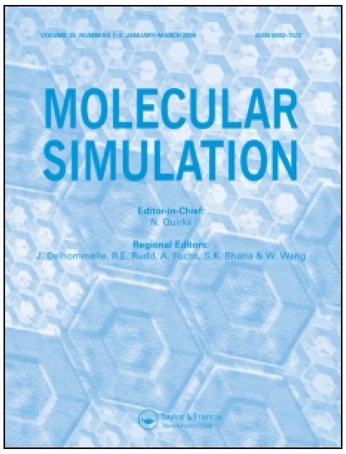

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

An Alternative to Ewald Sums. Part 3: Implementation and Results

R. Strebel[a]; R. Sperb[b]
[a] Institut für Wissenschaftliches Rechnen, [b] Seminar für Angewandte Mathematik, Zürich

To cite this Article Strebel, R. and Sperb, R.(2001) 'An Alternative to Ewald Sums. Part 3: Implementation and Results', Molecular Simulation, 27: 1, 61 — 74
To link to this Article: DOI: 10.1080/08927020108024519
URL: http://dx.doi.org/10.1080/08927020108024519

Molecular Simulation, 2001, Vol. 27, pp. 61–74
Reprints available directly from the publisher
Photocopying permitted by license only

AN ALTERNATIVE TO EWALD SUMS. PART 3: IMPLEMENTATION AND RESULTS

R. STREBEL[a] and R. SPERB[b,*]

[a]*Institut für Wissenschaftliches Rechnen,*
[b]*Seminar für Angewandte Mathematik, ETH-Zentrum, CH-8092 Zürich*

(Received June 2000; accepted July 2000)

This paper describes the implementation of a method for computing the Coulombic interaction in a periodic system. If the basic cell contains n charges the CPU time required to compute all forces and the total energy is $O(n \cdot \log n)$ in contrast to Ewald's method with $O(n^{3/2})$.

Keywords: MMM algorithm; Ewald sum; Coulomb interactions

1. BASIC FORMULAE

We consider a basic cell C containing n charges q_i with total charge zero. For simplicity we restrict ourselves to the most important case that C is a cube of side 1. The case that C is an arbitrary orthorhombic box requires some small corrections. This case is discussed in [7]. The Coulomb potential can then be written as

$$\phi(\vec{r}) = \sum_{\vec{\ell} \in \mathbb{Z}^3} \sum_{i=1}^{n} \frac{q_i}{|\vec{r} - \vec{r}_i + \vec{\ell}|}, \tag{1.1}$$

where $\vec{r}$ is the position vector of an arbitrary point in C, $\vec{r}_i$ is the position vector of the charge q_i, and $\vec{\ell}$ is a lattice vector, *i.e.*, the components of $\vec{\ell}$ are in $\mathbb{Z}$.

*Corresponding author.

The force on a given charge q_j is then obtained from the gradient of ϕ where the singular term occurring in (1.1) for $i=j$ and $\vec{\ell}=(0,0,0)$ is omitted.

It was shown in [5] that ϕ can be written as

$$\phi(x,y,z)=\sum_{\substack{\vec{\ell}\in\mathbb{Z}^2\\|\vec{\ell}|>0}}\frac{1}{|\vec{\ell}|}\frac{\cosh(\pi|\vec{\ell}|(1-2|z|))}{\sinh(\pi\cdot|\vec{\ell}|)}\cos(2\pi\vec{\ell}\cdot\vec{\rho}_1)+2\pi(z^2-|z|) \quad (1.2)$$

where $\vec{\rho}_1=(x,y)$. Note that here $\vec{\ell}\in\mathbb{Z}^2$ in contrast to the corresponding expression of the Ewald summation where $\vec{\ell}\in\mathbb{Z}^3$. Although not obvious from (1.2) ϕ is symmetric in x, y, z as it has to be.

The Coulomb energy can then be written (see [5]) as

$$E=\frac{1}{2}\sum_{i\neq j=1}^{n}q_iq_j\phi(x_i-x_j,y_i-y_j,z_i-z_j)+Q_0\cdot\sum_{i=1}^{n}q_i^2+\frac{2\pi}{3}\left(\sum_{i=1}^{n}q_i\vec{r}_i\right)^2. \quad (1.3)$$

The dipole term on the very right of (1.3) has to be added if one takes spherical means in the conditionally converging sums (see the discussion in [3]). The constant Q_0 is given by analytical expressions (see [4–6]) and the numerical value is -1.942248.

If the distance between two charges is very small the sum in (1.2) converges very slowly. An alternative expression for ϕ was derived in [5] and [6]. It can be written in the form

$$\phi=\frac{1}{r}+\hat{\phi}, \quad (1.4)$$

with $r=\sqrt{x^2+y^2+z^2}$ and $\hat{\phi}(x,y,z)$ is regular at (0,0,0). The rather complicated expression for $\hat{\phi}$ is

$$\begin{aligned}\hat{\phi}(x,y,z)=&\sum_{|\vec{\ell}|>0}\sum_{p=1}^{\infty}K_0[2\pi p|\vec{\rho}_2+\vec{\ell}|]\cos(2\pi px)-\sum_{m\neq 0}L[y,z+m]\\&+2\mathrm{Re}\left[\sum_{m\geq 1}\frac{b_{2m}}{2m(2m!)}(2\pi(z+iy))^{2m}\right]\\&+\sum_{k\geq 0}\binom{-1/2}{k}\frac{\psi^{(2k)}(1+x)+\psi^{(2k)}(1-x)}{(2k)!}(y^2+z^2)^k\\&+2\pi(z^2-|z|)-2\log(4\pi)+4\gamma. \quad (1.5)\end{aligned}$$

Here K_0 is the Besselfunction and the ψ are Polygamma functions, b_{2m} are Bernoulli numbers, γ is the Euler constant and $\vec{\rho}_2$ stands for (y, z). Further,

$$L[y,z] := \log\left[1 - 2\cos(2\pi y)\exp(-2\pi|z|) + \exp(-4\pi|z|)\right]$$

is used as an abbreviation as well.

The self energy constant Q_0 is thus given by $(1/2)\hat{\phi}(0,0,0)$.

The regular part $\hat{\phi}$ is used for charges that are close to each other and then the series converge very quickly.

An analysis of the rate of convergence of the sums in (1.2) and (1.5) is carried out in [7]. For a maximal error of 10^{-4} one should use for example (1.5) for $|z| \leq 0.35$ and then switch to the form (1.2).

2. APPLICATION OF PRODUCT EXPANSION

Let us first repeat the simple idea of a product expansion. Assume we have to calculate an expression of the form (*e.g.*, the Coulomb energy)

$$E = \sum_{i,j=1}^{n} f(x^i, x^j) \tag{2.1}$$

and suppose one has an expansion of the form

$$f(x^i, x^j) = \sum_{\ell=1}^{\infty} a_\ell g_\ell(x^i) h_\ell(x^j). \tag{2.2}$$

Then we can write

$$E = \sum_{\ell=1}^{\infty} a_\ell \sum_{i=1}^{n} g_\ell(x^i) \sum_{j=1}^{n} h_\ell(x^j) \cong \sum_{\ell=1}^{L} a_\ell \sum_{i=1}^{n} g_\ell(x^i) \sum_{j=1}^{n} h_\ell(x^j). \tag{2.3}$$

The expression in (2.1) requires n^2 terms whereas the approximation in (2.3) needs only $2 \cdot L \cdot n$ terms. Since we are interested in applications with large n ($10^3 - 10^7$) the product expansion will drastically reduce the amount of work required for a given accuracy. The product expansion for the terms occurring in (1.2) is elementary since we have to deal with trigonometric and exponential functions, and a quadratic term only.

For the form (1.5), *i.e.*, the regular part $\hat{\phi}$ we will use an expansion in cosh-functions as described in [6] and this in turn leads to a product expansion.

Let us briefly describe the method for the form (1.2) of the potential. Consider the interaction of two cubic boxes B_1 and B_2 given by

$$I_{12} = \sum_{q_i \in B_1} \sum_{q_j \in B_2} \phi(x^i - x^j, y^i - y^j, z^i - z^j). \tag{2.4}$$

Neglecting the quadratic term in (1.2) the interaction I_{12} may be written in the form

$$\begin{aligned} I_{12} = \sum_{\nu > 0} c(\nu) \sum_{q_i \in B_1} \sum_{q_j \in B_2} \sum_{\alpha=1}^{8} & e^1(\nu, z_i) t^1_\alpha(\nu, x_i) s^1_\alpha(\nu, y_i) \\ & \times e^2(\nu, z_j) t^2_\alpha(\nu, x_j) s^2_\alpha(\nu, y_j) \end{aligned} \tag{2.5}$$

$$\begin{aligned} I_{12} &= \sum_{\nu > 0} c(\nu) \sum_{\alpha=1}^{8} \sum_{q_i \in B_1} e^1_i t^1_{\alpha i} s^1_{\alpha i} \sum_{q_j \in B_2} e^2_j t^2_{\alpha j} s^2_{\alpha j} \\ I_{12} &= \sum_{\nu > 0} c(\nu) \sum_{\alpha=1}^{8} S^\alpha_1 \cdot S^\alpha_2. \end{aligned} \tag{2.6}$$

Here $\nu = |\vec{\ell}|$, $e^1(\nu, z_i)$, $e^2(\nu, z_j)$ are exponential terms of the form $\exp(-2\pi \nu z_i)$, $\exp(2\pi \nu z_j)$ and $t^1_\alpha, t^2_\alpha, s^1_\alpha, s^2_\alpha$ are sin or cos terms with arguments $2\pi x_i$, $2\pi y_i$ or $2\pi x_j$, $2\pi y_j$.

Equation (2.6) shows a decoupled form of sums extended separately over the two boxes B_1 and B_2. A similar equation can be derived for the regular part $\hat{\phi}$.

So far we have only exploited the fact that ϕ and $\hat{\phi}$ may be expanded into products. In the following we will also make essential use of the other features of ϕ:

(a) ϕ is symmetric in x, y, z
(b) ϕ is 1-periodic in x, y, z
(c) The number of terms required for given accuracy decreases with increasing distance of two boxes B_1, B_2.

3. ALGORITHM

In this section, we describe the algorithm to compute ϕ and $\hat{\phi}$ in a graphical way. We have seen above how to efficiently compute ϕ for two subsets B_1 and B_2 by using the product expansion, as long as B_1 and B_2 are well separated. We illustrate the computation of the interaction from a *grey* subset B_1 to a *black* subset B_2 as shown in Figure 1.

By combining the contributions of all grey boxes, which are distant enough from the black box, we obtain the following splitting into far and near contributions as shown in Figure 2.

Note that all the interactions are still assumed to be periodic, so the near contribution may actually be represented more clearly as shown in Figure 4.

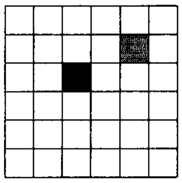

FIGURE 1

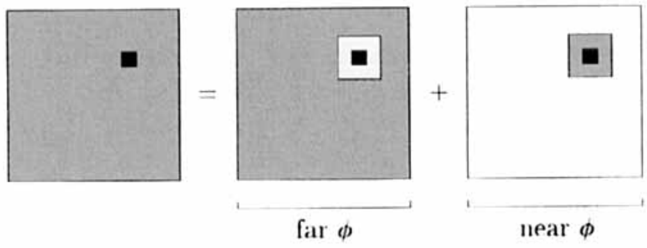

FIGURE 2

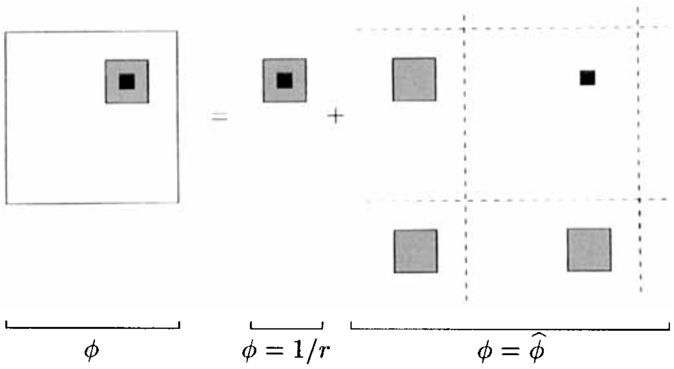

FIGURE 3

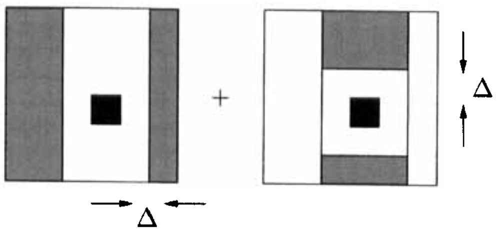

FIGURE 4

We will describe later how to efficiently compute the simple Coulomb potential $\phi = 1/r$ and the smooth contribution $\phi = \hat{\phi}$ from the mirror images.

3.1. Far Contribution ϕ

An analysis of the expansion (1.2) shows that the number $\#\phi$ of terms required in the expansion is proportional to $1/\Delta^2$, where Δ is the minimal separation of the black box from the grey region. The situation for reductions along the z- and y-axes is sketched below in Figure 4.

If we restrict ourselves to reduce the problem only once along each coordinate axis with some separation $\Delta = \Delta(n)$, then the time complexity of the algorithm turns out to be $O(n^{7/5})$. The exponent 7/5 can be derived from the following considerations. Assuming that the near interaction is computed *pairwise*, the amount of work is $O(n^2\Delta^3)$ for the near ϕ. Using $\#\phi \sim 1/\Delta^2$, the amount of work for the far contribution is given by $O(n/\Delta^2)$. Letting $O(n^2\Delta^3) = O(n/\Delta^2)$, we obtain $1/\Delta = O(n^{1/5})$, and the result $O(n^{7/5})$ drops out.

We can reduce the time complexity further from $O(n^{7/5})$ to $O(n \log n)$, if we reduce the problem in a hierarchical way described in Figure 5.

This approach makes use of the fact that $\#\phi$, the number of terms in the expansion, is actually proportional to $(\lambda/\Delta)^2$, where λ is the side length of the cubic box. The above sketch shows that $\lambda = 1$ for the first level, $\lambda = 1/2$

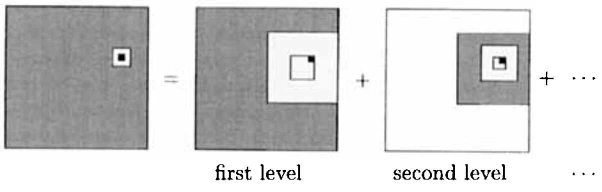

FIGURE 5

for the second level, and so forth. The second level can be regarded as a scaled-down version of the original problem, so the relation $\#\phi \sim (\lambda/\Delta)^2$ is natural from a geometric point of view. Note, however, that this is strictly correct only if we count the number of terms $\#\phi$ for a given *relative* error tolerance.

The hierarchic approach fails so far because the scaled-down problem is *not periodic* any more. Consequently, the method is to *make* the problems in the second and further levels periodic. This may be achieved by adding artificial mirror images, the idea is sketched below in Figure 6 for a simple reduction along one dimension.

The pictorial equation in words is about the following. We want to compute a *near* λ-periodic problem in a $\lambda/2$-box, which represents the second level in the hierarchic scheme. This problem may be reduced to a *periodic* problem with period $\lambda/2$ minus a *far* correction in the λ-period. Basically, this correction consists of artificial mirror images, and their contribution requires the same terms, which are already needed for the usual computation. Consequently, the additional cost is small, and we have an efficient $O(n \log n)$ method by using this hierarchic approach.

The $\log n$ factor in the time complexity stems from the number of levels required, so that we are left with small boxes containing relatively few particles. However, this complexity measure is too pessimistic from a practical point of view, since the terms for the second and further levels are significantly *cheaper* than the terms for the first level. Basically, this follows from the scaling property of $1/r$, which is homogeneous in the variables x, y, z. For instance, if the first level requires the computation of $e^{\beta z}$, then the second level needs $e^{2\beta z} = (e^{\beta z})^2$. The relationship is similar for the $\sin(\cdots)$ and $\cos(\cdots)$, where we can apply the addition theorem. Therefore, we only need to compute exponentials and trigonometric functions for the first level, while terms for later levels can be derived by *squaring* terms from the previous level. To exploit this property, several levels along a coordinate axis are computed at once, as illustrated below in Figure 7.

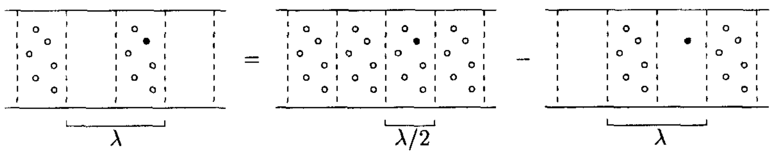

FIGURE 6

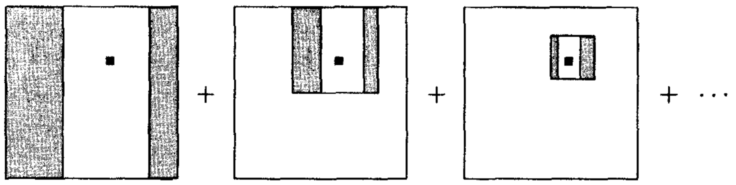

FIGURE 7

3.2. Near Contribution $\hat{\phi}$

We have already seen the splitting of the near contribution into a Coulomb $1/r$ potential and the smooth $\hat{\phi}$ portion from the mirror images shown in Figure 8.

Although the Coulomb potential $\phi = 1/r$ looks simple, a highly efficient implementation is *not* straightforward. Most prominently, we have optimized the computation of $1/\sqrt{x}$. Several other implementation issues are nontrivial, as well.

In some way, things are reversed for the potential $\hat{\phi}$. The formula (1.5) for $\hat{\phi}$ looks scary, but for small values of x, y and z the function is smooth and friendly. Since $\hat{\phi}$ is an even function in its arguments x, y and z, a simple approximation is given by

$$\hat{\phi} = \sum_{i,j,k} c_{ijk} \underset{\alpha_i x}{\cosh} \underset{\alpha_j y}{\cosh} \underset{\alpha_k z}{\cosh}, \tag{3.7}$$

where the coefficients c_{ijk} and factors α_k may be determined to minimize the maximal absolute error, for example. This is essentially an exponential

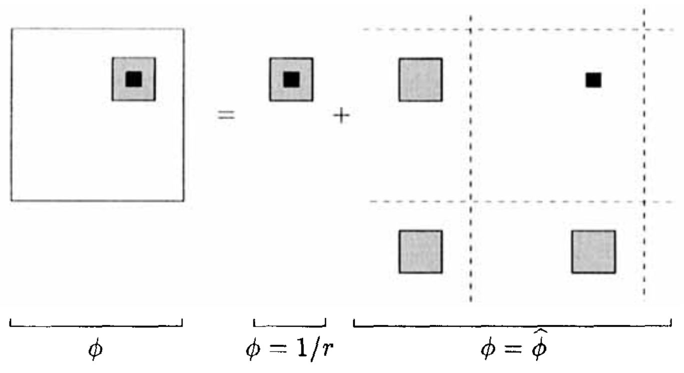

FIGURE 8

TABLE I

m	1	2	3	4
cosh	4.0×10^{-5}	1.6×10^{-6}	4.0×10^{-8}	2.0×10^{-10}
cos	7.9×10^{-4}	1.0×10^{-6}	4.0×10^{-8}	1.0×10^{-10}

TABLE II

m	1	2	3	4
cosh	6.0×10^{-7}	4.0×10^{-9}	1.6×10^{-10}	3.2×10^{-13}
cos	5.0×10^{-5}	1.3×10^{-8}	1.6×10^{-10}	2.5×10^{-13}

expansion for $\hat{\phi}$, and we may apply the ideas of product expansion to compute it efficiently. Another option is to use $\cos(\cdots)$ instead of $\cosh(\cdots)$.

The number of terms required in (3.7) depends obviously on x, y and z. If we let $\alpha_0 = 0$ and assume that $|x|, |y|, |z| \leq 1/6$, then the maximal absolute errors in $\hat{\phi}$ for $m+1$ factors $\alpha_0, \ldots, \alpha_m$ are as given in Table I.

We conclude that the cosh expansion is more accurate for few terms, and the cos expansion is slightly more accurate for $m \geq 4$. The cosh is both more efficient and easier to implement, since it contains exponential functions instead of trigonometric functions. Therefore, nearly everything is in favour of the cosh expansion.

For very high accuracy, an alternative to increasing m is to reduce the range of allowed x, y and z. Using the notation from above, and assuming $|x|, |y|, |z| \leq 1/12$, the maximal absolute errors are given by Table II.

This data clearly confirms the above conclusions.

4. EFFICIENCY

In this section, we investigate the execution times of some elementary functions, and perform a quantitative and qualitative comparison of our method MMM[1] to PPPM and Ewald's method, described in [2] and [1], respectively. We have optimized these algorithms for the Alpha 21164 microprocessor, so the results herein do not necessarily apply to other processors. The particular machine we were performing the experiments on has a clock rate of 500 MHz, 8M external cache, and 512M main memory.

[1]MMM is the name of our algorithm.

4.1. Elementary Functions

To compute the Coulomb potential $\phi = 1/r$ for given $r^2 = x^2 + y^2 + z^2$, the efficient computation of $1/\sqrt{r^2}$ is of paramount importance. A straightforward standard solution is to compute $u = \sqrt{r^2}$ first and then $v = 1/u$. On the Alpha 21164, $\sqrt{r^2}$ is computed by a reasonably efficient library routine, and the division $1/v$ is done in hardware. We have exploited the facts that we may compute $1/\sqrt{r^2}$ *directly*, and that we have to compute *several* $1/\sqrt{r_k^2}$ simultaneously. A manually optimized version turns out to be about 7 times faster compared to the standard approach, if we want to compute *many* $1/\sqrt{r_k^2}$. Table III lists the execution time in *ns* per $1/\sqrt{r^2}$ for varying sizes of the argument vector $(r_1^2, \ldots, r^2 d_n)$. We use the standard approach for $n = 1$ and the manually optimized version for $n > 1$.

Since we are using exponential expansions, the computation of exponential and trigonometric functions cos and sin is important as well. In our specific problem, we always need *both* e^x and e^{-x}, so our manually optimized routine computes pairs (e^{+xk}, e^{-xk}) for a given argument vector $(x_1, \ldots, x_n)$. For the comparison, we use the standard approach, which computes one exponential by a library routine and uses the inverse for the other, *i.e.*, $y_+ = e^x$ and then $y_- = 1/y_+$. Table IV displays the execution times in *ns* per

TABLE III Execution time for $1/\sqrt{r^2}$

n	$t[ns]$	▬ = 20 ns
1	163.0	▬▬▬▬▬▬▬▬
4	50.0	▬▬▬
8	35.4	▬▬
16	28.1	▬
32	24.5	▬
64	23.8	▬
128	21.7	▬
256	21.6	▬

TABLE IV Execution time for $e^{\pm x}$

n	$t[ns]$	▬ = 30 ns
1	185.0	▬▬▬▬▬▬
4	95.7	▬▬▬
8	64.6	▬▬
16	50.0	▬▬
32	41.2	▬
64	36.4	▬
128	34.5	▬
256	33.9	▬

pair $e^{\pm x}$ for varying vector sizes n. Again, the standard approach is used for $n = 1$ and the manually optimized routine for $n > 1$. For $n \geq 64$, the optimized code is about 5 times faster than the standard approach.

4.2. Comparison

In this subsection, we investigate the time complexity of MMM first. Subsequently, we compare MMM to PPPM and Ewald's method. The error ε will be computed from the exact forces $\mathbf{F}_i$ and the computed forces $\bar{\mathbf{F}}_i$ for all particles by

$$\varepsilon^2 = \frac{\sum_i \|\bar{\mathbf{F}}_i - \mathbf{F}_i\|^2}{\sum_i \|\mathbf{F}_i\|^2}.$$

Another simple measure of the error which is often used compares the potentials instead of the forces, namely

$$\varepsilon_\phi^2 = \frac{\sum_i (\bar{\phi}_i - \phi_i)^2}{\sum_i \phi_i^2}.$$

For MMM we have roughly $\varepsilon_\phi \approx 2\varepsilon$ in typical cases, so there is not a huge difference.

We experimentally determine the time complexity of MMM by computing potentials and forces with an error $\varepsilon = 10^{-5}$ and varying numbers of particles from $n = 2^{10} \approx 10^3$ to $n = 2^{20} \approx 10^6$. The optimal number of box reductions grows more or less logarithmically with the number n of particles. For small n, the box is split into $\pi_3 = 6 \times 6 \times 6$ subboxes, which corresponds to 3 reductions, one along each coordinate axis. For $n = 2^{20}$, the optimal subdivision involves 10 reductions and is given by $\pi_{10} = 24 \times 24 \times 48$. Table V lists the execution times in s for varying n and varying subdivisions π_*.

TABLE V Execution time of MMM for $\varepsilon = 10^{-5}$

n	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}
2^{10}	0.13	*	*	*	*	*	*	*
2^{11}	0.17	*	*	*	*	*	*	*
2^{12}	0.27	0.35	*	*	*	*	*	*
2^{13}	0.60	0.62	0.80	*	*	*	*	*
2^{14}	1.57	1.20	1.17	1.6	*	*	*	*
2^{15}	*	3.17	2.45	2.5	3.7	*	*	*
2^{16}	*	*	*	4.6	5.4	7.5	*	*
2^{17}	*	*	*	12.5	10.8	10.3	15.4	*
2^{18}	*	*	*	*	28.3	22.1	24.7	37.3
2^{19}	*	*	*	*	*	57.8	47.3	55.0
2^{20}	*	*	*	*	*	170.8	118.2	105.0

A graphical representation of the same data is given in Figure 9. *Measured* execution times are indicated by symbols, and the optimal execution time of MMM is simply the minimum over all curves associated with different subdivisions π_*. To simplify the interpretation, the execution time is scaled by a factor σ/n for some constant σ, a method of time complexity $O(n)$ would be displayed as a constant.

For small $n \approx 1000$, we see that MMM suffers from the fact that there are not enough particles to justify subdivisions, which are required to make the near interaction $\hat{\phi}$ from the mirror smooth. Although it is possible to correct this weakness by introducing artificial boxes, it is actually more efficient to use Ewald's method for few particles $n \leq 2^{10}$. In Figure 10, the graphical representation of the execution time of Ewald's method for $n = 2^5$ to $n = 2^{12}$ is shown with the same scaling factor σ/n used for MMM's graph. The values t_ω given in the Table VI and Figure 10 indicate the deviation of our implementation of Ewald's method from the theoretical $O(n^{3/2})$ time complexity, *i.e.*, t_ω is proportional to $t/n^{3/2}$.

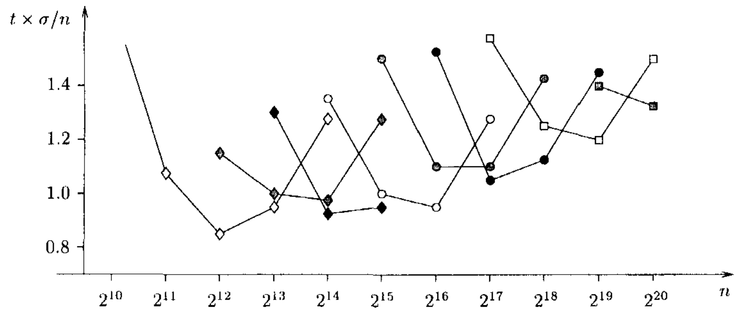

FIGURE 9 Execution time of MMM scaled by σ/n.

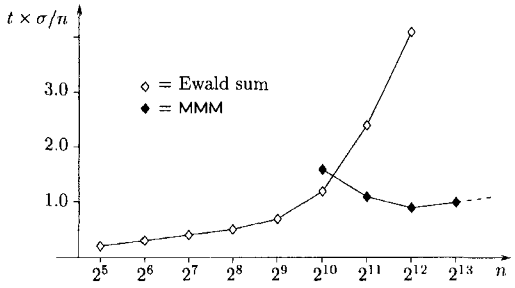

FIGURE 10 Execution time for Ewald's method with $\varepsilon = 10^{-5}$.

TABLE VI

n	$t[ms]$	t_ω
2^5	0.6	1.45
2^6	1.5	1.23
2^7	3.8	1.13
2^8	10.2	1.08
2^9	26.8	1.00
2^{10}	89.7	1.18
2^{11}	369.0	1.72
2^{12}	1260.0	2.07

TABLE VII

ε	*PPPM*	*MMM*
10^{-2}	0.3	1.0
10^{-3}	1.1	1.2
5×10^{-4}	2.9	1.4
10^{-4}	*	1.7
10^{-5}	*	2.6
10^{-6}	*	3.9

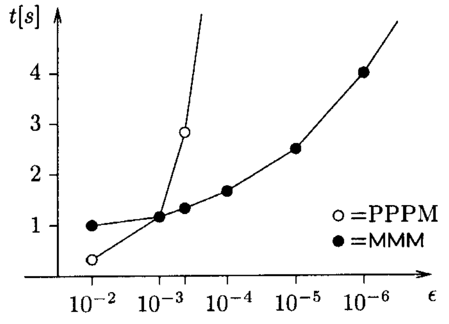

FIGURE 11 Execution times of PPPM and MMM for $n = 2^{15}$ and varying relative error ε.

In contrast to MMM and Ewald's method, PPPM is only applicable for modest accuracy requirements. But then PPPM is very efficient for small ε, as the data in Table VII and graphical representation in Figure 11 demonstrate. Note that we have tried to make the comparison as *fair* as possible by manually optimizing critical parts of PPPM as well.

The time complexity of MMM is roughly $O(\log^2 \varepsilon)$ for varying accuracy ε. The quantitative comparison of PPPM and MMM supports the initial qualitative claim that PPPM is very efficient for large ε and more or less fails for small error tolerance ε.

5. FURTHER WORK

The current implementation of MMM only solves the periodic problem with a more or less homogeneous particle distribution efficiently. However, modifications of MMM could be applied to the following problems

(i) Non-periodic Coulombic problems, for example by adding artificial mirror images, which have to be eliminated with similar methods MMM uses right now. This is not necessarily the best approach, and we think that a *specialized* non-periodic code based on exponential expansions should be superior.

(ii) Coulombic problems, which are periodic in two directions only. Again, we may use artificial mirror images along the non-periodic axis. It depends on the dimensions of the unit cell, if MMM can be applied efficiently without modifications or not.

(iii) Non-homogeneous particle distributions require a non-uniform subdivision of the unit cell into smaller boxes. This could be implemented within MMM itself, or we might use MMM on a uniform high-level and a non-periodic simpler computation for the particle clusters.

(iv) Computing interactions for the potential e^{-kr}/r instead of the Coulomb potential $1/r$. For $|k|$ not too large, the same approach as in MMM may be applied. The derivation of the exponential expansion for e^{-kr}/r is more or less identical to the case $1/r$.

References

[1] Ewald, P. P. (1920). Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.*, **64**, 253–287.

[2] Hockney, R. W. and Eastwood, C. R., *Computer Simulation Using Particles.* McGraw-Hill, New York, 1981.

[3] De Leeuw, S. W., Perram, J. W. and Smith, E. R. (1980). Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants. *Proc. R. Soc. Lond.*, **A373**, 27–56.

[4] John Lekner (1998). Coulomb forces and potentials in systems with an orthorhombic unit cell. *Molecular Simulation*, **20**(6), 357–368.

[5] Sperb, R. (1998). An alternative to Ewald sums, part 1: Identities for sums. *Molecular Simulation*, **20**(3), 179–200.

[6] Sperb, R. (1999). An alternative to Ewald sums, part 2: The Coulomb potential in a periodic system. *Molecular Simulation*, **22**(3), 199–212.

[7] Rolf Strebel, *Pieces of software for the Coulombic m body problem. Ph.D. Thesis*, ETH Zürich, February, 2000. Diss. ETH No. 13504, available by anonymous ftp as ftp://ftp.inf.ethz.ch/pub/publications/diss/th13504.ps.