$$

and finally by defining

$$S(\boldsymbol{m}) = \sum_{k=1}^{N} q_k \exp(2\pi i \boldsymbol{m}\cdot \boldsymbol{r}_k), \tag{15}$$

known by protein crystallographers as the "structure factor" [33], we obtain the following:

$$U^m = \frac{1}{2\pi V} \sum_{m\neq 0} \frac{\exp(-(\pi m/\alpha)^2)}{m^2} S(\boldsymbol{m})S(-\boldsymbol{m}) = \frac{1}{2\pi V} \sum_{m\neq 0} \frac{\exp(-(\pi m/\alpha)^2)}{m^2} |\, S(\boldsymbol{m})\,|^2 . \tag{16}$$

Therefore, the above manipulation has transformed the double sum over i and j of order N^2, Eq. (11), into two single sums of order N, Eq. (14), and finally into a single sum, Eq. (16), that is much more efficient.

3.2. Truncation schemes

In the context of Ewald sum, truncation is often used to reduce the cost of the real-space sum. Traditionally, two types of truncations have been used to simplify the field evaluations and limit the long-range interactions in PBC, namely the minimum image convention and spherical cutoffs, see Fig. 3. In the minimum image scheme, each particle interacts with exactly $N-1$ particles that are inside a fictitious box of size L centered on the particle. The total number of interactions is therefore $\frac{1}{2}N(N-1)$, which is prohibitive for large systems. A further simplification is achieved when a spherical cutoff radius is used. In this scheme, a sphere of radius R_{cutoff}, typically between 8–12 Å, is centered at the particle of interest and particles outside the sphere are excluded from interacting with the particle. The total number of interactions is a function of the cutoff radius and is an $\mathcal{O}(N)$ operation for $R_{\text{cutoff}} < L/2$. Fig. 3 illustrates both methods. In the spherical cutoff method, particle 5 in the original cell O will only interact with particles $1A, 2G, 4A$ while in the minimum image scheme, particle 5 in the original cell interacts with particles $1A, 2G, 3O, 4A$.

3.3. Neglecting the reciprocal-space

To speedup an MD simulation, Rycerz and Jacobs [42] suggested that by properly choosing the simulation parameters, the reciprocal-space sum contribution to the total energy can be neglected entirely. For MD simulations of Bi_2O_3, one with $N = 270$ and another with $N = 2160$, $\alpha = 5.6/L$, and using the minimum image convention, the authors estimated the contribution of reciprocal potential (U_m) to total potential (U) to be about 1 : 1500. Similar observations were made for a crystalline and molten NaCl system.

It was pointed out [42], however, that this approach should not be applied to small systems as the energy becomes strongly dependent on the system's configuration. Moreover, there is no indication that other ionic systems will have a similar small U_m contribution to the energy. It is also unclear whether the method can be extended to polar systems. Therefore, this method requires more investigation and is currently limited to large ionic systems and hence is not recommended for general MD simulations.

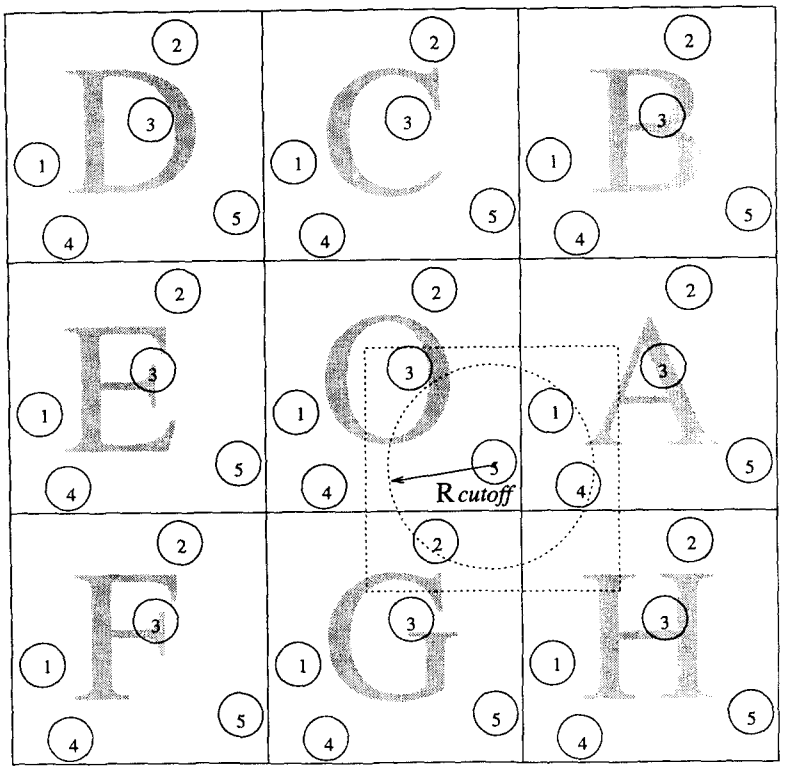

Fig. 3. Periodic boundary conditions: a 2D periodic lattice where the unit cell contains five particles (adapted from [3]).

3.4. Tabulation methods

Tabulation methods aim at reducing the time spent in performing the Ewald sum by tabulating the energy and the three components of forces as a function of the distance vector $\boldsymbol{r}_{ij}$. A table is constructed as a function of the distance between a pair of particles, one of which is at the center of the original unit cell while the other spans a grid. Interpolation is used to approximate untabulated force/potential components. To reduce the run time, the table can be computed off-line. For this approach to be feasible, the interpolation method used must be simple and the table has to be "moderately" fine for it to be sufficiently accurate. Therefore, a trade-off exists between the interpolation scheme and grid discretization used, and the resultant efficiency and accuracy.

Sangester et al. [43] used simple 3D linear interpolation. In their method, the vectors $\boldsymbol{r}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ were tabulated with components satisfying $\frac{1}{2}L \geqslant x_{ij} \geqslant y_{ij} \geqslant z_{ij} \geqslant 0$ and thus making use of the symmetry of the cubic cell to realize savings in storage. Sangester's tabulation method [43] initially calculates and stores the energy and force components at a particle i, centered at the basic cell, and particle j at a mesh point defined by the segment above. All nearest-image particles, however, are excluded from the tabulation and are calculated on-line for accuracy. To obtain the tabulated force between particle i and all j's images (except the nearest image), a transformation must be first applied to find the corresponding nearest-image vector $\boldsymbol{r}^*_{ij}$ which is followed by interpolation and backward transformation to yield the interpolated force. The total inter-particle force felt by particle i due to particles $\{j\}$ is therefore the sum of the contribution from all image particles (tabulated) and the nearest particle (on-line). The above process is repeated for all particles j.

Compared to the standard Ewald sum, this approach accelerates Ewald's computation; however, due to the direct force evaluation, the method is still $\mathcal{O}(N^2)$. In addition, interpolation and forward/backward transformation may end up having a large runtime overhead.

Alternatively, one can tabulate only the direct-space sum, provided that the reciprocal-space sum can be computed efficiently. As discussed earlier, the direct-space sum, an $\mathcal{O}(N^2)$ operation, is reducible to $\mathcal{O}(N)$ with the appropriate choice of Ewald parameters. For example, the choice of α can bias the intensity of calculations in either space at the expense of the other. Therefore, depending on the Ewald parameters used, tabulation may benefit the overall runtime. Belhadj et al. [6] reported a speedup of four when look-up tables were incorporated in their water simulations.

Another approach can tabulate only the exponent, exp( ), and/or the complimentary error function, erfc( ), if they are more expensive to evaluate explicitly on a particular computer. Our own tests show a speedup of about 50% when the table look-up is used for both erfc( ) and exp( ). Moreover, interpolation schemes that compute and store an estimate of the first derivative such as spline interpolation [24], offer a good approximation for the tabulated function and its derivative. For example, $\exp(-x^2)$, in Eq. (8), is obtained for free when $\operatorname{erfc}(x)$ is tabulated. This is based on the observation that $(d/dx)\operatorname{erfc}(x) = (-2/\sqrt{\pi})\exp(-x^2)$.

In summary, tabulation methods attempt to strike a balance between accuracy versus speed, a trade-off that is dependent on the size of the system and the objective of the simulation. These methods can produce a major impact in speeding up MD/MC applications where medium accuracy is acceptable. Higher accuracy simulations with tabulation may suffer in performance due to low resolution in coarse tabulation and/or expensive interpolation, or run out of storage for high-resolution tables.

3.5. Polynomial approximation methods

This approach attempts to improve the accuracy and speed of the Ewald sum calculation by using a polynomial approximation rather than tabulation. The idea is to find a polynomial approximation to the Ewald potential that is cheaper to evaluate and then differentiate it analytically to get the forces. There exist several polynomials to approximate the Ewald sum ranging from the simple method of Brush et al. [13] to the more accurate cubic harmonics of Von der Lage and Bethe [52].

Hansen's work [28] for Monte Carlo simulations used an anisotropic approximation where $\phi(\boldsymbol{r})$ was split into an isotropic part and a cubically symmetric term while utilizing exact cubic harmonics for better convergence. The error reported in Hansen's method for the total potential energy was less than 0.1%. Similar work was done by Slaterry et al. [48], but the solution was expressed as a sum of Poisson's equation with cubic symmetry.

Adams and Dubey [2] offer a variety of approximations for charge–charge, charge–dipole, and dipole–dipole systems. In one accurate approximation, the Ewald potential $\phi(\boldsymbol{r})$ was expanded in powers of r and then the expansion was approximated with functions of cubic symmetry to obtain the coefficients. The resulting sum expansion has this form:

$$\phi_l(\boldsymbol{r}) = \frac{1}{r} + S + A_2 r^2 + \sum_{n=4,6}^{l} (A_n K h_n(\boldsymbol{r}) + B_n K h_n'(\boldsymbol{r})) , \tag{17}$$

where $4 \leqslant l \leqslant 20$, l is even; B_n is zero except for $n \geqslant 12$; and S is the "self term". The A_n and B_n coefficients are tabulated in [2]. Adams and Dubey report an RMS error of 4×10^{-3} for $l = 6$, and 2.5×10^{-5} with $l = 14$. Our experimentation showed that the above approximations are expensive for moderate to high accuracy calculations. Alternatively, one can use a polynomial approximation to the erfc() function that is faster to evaluate if tabulation is expensive, e.g. see [1]. However, for high accuracy computations or long time simulations, such approximations will perform poorly as errors accumulate. In summary, approximations methods, analogous to tabulation, offer a fast alternative to Ewald simulations with limited accuracy.

3.6. An $\mathcal{O}(N^{3/2})$ algorithm

The work of Perram et al. [39] was the first recognized method that reduced the Ewald sum complexity without using approximation schemes. In this method, Perram et al. utilized the linked-cell spatial decomposition technique devised by Hockney and Eastwood [31]. It was shown that by properly dividing the potential evaluation into a short-range interaction, which ordinarily grows as N^2, and a long-range interaction, that ordinarily grows as N, the Ewald sum can be performed with an overhead that grows as $N^{3/2}$. The argument is as follows.

Assume that the simulation box is spatially decomposed into $m \times m \times m$ sub-boxes. Moreover, let α be chosen such that the maximum term neglected in the real sum is $O(\delta)$, where δ is some relative error constant. Hence, α is chosen such that $\alpha \approx m\sqrt{-\ln(\delta)}$. Further, assume that t_r and t_f are the unit times to perform a unit force computation in the direct (real) and reciprocal (Fourier) space, respectively. Let T_r denote the time (on average) to compute the real-space sum and T_f denote the time (on average) to evaluate the reciprocal-space sum. If α is chosen large enough such that particle-interactions beyond one sub-box are neglected, i.e. only interactions between particles within the nearest neighbor sub-boxes are considered, then T_r is proportional to the number of boxes in the system times the average number of particles in each box and can be approximated by

$$T_r = c_1 t_r M \left(\frac{N}{M}\right)^2 = c_1 \frac{N^2}{M} t_r \,, \tag{18}$$

where c_1 is constant and $M = m^3$. Furthermore, from Eq. (16), T_f is proportional to the number of particles times the maximum number of reciprocal vectors K^3, where K is an integer that controls the range of reciprocal vectors, and is estimated by

$$T_f = c_2 K^3 N t_f = c_3 m^3 N t_f = c_3 M N t_f \,. \tag{19}$$

Note that $K \propto m$ and c_2 and c_3 are constants that depend on Ewald parameters and the system dimensions. The total time, therefore, for Ewald computation is $T = T_r + T_f$. To minimize T, we treat M as a variable and evaluate $(dT/dM) = 0$. Carrying out the derivative and solving for M results in $M_{\text{opt}} = c_4 \sqrt{t_r/t_f} N^{1/2}$. Substituting in the above expression for T yields

$$T = c_5 f(t_r, t_f) N^{3/2} = \mathcal{O}(N^{3/2}) \,, \tag{20}$$

where c_4, c_5 are also constants. The above equation shows that for M_{opt}, the overhead in computing the Ewald sum grows as $N^{3/2}$. Fincham [25] offers another derivation to prove the above result.

In summary, the previous subsections offered an account of the traditional approaches to Ewald summation. Among these methods, Perram's $\mathcal{O}(N^{3/2})$ algorithm has probably the most useful combination of speed and accuracy. Rycerz and Jacobs' approach is the least useful because it is system-dependent and requires extra experimentation to validate its use. Most MD "production" codes that perform Ewald summation use one (or more) of the approximation techniques discussed in Section 3 to enhance their performance.

4. Fourier-based Ewald summation methods

In this section we review methods that have made use of Fast Fourier Transform (FFT) techniques with the charges interpolated to a 3D grid. These methods reduce the complexity of Ewald summation to $\mathcal{O}(N \log(N))$ by computing the traditional reciprocal sum over reciprocal vectors more efficiently using FFT.

4.1. Particle–Particle Particle–Mesh

Luty et al. [37] and Rajagopal et al. [40] offered an alternative approach to the Ewald sum by extending the Particle–Particle Particle–Mesh method (**PPPM**) developed by Hockney and Eastwood [31]. The method relies on expressing the long-range inter-particle force as the sum of two components: the short-range force, which is only nonzero within some cutoff radius, and the "reference" force, that is long-ranged and smooth and can be approximated on a grid. The analogy between PPPM and the Ewald sum is clear. In the traditional Ewald sum, the direct sum, due to the point-charge and Gaussian counter-distributions, is also short-ranged; while the reciprocal sum, due to the Gaussian distributions, is a smooth function and its Fourier transform converges rapidly.

The authors [37] used PPPM's standard charge distribution, a sphere with a uniform decreasing density (the S2 function), rather than the Gaussian distribution used for the Ewald sum. The S2 distribution is given by

$$\lambda(r) = \begin{cases} \dfrac{48}{\pi a^4}\left(\frac{1}{2}a - r\right), & r < \frac{1}{2}a, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where a is a parameter that adjusts the S2 distribution.

The short-range potential between two particles, each with an S2 charge distribution, is evaluated with a cutoff radius, $r_{\text{cutoff}} \simeq 0.7a$, given in terms of $\zeta_{ij} = 2r_{ij}/a$ by

$$\psi(\zeta_{ij})_{\text{short}} = \frac{1}{4\pi\epsilon_0}\left(\frac{1}{r_{ij}} - \frac{1}{70a}\sum_{n=-1}^{7} C_n \zeta_{ij}^n\right), \quad 0 \leqslant \zeta_{ij} < 2, \tag{22}$$

where

$$C_{-1,\ldots,7} = \begin{cases} (0, 208, 0, -112, 0, 56, -14, -8, 3), & 0 \leqslant \zeta_{ij} \leqslant 1, \\ (12, 128, 224, -448, 280, -56, -14, 8, -1), & 1 < \zeta_{ij} \leqslant 2. \end{cases}$$

The long-range potential is evaluated in Fourier space using

$$\hat{\psi}_{\text{long}}(\boldsymbol{k}) = \hat{\rho}(\boldsymbol{k})\hat{G}(\boldsymbol{k}), \tag{23}$$

where " $\hat{}$ " indicates the Fourier transform of a function. The influence function is usually given by $\hat{G}(\boldsymbol{k}) = \hat{\lambda}(\boldsymbol{k})/\epsilon_0 k^2$ but can be optimized depending on the system size, charge shape function, and the interpolation function. The long-range potential is computed using the following steps, see Fig. 4:

(i) Assign charge to a 3D grid that fills the simulation space. This step yields the charge distribution ρ which is a function of both the charge distribution λ and assignment functions. Several charge assignments can be used depending on the accuracy desired, e.g. triangle-shaped charge function (TSC). However, for a charge assignment function to be feasible, it has to cover a relatively small number of grid points. Moreover, the assignment should vary smoothly with the particle's location, which requires adequate grid spacing and in turn determines the runtime cost.

(ii) Using Fourier transform over the grid, obtain $\hat{\rho}$, and calculate $\hat{\psi}_{\text{long}}(\boldsymbol{k})$ using Eq. (23). Apply the inverse Fourier Transform to obtain $\psi_{\text{long}}(\boldsymbol{r})$ evaluated at the grid points.

(iii) Obtain the grid-defined electrostatic forces by differentiating the potentials numerically. Several numerical differencing techniques can be used such as 4-point central differencing.

(iv) Interpolate the electrostatic fields (potentials) from the grid to particle locations using the same function of step 1.

Fast Fourier Transforms are used in PPPM methods resulting in an $\mathcal{O}(N\log(N))$ algorithm. It should be noted that in using this approach, the influence function $\hat{G}(k)$ is system-specific and hence for each new system,

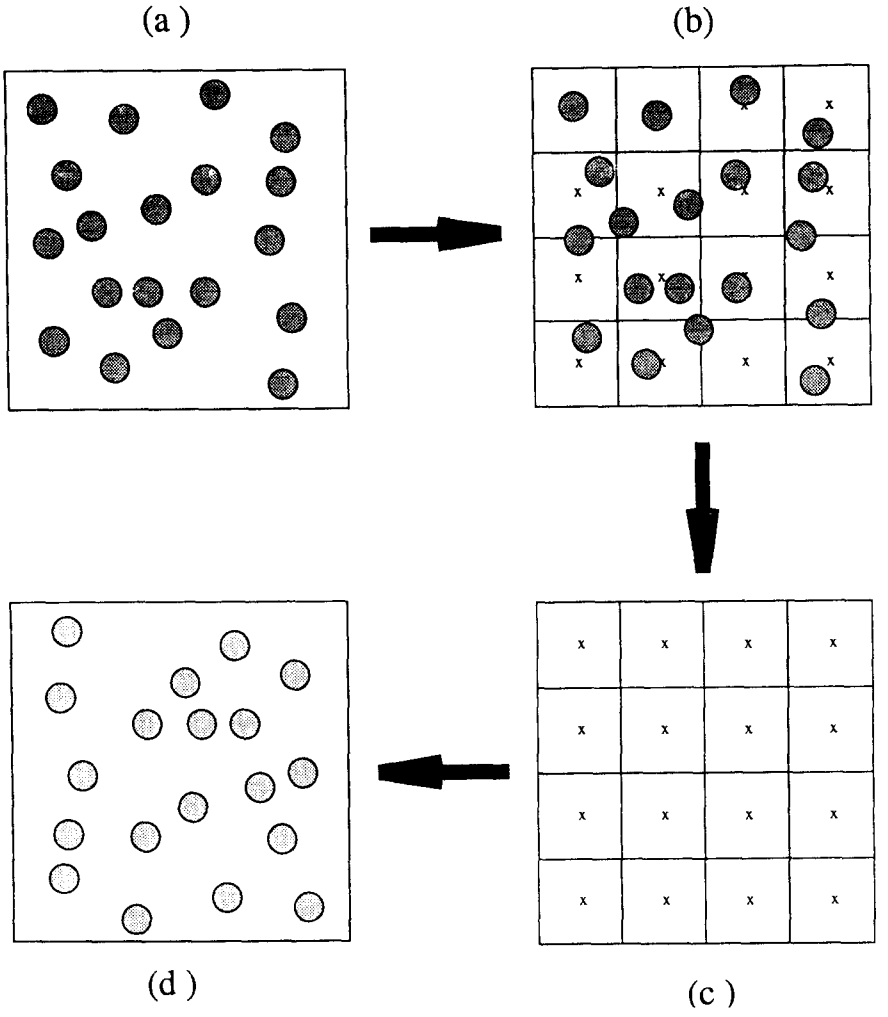

Fig. 4. A 2D schematic of particle–mesh technique used in most Fourier-based methods. (a) A system of charged particles. (b) The charges are interpolated on a 2D grid. (c) Using FFT, the potential and forces are calculated at grid points. (d) Interpolate forces back to particles and update coordinates.

and depending on system parameters, e.g. size or charge shape, a new optimal influence function has to be computed resulting in some loss of generalization. In addition, the PPPM implementation of [37] implies that in order to increase the accuracy of potential/force computations, one needs to either refine the mesh, or use a better weighing/interpolation scheme; both choices can be computationally expensive. In addition, since the electrostatic force experienced at a grid point is obtained by numerically differentiating the potential, another source of error is introduced to the results. In order to improve the accuracy and reduce the error in the above steps, higher-order differencing schemes have to be incorporated. There is a rich interaction between all of the above choices (i.e. weighing/interpolation, differentiation, charge shape, influence function, etc.) and the user has to experiment with the system of interest in order to utilize this algorithm efficiently.

4.2. Particle–Mesh Ewald

The Particle–Mesh Ewald method (**PME**) [14] is also inspired by Hockney and Eastwood's particle–particle particle–mesh method (PPPM) [31,20]. Unlike PPPM, PME divides the potential energy into Ewald's standard direct and reciprocal sums and uses the conventional Gaussian charge distributions. The direct sum, Eq. (3), is evaluated explicitly using cutoffs while the reciprocal sum, Eq. (4), is approximated using FFT with convolutions on a grid where charges are interpolated to the grid points. In addition, in contrast to particle–mesh methods, PME does not interpolate but rather evaluates the forces by analytically differentiating the energies, thus reducing memory requirements substantially.

This method is reported to be highly efficient incurring only 30–40% overhead over conventional truncated list-based (i.e. non-Ewald) methods at a relative force accuracy around 10^{-4}. PME is also capable of achieving higher accuracy ($\approx 10^{-6}$ relative force error) with relatively little increase in computational cost.

In computing the direct sum, the Ewald parameter α is chosen large enough so that a fixed cutoff radius can be applied thus reducing the complexity of the direct sum from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. To compensate for the truncation in evaluating the direct sum, the number of reciprocal vectors is increased proportionally to N to bound errors.

The reciprocal sum is computed using 3D-FFT with an overhead that grows as $N\log(N)$. PME is therefore an $\mathcal{O}(N\log(N))$ method. The reciprocal sum, Eq. (4), is given by

$$E_{\text{recip}} = \frac{1}{2\pi V}\sum_{\boldsymbol{m}\neq\boldsymbol{0}} \frac{\exp(-(\pi \boldsymbol{m}/\alpha)^2)}{\boldsymbol{m}^2} S(\boldsymbol{m})S(-\boldsymbol{m}) , \tag{24}$$

where $S(\boldsymbol{m})$ is defined in Eq. (15). The structure factor can be approximated by

$$\begin{aligned} S(\boldsymbol{m}) \simeq \tilde{S}(\boldsymbol{m}) &= \sum_{k_1,k_2,k_3} Q(k_1,k_2,k_3)\exp\left(2\pi \mathrm{i}\left(\frac{m_1 k_1}{K_1}+\frac{m_2 k_2}{K_2}+\frac{m_3 k_3}{K_3}\right)\right) \\ &= \mathcal{F}(Q)(m_1,m_2,m_3)) , \end{aligned} \tag{25}$$

where $\mathcal{F}(Q)$ is the 3D FFT of Q, the charge matrix. The Q matrix is a three-dimensional matrix that is obtained by interpolating the point charges to a uniform grid of dimensions $K_1 \times K_2 \times K_3$ that fills the simulation cell. By combining Eq. (25) with Eq. (24), the reciprocal energy can be also approximated by

$$\tilde{E}_{\text{recip}} = \frac{1}{2\pi V}\sum_{\boldsymbol{m}\neq\boldsymbol{0}} \frac{\exp(-(\pi \boldsymbol{m}/\alpha)^2)}{\boldsymbol{m}^2} \mathcal{F}(Q)(m)\mathcal{F}(Q)(-m) . \tag{26}$$

The above equation is rewritten, after some manipulation, as a convolution,

$$\tilde{E}_{\text{recip}} = \tfrac{1}{2}\sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1} Q(m_1,m_2,m_3)(\psi_{\text{rec}} * Q)(m_1,m_2,m_3) , \tag{27}$$

where ψ_{rec} is the reciprocal pair potential and "$*$" indicates a convolution. To evaluate the reciprocal sum, the Q matrix is first computed over a 3D uniform grid and then transformed using inverse 3D FFT to obtain the structure factors. The reciprocal energy is then calculated using Eq. (27) with the aid of FFT to compute the convolution $\psi_{\text{rec}} * Q$.

The charge interpolation function used originally in PME was Lagrange interpolation [14]. However, an enhanced PME [15] utilizes the B-spline interpolation function, which is smoother and allows higher accuracy by simply increasing the order of interpolation. The smoothness of B-spline interpolation allows the force expressions to be evaluated analytically, with high accuracy, by differentiating the real and reciprocal energy equations rather than using finite differencing techniques.

4.3. Fast Fourier Poisson Method

The Fast Fourier Poisson method (**FFP**) [53], recasts Ewald summation in yet another form that is also evaluated, as in PME, using FFT in $\mathcal{O}(N\log(N))$. The method claims to achieve high accuracy without using interpolation [14] or multipole expansion [44] schemes. The main difference is in the implementation of the reciprocal space sum, ϕ_{recip}. Unlike PME, FFP does not interpolate the solution of the reciprocal sum; rather, FFP samples the Gaussian sources associated with each point charge on a grid and solves for the potential.

In FFP, the reciprocal energy is written as follows:

$$E_{\text{recip}} = \tfrac{1}{2}\sum_{i=1}^{N} q_i \phi_{\text{recip}}(\boldsymbol{r}_i) = \tfrac{1}{2}\int \rho(\boldsymbol{r}')\phi_{\text{recip}}(\boldsymbol{r}')d^3\boldsymbol{r}', \tag{28}$$

where ϕ_{recip} is the reciprocal potential associated with the Gaussian distributions of the artificial screening charge distribution $\rho_s(\boldsymbol{r})$, and $\rho(\boldsymbol{r}) = \sum_{i=1}^{N} q_i\delta(\boldsymbol{r}-\boldsymbol{r}_i)$ is the point charge density, see Section 2.2. Initially, ϕ_{recip} and its gradient are determined over the grid's points using FFT as a solution to Poisson's equation, $\nabla^2\phi_{\text{recip}}(\boldsymbol{r}) = 4\pi\rho_s(\boldsymbol{r})$. The total energy, E, is reformulated so that there is no need to evaluate the reciprocal potential at particles' locations. This can be accomplished by replacing the interaction of each point charge with ϕ_{recip} by the interaction of the introduced charge density having the same net charge at the same location. This is achieved by splitting ϕ_{recip} of Eq. (28) into two integrals,

$$E_{\text{recip}} = \tfrac{1}{2}\int [\rho(\boldsymbol{r}') + \rho_s(\boldsymbol{r}')]\phi_{\text{recip}}(\boldsymbol{r}')d^3\boldsymbol{r}' - \tfrac{1}{2}\int \rho_s(\boldsymbol{r}')\phi_{\text{recip}}(\boldsymbol{r}')d^3\boldsymbol{r}'. \tag{29}$$

The first integral in Eq. (29) is canceled out by the real-space sum leaving the second integral for evaluation only.

The FFP method is not restricted to orthogonal unit cells and has the advantage of the energy and gradients being continuous functions of the point charge position. However, a careful inspection of the timings presented in this paper [53] shows that for moderate accuracy, i.e. 10^{-4} relative force error, and a system size of $N = 5768$ particles, the runtime is about 3 times more expensive than a conventional 9 Åcutoff method. This suggests that the implementation of this method may need to be optimized further.

In summary, this section has provided an account of various methods that perform the Ewald sum using efficient FFT with a computational complexity of $\mathcal{O}(N\log N)$. In the methods discussed above, both PME and Luty's PPPM [37] methods are highly efficient. It is not clear, however, whether PPPM can easily achieve the high accuracy levels attainable by PME, while FFP can benefit from a more efficient implementation. Table 1 cites simulation results of these methods for comparison.

5. Multipole-based Ewald summation methods

5.1. The Fast Multipole Algorithm

The Fast Multipole Algorithm (**FMA**) of Greengard and Rokhlin [26,27] has been successfully used for efficiently computing the N-body problem for a single (non-periodic) cell. The main feature of the FMA is that it offers a solution to the traditional $\mathcal{O}(N^2)$ N-body problem that is linear in N in many cases (and never worse than $(\mathcal{O}(N\log N))$ while maintaining known accuracy bounded by rigorously derived error bounds. Many applications have capitalized on the FMA's performance, especially in molecular dynamics and celestial mechanics.

Table 1
This table lists the relative performance of Fourier- and Multipole-based Ewald methods as reported in the literature

Method	Computer	CPU time [s]	No. of particles	Error	Comparison	Simulation conditions
PPPM [37]	IBM RISC 6000	7.18	5024	Rel-Force = 0.8×10^{-3}	$\frac{\text{Ewald1}}{\text{PPPM1}} = 1.6$	N-grid(no. of grid points) = 48 h-space(grid spacing) = 1.164 Å
	Remarks: SUN Sparc2 times available; add 64.66 s pair-list time					
PME [14,15]	SGI Indigo R4400	30	20000	Rel-Force = 5×10^{-4}	$\frac{\text{Std.Ewald}}{\text{PME}} = 59.3$	9 Å cutoff direct sum includes Van der Waals interactions
	Remarks: FORTRAN code; TIP3P water system					
FFP [53]	SGI Indigo R4000	3.2	648	Rel-Pot = 2.2×10^{-4} Rel-Force = 3.8×10^{-4}	$\frac{\text{Residu}-9\ \text{Å}-\text{cutoff}}{\text{FFP}} = 0.33$ for 1936 particles	grid-spacing: 1.17 Å(648 particles) 1.25 Å(1936)
		29.1	1936	Rel-Pot = 4.2×10^{-4} Rel-Force = 1.6×10^{-5}		
	Remarks: FORTRAN code; TIP3P water system and NaCl					
FMA [44]	CRAY Y-MP 1 processor	16.77	4000	Rel-Pot = 0.28×10^{-3} Rel-Force = 0.44×10^{-3}	$\frac{\text{Std.Ewald}}{\text{FMA}} = 6.51$ (4000 particles)	l(order of expansion) = 8 R(level of refinement) = 3
		59.79	20000	Rel-Pot = 0.63×10^{-3} Rel-Force = 0.73×10^{-3}		
	Remarks: FORTRAN code; system of random charge distribution					
RCMM [19]	SGI 380 1 processor	23.6	4816	Abs-Pot = 1.85 Abs-Force = 0.106	$\frac{\text{min}-\text{image}}{\text{RCMM3}} = 10.7$	multiple expansion up to 5th order Taylor expansion up to 3rd order
	Remarks: Polyethelene crystal system					
MPE [46,47]	NEC SX-2	2.51	23531	Abs-Pot = 0.64 Abs-Force = 0.21	$\frac{\text{GR}-\text{FMA}}{\text{MPE}} = 1.51$	m(grid points) = 8 n_B(controls neighbor interaction) = 1
	Remarks: Using VECTOR processing; BPTI-water system					
MMM [36]	HP 735/125	48	20000	Rel-Force = 3×10^{-4}	N/A	p(multiple expansion terms) = 8 k(control no. image cells) = 5
	Remarks: N = 1000–100000 times available					

Note:
Abs-Pot is the absolute error in potential measured in kcal/mol;
Abs-Force is the absolute error in force measured in kcal/mol·Å;
Rel-Pot is the relative error in potential;
Rel-Force is the relative error in force.

The basic idea behind FMA is simple. The force (potential) exerted on a particle due to all the pair-wise interactions can be divided into two components: one due to nearby particles that can be computed directly and one due to the distant particles approximated by their multipole expansions.

Given a collection of point charges $q_i : i = 1, \ldots, k$, Fig. 5, enclosed by a sphere of radius a whose center is a distance r away such that $r > a$, the potential $\Phi(r)$ at P due to all the well-separated particles q_i is given by the infinite multipole expansion

$$\Phi(\boldsymbol{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{M_l^m}{r^{l+1}} Y_l^m(\theta, \phi) , \tag{30}$$

where $Y_l^m(\theta, \phi)$ is the spherical harmonic polynomial [32], and

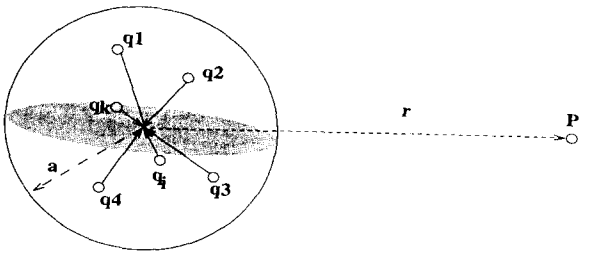

Fig. 5. The multipole expansion principle.

$$M_l^m = \sum_{i=1}^{k} q_i \rho_i^l Y_l^{*m}(\alpha_i, \beta_i) \,. \tag{31}$$

In a system of charged particles, the FMA starts by using hierarchical spatial decomposition (typically an oct-tree) to divide the simulation cell into smaller sub-cells. A truncated multipole expansion is then calculated for each sub-cell at the finest refinement level which expresses the effect of all particles in that sub-cell on distant particles. These expansions are combined in a hierarchical fashion to represent the effects of larger and larger groups of particles in what is known as the upward pass. The far-field potential at the center of a sub-cell due to particles of another sub-cell that is sufficiently far apart is computed via a Taylor (local) expansion that utilizes the multipole expansion of the distant sub-cell. As distant sub-cells interact, each sub-cell accumulates the far-field effects into a single local Taylor expansion. In the downward pass, the local expansions are used to transfer the field effect from parent to children sub-cells in a non-redundant procedure. Finally, far-field effects are combined with direct short-field evaluations to yield the potential at each particle. For more details, the reader is referred to [26,27] and [9,10] for FMA's 3D parallel implementation. It is worth noting that any other converging series approximation of the potential due to the far cells can be used in place of multipoles. Anderson [4], for example, used a "ring" ("sphere" in 3D) approximation whereby the induced potential of a group of charged particles is approximated over certain locations on the circumference of a surrounding ring.

In the FMA, there are several translations used to facilitate computing the electrostatics. To calculate the multipole expansion at the lowest level of spatial decomposition, Eqs. (30) and (31) are used which represent the aggregate potential due to the k particles. Similar multipole expansions are carried out for all cells at all levels using the Multipole to Multipole translational property, as shown in Fig. 6. Cell-to-cell interactions are carried out between well-separated cells using Multipole to Local translations. In the final step, each parent cell passes down its accumulative far-field interaction to each of its children using the Local to Local translation.

We have so far introduced FMA for the simulation of isolated systems, i.e. free boundary conditions. However, the fast multipole method has been extended for use with periodic boundary conditions. Schmidt and Lee [44] have developed a method to simulate point-charge systems with infinite PBC in three dimensions in a procedure that uses both FMA and Ewald techniques. Given a charge-neutral system, all multipole and local expansions at all refinement levels of the basic unit cell are calculated and the potential due to all particles in the unit cell is obtained. A multipole expansion approximation is now available for the original unit cell as well as the image cells. This is based on the fact that all image cells have the same multipole expansion as the basic unit cell with respect to their centers. The local expansions of all periodic image cells, except the nearest neighbors (26 cells), are next used to evaluate the local expansion at the basic unit cell. However, to obtain the local expansion of the image cells, the multipole to local translation is expressed using an Ewald sum formulation. The algorithm next proceeds with its downward pass and terminates by taking the sum over all image cells using a recursive formulation of the Ewald sum.

In their paper, Schmidt and Lee [44] presented a timing and accuracy comparison between their FMA implementation of PBC and Ewald summation. It is not clear however, where the break even point is, i.e. the number of particles at which both methods are equally fast, as both implementations have not been optimized

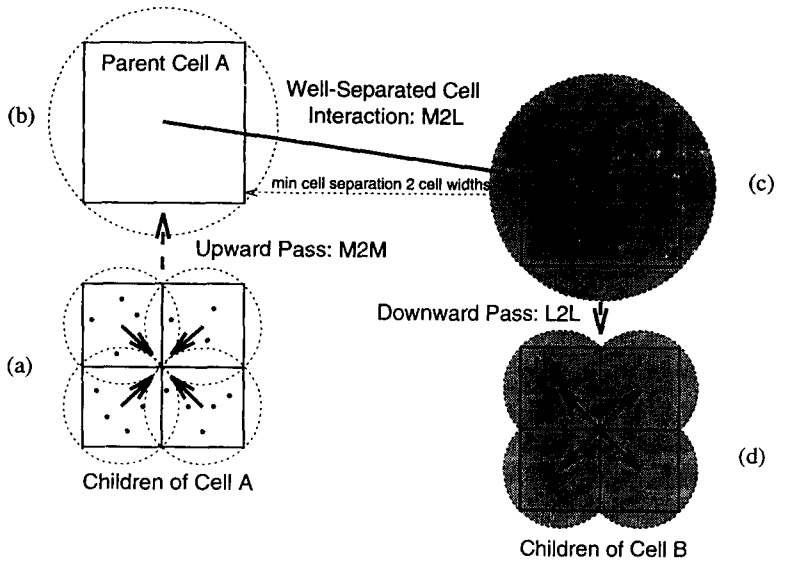

Fig. 6. The Fast Multipole Algorithm mechanisms: (a),(b) after spatial decomposition, the child cells use the Multipole to Multipole translation to shift their multipole expansion to the center of the parent cell, (c) using the Multipole to Local translation, well-separated cells interact by creating a local expansion at the center of cell B due to cell A, (d) the children of cell B feel the potential of cell A by using the Local to Local translation to shift the parent's (cell B) local expansion.

(by authors admission). Therefore, the only conclusion to be made about the efficiency of their implementation regards the marginal overhead of doing PBC over a single unit cell, roughly 2%.

5.2. *Reduced Cell Multipole Method*

The Reduced Cell Multipole Method [19], **RCMM**, attempts to reduce the cost of the Ewald sum by utilizing the hierarchical approach of [27] and [18]. The main difference is that interactions between the unit cell and near neighbors image of the unit cell (26 cells) are computed using the cell multipole method of [18], which is very similar to FMA, while interactions with the distant cells are calculated using the Ewald sum.

To compute the distant interactions efficiently, each distant unit cell is replaced by a reduced cell. The reduced cell consists of 35 randomly placed charged particles. Each of the 35 charges is assumed to be a point charge whose strength q_i is calculated such that the first five moments of the reduced cell equal those of the original cell. The authors claim that this method is highly accurate and that it scales linearly with the number of atoms in the unit cell. It should be pointed out that the 35-particle reduced cell approximates the moments of a particle system only up to the 5th-order, hence for simulations that require accuracy beyond the 5th-order, RCMM may be limited to an average accuracy.

5.3. *Particle^3–Mesh/Multipole Expansion Method*

The **PPPM/MPE** method, Particle–Particle Particle–Mesh/Multipole Expansion, by Shimada et al. [46,47] is basically an extension of Hockney and Eastwood's method [31] for a periodic system with the aid of multipole expansions. Firstly, the method starts by partitioning the simulation box into $M_1 \times M_2 \times M_3$ cells. At the center of each cell is a mesh point at which the potential, multipole expansion, etc. are considered. The electrostatic potential (force) exerted on particle i is decomposed into the PP (particle–particle) and PM (particle–mesh) interactions. Secondly, the short-range PP interactions are computed directly between particle pairs in the same or neighboring cells. Thirdly, the long-range PM interactions of the remaining cells that are well separated

from particle i are evaluated at i's cell center by expressing the potential due to all remote cells as a multipole expansion. The PM techniques are then employed [31] which relay on the use of FFT rather than hierarchical schemes of [26]. The PM potential (force) evaluations are considered to be a smooth function of the grid coordinates and hence the results can be interpolated back to the particles' locations.

The performance of the P^3M/MPE method improved when the twin-range procedure was incorporated in the following fashion. The PP interactions were calculated at each time step using the most up-to-date particles' locations, while the PM interactions were only updated every 10–20 time steps. This improvement has reportedly reduced the CPU time by a factor of three on average.

In their paper [46], the authors stated that PPPM methods alone "do not give extremely accurate results" and hence should be used with caution for precise, long time-scale simulations. The paper also compared the accuracy and CPU time of their method to Hockney and Eastwood's method by examining two systems (BPTI and a random configuration). The comparison regarded both Hockney's et al. and Shimada's et al. methods as "nearly comparable" in overall performance.

5.4. Macroscopic Multipole Method

The Macroscopic Multipole Method [36], **MMM**, employs fast multipole techniques to calculate electrostatic forces in a system of finite periodic unit cells arranged in a lattice of $3^k \times 3^k$ cells in 2D or $3^k \times 3^k \times 3^k$ in 3D in $\mathcal{O}(N)$ operations. The method is based on the observation, also realized by Schmidt and Lee [44], that once the multipole expansion of the unit cell is computed, translates of this unit cell (i.e. periodic images) are independent of the position of the cell and hence have the same multipole coefficients as the unit cell. The algorithm presented here utilizes the Fast Multipole Algorithm mechanisms [26,9], however it introduces new error criteria to determine the number of macroscopic images it needs to maintain an appropriate error bound.

In a point charge system, given p (the number of multipole terms) and k (lattice size), the macroscopic multipole method commences by dividing the simulation cell recursively into smaller sub-cells as in the FMA method. The multipole expansion of each sub-cell at the finest level of refinement is calculated and subcells are grouped into bigger structures up to the unit cell in what is known as the upward pass. At this point, macroscopic multipole procedures are called to compute the multipole expansion M_i for cell S_i, a multiple of the unit cell S_0, for $i = 1, \ldots, k-2$, by using multipole to multipole transformations. The following step starts at the highest level ($i = k$) and converts the multipole expansion M_i to a local expansion about the unit cell center only if the macroscopic region at this level is "well separated" from the central unit cell, otherwise the region is subdivided recursively into 9 cells (2D) or 27 cells (3D) and this step is repeated again. At the deepest level of recursion, the forces are evaluated directly between cells that are not well separated. Once all macroscopic cells are dealt with, the downward pass of the FMA can proceed.

In practical simulations, $k = 3, \ldots, 6$ and $p = 8, 16$ are adequate for average and high accuracy simulations. This algorithm is also highly efficient incurring only an overhead of about 25–30% over FMA simulation of a single unit cell. The results of this method were within 3–4 significant figures of the Ewald sum results for $p = 8$ and $k = 4$. However, the method is capable of achieving higher accuracy by increasing p at the expense of longer execution time. In addition, this method can efficiently handle non-cubic systems, i.e. $3^i \times 3^j \times 3^k$ lattice ($i \neq j \neq k$), which allows the study of surfaces, i.e. systems that are finite in one of three dimensions.

5.5. Other related approaches

Recently, Berman and Greengard [8] introduced a new general method to rapidly evaluate lattice sums of an infinite lattice of certain potential energy functions, e.g. the Ewald sum potential. This method is based on a new renormalization identity and has been developed for both 2D and 3D systems. Periodicity is accomplished by assuming that space is filled with translates of the unit cell and hence each image has the same far field expansion relative to its center. For electrostatic point charge systems, the method utilized multipole and Taylor

expansions to arrive at a recursive, infinite sum that required the evaluation of certain finite sums as a function of the coordinates of the lattice points.

The Ewald sum method has been modified in [41,29] to simulate systems that are infinite only in two of the three dimensions with long-range interactions. Examples of such systems are biological membrane and polar fluids. Such quasi-two-dimensional systems cannot utilize 3D implementations of the Ewald sum as this has been shown to be computationally inefficient and may lead to unrealistic interactions between the sheets of the finite dimension. The authors have presented a reformulation of the Ewald sum that handles such systems and used a system of water trapped between two dielectrics to test their methods. The method is reported to be of reasonable speed and accuracy but requires large memory.

In summary, this section presented an account of Multipole-based Ewald sum methods. These methods have an attractive computational cost of $\mathcal{O}(N) - \mathcal{O}(N \log N)$. Among the methods reviewed above, the Macroscopic Multipole Method is the better method since it combines accuracy and efficiency with the flexibility to simulate systems of any geometrical configuration.

6. Concluding remarks

This paper presented a survey of the different approaches to simulating electrostatic point charge systems in periodic boundary conditions via Ewald sums. The paper has reviewed the popular approaches along with some recent state of the art methods that handle Ewald sums more efficiently, i.e. Fourier- and multipole-based methods. Fourier-based techniques perform the "true" Ewald sum and rely on the reformulation of the reciprocal-space sum into a form that is effectively calculated using FFT in a $\mathcal{O}(N \log N)$ method. Multipole-based methods, theoretically $\mathcal{O}(N)$ algorithms, have been suggested as feasible alternatives to the Ewald sum. The computational efficiency and accuracy of multipole-based methods place them as strong contenders against today's fastest true Ewald sum methods. Ewald summation methods that truly evaluate the infinite sum will remain a favorable approach in the MD community as a well established simulation technique. Furthermore, the Ewald sum is still considered more suitable for crystalline structures than any other method. Presently, there is no conclusive evidence as to which of the methods (Fourier-based Ewald or periodic FMA) have better performance as such comparisons are strongly dependent on the implementation of the algorithm and the optimizations for a particular computer. Reports of the break-even point, i.e. the number of particles at which the two methods are equally fast, have ranged from $N = 300$ [19], to $N = 30\,000$ [49]. Recently, a break-even point of $N = 100\,000$ between FMA and a direct implementation of Ewald summation was reported by [22]. To put Fourier- and Multipole-based Ewald methods in perspective, we have tabulated sample simulation results as reported in the literature, Table 1.

From this account and our experience in evaluating long-range electrostatics we offer the following concluding remarks:

- Algorithms from all three approaches (standard, Fourier, and Multipole-based Ewald sum) are being used in MD "production" packages. Standard Ewald sum implementations, Section 3, are relatively the easier to program, while Multipole-based algorithms, Section 5, are more difficult to program with Fourier-based methods, Section 4, somewhere in the middle.
- Fourier- and Multipole-based methods are both competitive approaches for evaluating the Ewald sum. Small systems, $N \leqslant 1000$ particles, can be efficiently simulated with standard Ewald sum techniques. For system sizes on the order of 10^3–10^4, Fourier-based methods are likely to be faster, whereas for system sizes of 10^5 and above, Multipole-based methods are probably more efficient.
- Particle systems that are inherently periodic, e.g. crystals, can be simulated highly accurately with Fourier-based methods very efficiently.
- The Fast Multipole Algorithms have been extended to compute the Leonard-Jones potential [21] and Polarization [35] which are likely to be further extended for systems with PBC. This will provide Multipole-based

methods with the flexibility to simulate more realistic potentials for periodic particle systems. It is worth mentioning that the PME method has been extended to include the Leonard-Jones potential [15] as well. There exist numerous methods to perform the Ewald sum, which is indicative of the importance of the problem and the necessity of finding efficient algorithms for solving it. The survey presented here is meant to provide a thorough account of the available competitive methods to the best of the authors knowledge.

Acknowledgements

This work was supported by NSF grant ASC-9318159 and by NIH grant RR-08102-01. The authors wish to thank Tom Darden for his valuable suggestions and useful discussions.

References

[1] M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, AMS 55 (US Government, Washington, DC, 1964).
[2] D. Adams and G. Dubey, J. Comp. Phys. 72 (1987) 156.
[3] M. Allen and D. Tildesley, Computer Simulation of Liquids (Oxford Science, London, 1990).
[4] C.R. Anderson, SIAM J. Sci. Stat. Comput. 13 (1992) 923.
[5] J. Bader and D. Chandler, J. Phys. Chem. 96 (1992) 6423.
[6] M. Belhadj, H. Alper and R. Levy, Chem. Phys. Lett. 179 (1991) 13.
[7] H.J. Berendson, in: Computer simulation of biomolecular systems 2, eds. W.F. van Gunsteren and P.K. Weiner (ESCOM Science Publishers, Leiden, The Netherlands, 1993) p. 161.
[8] C. Berman and L. Greengard, J. Math. Phys. 35 (1994) 6036.
[9] J. Board Jr., R. Batchelor and J. Leathrum Jr., Proc. AIAA/ASME Thermophysics and Heat Transfer Conf. (1990).
[10] J. Board Jr., J. Causey, J. Leathrum Jr., A. Windemuth and K. Schulten, Chem. Phys. Lett. 198 (1992) 89.
[11] M. Born and Th. Von Karman, Physik. Z. 13 (1912) 297.
[12] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan and M. Karplus, J. Comput. Chem. 4 (1983) 187.
[13] S. Brush, H. Sahlin and E. Teller, J. Chem. Phys. 45 (1966) 2101.
[14] T. Darden, D. York and L. Pederson, J. Chem. Phys. 98 (1993) 10089.
[15] T. Darden, U. Essmann, H. Lee, L. Perera, M. Berkowitz and L.G. Pederson, J. Chem. Phys. 103 (1995) 8577.
[16] M. Deem, J. Newsam and S. Sinha, J. Phys. Chem. 94 (1990) 8356.
[17] S. De Leeuw, J. Perram and E. Smith, Proc. Roy. Soc. London A 373 (1980) 27.
[18] H. Ding, N. Karasawa and W.A. Goddard III, J. Chem. Phys. 97 (1992) 4309.
[19] H. Ding, N. Karasawa and W.A. Goddard III, Chem. Phys. Lett. 196 (1992) 6.
[20] J. Eastwood, R. Hockney and D. Lawrence, Comput. Phys. Commun. 19 (1980) 215.
[21] W. Elliott and J. Board, Technical Report 94-005, EE Dept. Duke University (1994), unpublished.
[22] K. Esselink, Comput. Phys. Commun. 87 (1995) 375.
[23] P. Ewald, Ann. Phys. 64 (1921) 253.
[24] R. Farouki and S. Hamaguchi, J. Comp. Phys. 115 (1994) 276.
[25] D. Fincham, Mol. Simulation 13 (1994) 1.
[26] L. Greengard and V. Rokhlin, J. Comp. Phys. 73 (1987) 325.
[27] L. Greengard, The Rapid Evaluation of Potential Fields in Particle Systems (MIT Press, Cambridge, MA, 1988).
[28] J. Hansen, Phys. Rev. A 8 (1973) 3096.
[29] J. Hautman and M. Klein, Mol. Sim. 75 (1992) 379.
[30] D. Heyes, J. Chem. Phys. 74 (1980) 1924.
[31] R. Hockney and J. Eastwood, Computer simulation using particles (McGraw-Hill, New York, 1981).
[32] J.D. Jackson, Classical Electrodynamics (Wiley, New York, 1975).
[33] C. Kittel, Introduction to Solid State Physics (Wiley, New York, 1971).
[34] J. Kolafa and J. Perram, Mol. Simulation 9 (1992) 351.
[35] R. Kutteh and J. Nicholas, Comput. Phys. Commun. 86 (1995) 236.
[36] C. Lambert and J. Board Jr., J. Comp. Phys., submitted.
[37] B. Luty, M. Davis, I. Tironi and W. van Gunsteren, Mol. Simulation 14 (1994) 11.
[38] B. Nijboer and F. De Wette, Physica 23 (1957) 309.

[39] J. Perram, H. Petersen and S. De Leeuw, Mol. Phys. 65 (1988) 875.
[40] G. Rajagopal and R. Needs, J. Comp. Phys. 115 (1994) 399.
[41] Y. Rhee, J. Halley, J. Hautman and A. Rahman, Phys. Rev. B 40 (1989) 36.
[42] Z. Rycerz and P. Jacobs, Mol. Simulation 8 (1992) 197.
[43] M. Sangester and M. Dixon, Adv. Phys. 25 (1976) 247.
[44] K. Schmidt and M. Lee, J. Stat. Phys. 63 (1991) 1223.
[45] H. Schreiber and O. Steinhauser, Biochemistry 31 (1992) 5856.
[46] J. Shimada, H. Kaneko and T. Takada, J. Comp. Chem. 14 (1993) 867.
[47] J. Shimada, H. Kaneko and T. Takada, J. Comp. Chem. 15 (1994) 28.
[48] W. Slattery, G. Doolen and H. DeWitt, Phys. Rev. A 21 (1980) 2087.
[49] D. Solvason, J. Kolafa, J. Peterson and J. Perram, Comput. Phys. Commun. 87 (1995) 307.
[50] W. Smith, Information Quarterly for Computer Simulation of Condensed Phases 21 (1986) 37.
[51] M. Tosi, in: Solid State Physics 16, eds. F. Seitz and D. Turnbull (Academic Press, New York, 1964) p. 107.
[52] F. Von der Lage and H. Bethe, Phys. Rev. 71 (1947) 612.
[53] D. York and W. Yang, J. Chem. Phys. 101 (1994) 3298.
[54] D. York, A. Wlodawer, L. Pedersen and T. Darden, Proc. Natl. Acad. Sci. 91 (1994) 8715.