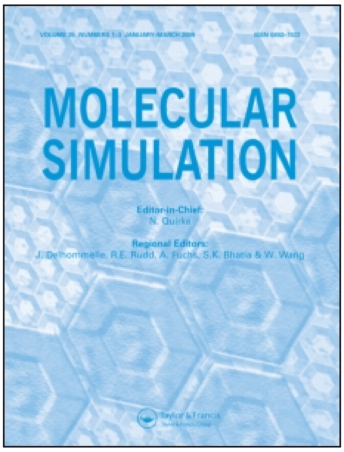

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Optimisation of the Ewald Sum for Large Systems

David Fincham[a][b]
[a] SERC Daresbury Laboratory, Warrington, UK [b] Department of Physics, Keele University, Staffordshire, UK

To cite this Article Fincham, David(1994) 'Optimisation of the Ewald Sum for Large Systems', Molecular Simulation, 13: 1, 1 — 9
To link to this Article: DOI: 10.1080/08927029408022180
URL: http://dx.doi.org/10.1080/08927029408022180

Molecular Simulation, 1994, Vol. 13, pp. 1–9
Reprints available directly from the publisher
Photocopying permitted by license only

OPTIMISATION OF THE EWALD SUM FOR LARGE SYSTEMS

DAVID FINCHAM

SERC Daresbury Laboratory, Warrington WA4 4AD, UK and Department of Physics, Keele University, Staffordshire ST5 5BG, UK

Email: D. Fincham @KEELE.AC.UK

(Received July 1993, accepted August 1993)

Techniques are described for optimising the Ewald sum in the simulation of systems containing large numbers of charged particles. Expressions are given for the best choice of parameters to maximise performance for a specified accuracy. The algorithm for the real-space sum uses a "small-cell" version of the link-cell method. Various other computational details are discussed. In the simulation of real systems there is considerable scope for speeding the calculation by reducing cutoffs, without loss of accuracy.

KEY WORDS: Ewald sum, link cells

1 INTRODUCTION

The simulation of ionic materials requires an efficient technique for handling the long-ranged Coulombic interactions. Similar considerations also arise in the simulation of many molecular systems. The most widely-used method is the Ewald sum (ES) [e.g 1]. Alternatives which have been used are the particle-particle/particle-mesh method (PPPM) [2] and the fast multipole method (FMM) [3] and its variants [4]. Whereas the ES method has an execution time which scales with N, the number of particles, as $N^{3/2}$ [5], the alternative methods scale as N. Of the two, the FMM is probably to be preferred, since it can be applied to non-periodic and inhomogeneous systems as well as periodic homogeneous systems. It is also more amenable to parallelisation by domain decomposition since it is entirely a real-space technique: both ES and PPPM cast part of the calculation in reciprocal space. FMM is complicated to program and slower than ES for small systems, but because of its order N performance it is bound to be faster for very large systems. But there is doubt about the value of N at which the crossover occurs, Ding *et al.* [6] giving a value less than 300, while Solvason *et al.* [7] say 30000. Most simulations use values of N between these two values, and in order to be able to make a rational choice of method it is important to know how to optimise both methods. This paper describes techniques which enable ES to be optimised, and a future paper will discuss FMM.

There are three aspects to the optimisation described in this paper. Section 2 discusses the choice of ES parameters based on simple arguments concerning the magnitudes of terms in the real and reciprocal sums at the cutoffs. The argument also confirms the order $N^{3/2}$ behaviour and acts as a corrective to a recent paper [8] which claims it is order N. Section 3 is concerned with optimisation of the program itself, showing how the performance of the real-space sum can be improved by a version of the link-cell method in which the size of the link cells is optimised. Several other points of practical importance are also discussed. Section 4 points out that in typical simulations it may be possible to significantly reduce cutoffs below the values suggested by the analysis of Section 2, and gives example timings.

2 CHOICE OF PARAMETERS

We will consider throughout a cubic simulation box of side L containing N particles. As we move to larger systems we of course keep the number density $n = N/L^3$ fixed, so L varies as $N^{1/3}$.

The ES decomposes the calculation of Coulombic energy and forces into real-space and reciprocal-space sums. The real-space part involves a sum over ion pairs of the form

$$\sum_{i<j} q_i q_j \operatorname{erfc}(\alpha r_{ij})/r_{ij}$$

which depends on the complementary error function erfc(αr) where α is the Ewald separation parameter. To ensure convergence of the real-space part of the sum we need to choose α so that the erfc function is small at the real-space cutoff, R. Suppose we specify "small" to be exp($-$p). Thus p is a parameter controlling the desired accuracy. At large values of the argument the erfc function behaves as $\exp(-\alpha^2 r^2)$. Thus we require

$$\alpha^2 R^2 = p$$

or

$$\alpha = p^{1/2}/R \tag{1}$$

The reciprocal-space contribution to the energy involves a sum of the form

$$\sum_k k^{-2} \exp(-k^2/4\alpha^2) \,|\sum_j q_j \exp(i\mathrm{k}\cdot\mathrm{r}_j)|^2$$

The rate of convergence of this sum is controlled by the factor $\exp(-k^2/4\alpha^2)$. If we require the terms to have the value $\exp(-p)$ at the reciprocal-space cutoff, K, then we find

$$p = K^2/4\alpha^2$$

or

$$K = 2\alpha p^{1/2} = 2p/R \tag{2}$$

using (1). Equations (1) and (2) specify the choice of separation parameter α and reciprocal space cutoff K once the accuracy parameter p and real space cutoff R have been chosen.

Now let us evaluate the execution time per time step of the real space sum. The number of ions within the cutoff sphere is

$$\frac{4\pi}{3}R^3n$$

where n is the number density. Each of the N ions interacts with the other ions in the surrounding cutoff sphere, but each pair interaction needs to be considered only once. Thus the execution time is

$$T_R = \frac{1}{2}N\frac{4\pi}{3}R^3nt_R \tag{3}$$

where t_R is the execution time to evaluate one interaction.

We also need to work out the time to evaluate the reciprocal-space sum. The volume of reciprocal space within the cutoff K is

$$\frac{4\pi}{3}K^3 = \frac{4\pi}{3}\frac{8p^3}{R^3}$$

using (2). The reciprocal-space points are given by $\mathrm{k} = \frac{2\pi}{L}(l,m,n)$ where l,m,n are integers, so the volume of reciprocal space per point is $(2\pi/L)^3$. The number of points in the cutoff sphere is thus

$$\frac{4\pi}{3}\frac{8p^3}{R^3}\frac{L^3}{8\pi^3}$$

Writing $L^3 = N/n$ to bring out the N dependence for fixed number density, this becomes

$$\frac{4\pi}{3}\left(\frac{p}{\pi}\right)^3\frac{N}{nR^3}$$

Note that for fixed cutoffs the number of k-points increases as N, because the density of points in reciprocal space increases with system size. In counting the number of points, an additional factor of a half may be included, because of the inversion symmetry of reciprocal space. A sum over the N ions must be performed for each k-point, so the execution time is

$$T_F = \frac{1}{2}\frac{4\pi}{3}\left(\frac{p}{\pi}\right)^3\frac{N^2}{nR^3}t_F \tag{4}$$

where t_F is the execution time to evaluate one term in the sum.

The total execution time is then

$$T = \frac{1}{2}\frac{4\pi}{3}\left[NnR^3t_R + \left(\frac{p}{\pi}\right)^3\frac{N^2}{nR^3}t_F\right] \tag{5}$$

The above derivation is rough and ready. We have considered only the energy, whereas in molecular dynamics one is also interested in the forces. We have not considered the overhead of locating the neighbours in the real space sum: this is discussed further below. We have ignored the factors r^{-1} and k^{-2} in the real- and reciprocal-space terms respectively, which increase the rate of convergence somewhat. A more thorough analysis can be found in [9], which estimates errors in energy and forces by making reasonable assumptions about the charge distributions beyond the cutoffs. Nevertheless the above argument is adequate to give a guide to the optimal choice of parameters.

Equation (5) shows that, for fixed p and R, T_R varies as N, but T_F varies as N^2, because of the increasing density of points in reciprocal space. Conversely, if we increase R as the system size increases in such a way that R/L is constant, T_R varies as N^2 but T_F varies as N. This suggests that by appropriate choice of parameters we may be able achieve better than N^2 behaviour in the total time.

For a given accuracy, the only free parameter is R, since this determines α and hence K by equations (1) and (2). So to find the value of R which minimises the total execution time we set $\mathrm{d}T/\mathrm{d}R = 0$. This gives

$$R_{OPT} = \left(\frac{p}{\pi}\right)^{1/2} \left(\frac{t_F}{t_R}\right)^{1/6} \frac{N^{1/6}}{n^{1/3}} \tag{6}$$

Thus the optimal choice of R increases slowly (1/6th power) with N. Substituting in Equation (5) we find for the optimal time

$$T_{OPT} = 2T_R = 2T_F = \frac{4\pi}{3} N^{3/2} \left(\frac{p}{\pi}\right)^{3/2} (t_R t_F)^{1/2} \tag{7}$$

When the total time is optimised it is equally divided between real and reciprocal space parts of the calculation. Equation (7) shows the anticipated $N^{3/2}$ behaviour. We also see that the time depends on the 3/2 power of the precision parameter p, and on the geometric mean of t_R and t_F.

To further the discussion, it is useful to introduce the dimensionless parameters

$$\bar{R} = R/L \tag{8a}$$

$$\bar{\alpha} = \alpha L \tag{8b}$$

$$\bar{K} = KL/2\pi \tag{8c}$$

Then, for precision parameter p, we have from (1) and (2)

$$\bar{\alpha} = p^{1/2}/\bar{R} \tag{9a}$$

$$\bar{K} = p/(\pi \bar{R}) = (p^{1/2}/\pi)\ \bar{\alpha} \tag{9b}$$

and, from (6)

$$\bar{R}_{OPT} = \left(\frac{p}{\pi}\right)^{1/2} \left(\frac{t_F}{t_R}\right)^{1/6} N^{-1/6} \tag{9c}$$

The factor of 2π in the definition of $\bar{K}$, (8c), is included since $\bar{K}$ then corresponds to the "integer cutoff" i.e the maximum value of $(l^2+m^2+n^2)^{1/2}$.

The precision required in the Ewald sum depends on the purpose for which it is being used. A lattice energy minimisation program may require greater accuracy than a molecular dynamics program. For the sake of argument let us follow [8] and assume the value $p = \pi^2$. This gives an accuracy of $\exp(-p) = 5.2 \times 10^{-5}$, which should be adequate for most purposes. With this choice we have, from (9a) and (9b), $\bar{\alpha} = \pi/\bar{R}$ and $\bar{K} = \bar{\alpha}$.

On the basis of operation counts alone we would expect $t_F \approx 2t_R$, though this is obviously very hardware and software dependent. We also need to take into account the efficiency with which we can locate neighbours in the real-space part of the calculation, which is discussed further in the next section. In order to have a concrete example, let us assume $t_F = t_R$. Combining with the suggested value for p we then have

$$\bar{R}_{OPT} = \pi^{1/2}N^{-1/6}; \quad \bar{\alpha} = \bar{K} = \pi^{1/2}N^{1/6}$$

There is a problem in optimising with small systems. In most molecular dynamics programs (but not lattice energy programs) the maximum real-space cutoff which can be used is half the box length. This is because the nearest-image convention is used in locating neighbours in nearby cells. The optimal value of $\bar{R}$ is greater than 0.5 for small systems. With our example parameters it only becomes less than 0.5 when N reaches about 2000. Simulations on smaller systems must therefore use the non-optimal value of 0.5 for $\bar{R}$. With the example parameters this corresponds to $\bar{\alpha} = \bar{K} = 2\pi = 6.28$. These values are close to the values which have been adopted in most simulations to date.

3 PROGRAM OPTIMISATION

We now discuss various computational aspects of the Ewald sum. The aim of this work was to develop a test program to calculate as efficiently as possible the energy of an arbitrary configuration of charged particles. The program was developed on a RISC super-scalar workstation (HP 750) and has also been run on an i860 processor.

In the real-space sum we need to locate all pairs of ions which are separated by less than the cutoff distance R. An all-pairs neighbour search is ruled out because of its N^2 performance, and we need to use a cell-based neighbour search method such as link-cells. In the link-cell method the computational box is divided into sub-cells, and a list made of the indices of the particles in each cell. This is in the form of a linked list: hence the name link-cell. The search for neighbours of particles in a particular cell can then be limited to those nearby cells whose closest distance of approach to the central cell is within the cutoff distance. In the conventional version, which goes back twenty years to a paper by Hockney, Goel and Eastwood [10], the cells have side equal to R (or a little larger, as necessary to fit a whole number of cells along the side of the computational box). In this case the search is limited to the central cell and the 26 touching cells.

This conventional version is not particularly efficient at finding neighbours. The volume searched, $27R^3$ is 6.45 times the actual volume, $4\pi R^3/3$, containing neighbours. The efficiency can be improved by taking more, but smaller, cells so that the volume of those included in the list of nearby cells can more nearly approximate to the cutoff sphere. It is not widely known that this idea also goes back to 1973. Quentrec and Brot [11] had a method in which they took cells so small that they could only include zero or one particles. However, the use of very small cells is not ideal because of the overheads of dealing with very short loops, or empty cells. Accordingly the test program used the link cell method but modified it to employ cells of side R/M with M an integer which can be varied to optimise the execution time. The list of nearby cells which may contain neighbours includes 13 cells when $M = 1$, 62 when $M = 2$, 155 when $M = 3$, 306 when $M = 4$ and 507 when $M = 5$. (These numbers take into account the fact that each pair needs to be considered only once, and exclude the central cell.) In general it was found that the program achieved optimal speed when the number of particles per cell was about 4. Only after doing these tests did the author find another 1973 paper, this time by Schofield [12], which came to exactly the same conclusion. Truly a vintage year for neighbour searching!

There is another advantage in the use of small link cells. The conventional link-cell method, with link cells of side R, only become more efficient than a simple all-pairs neighbour search when there are at least 4 link cells along the side of the box; otherwise the list of neighbour cells covers the whole computational box. This implies an $\bar{R}$ of 0.25 or less. With the example parameters given in the previous section this value is optimal only when $N \approx 127000$. In the present method, if there are at least $2M + 2$ cells along the box side the list of nearby cells excludes some of the cells in the box, and the link-cell neighbour search is more efficient than an all-pairs search. This condition is met provided

$$\bar{R} < \frac{0.5}{(1 + M^{-1})} \tag{10}$$

Larger values of M give some advantage over all-pairs search for the larger values of $\bar{R}$, approaching 0.5, which are optimal for more modestly sized systems.

In a molecular dynamics simulation the efficiency of neighbour search can always be improved by using the neighbour search algorithm to set up a neighbour list which is revised every few timesteps, rather than using the algorithm to locate neighbours afresh on every timestep. However, there is a penalty in terms of memory. The number of neighbours of each particle depends on the volume in the cutoff sphere, and varies as R^3, or $N^{1/2}$ when the parameters are optimised, see equation (6). Thus the number of entries in the list scales as $N^{3/2}$, and this may make the use of a neighbour list unacceptable in terms of its memory requirement.

When using a link-cell method the link-cell information can be used to identify those pairs of particles to which the nearest-image transformation needs to be applied. This proved to be particularly valuable on the HP750 since a commonly-used coding trick, which relies on the use of the nearest-integer function, proved to be very slow on this machine. The complementary error-function was evaluated by means of linear interpolation from a look-up table of 1000 points. This was about twice as fast as the use of a polynomial-exponential approximation [1], though the latter may be preferable on many machines since it vectorises.

The reciprocal space sum is more straightforward to optimise. The major effort is expended in the evaluation of the terms of the form $\exp(i\mathbf{k}.\mathbf{r}_j)$. The method adopted precomputes the components of these factors by recursion, and stores them [1]. This avoids calling the complex exponential function, but can involve substantial amounts of memory. The amount of storage involved is $(5\bar{K} + 4)N$ complex words, and hence varies as $N^{7/6}$ when performance is optimised. A compact version which computes these factors as they are needed was about four times slower on the HP, and about twice as slow as the non-compact version on the i860.

4 DISCUSSION AND TIMING TESTS

The arguments given in Section 2 concerning the choice of parameters were based simply on the magnitudes of the individual terms in the sums, and are therefore valid for arbitrary charge distributions. But the distributions found in molecular simulations have very special characteristics: in particular they observe approximate uniform density and local charge neutrality. Cancellations arising from local charge neutrality may well mean that in real systems the sums converge more rapidly. Many simulators have noticed that the reciprocal-space sum tends to be rather small when there are a large number of ions. In special cases the sum may be very small indeed: this is particularly noticeable with the ideal rocksalt structure, in which the reciprocal space energy is effectively zero for systems larger than 1000 ions. In general it may be possible to take fewer terms than the general analysis would suggest. If reducing cutoffs to save computer time, it is essential to use a configuration that is typical of the simulation to be performed: the ability to reproduce the rocksalt Madelung constant is not an adequate test.

Reference [8] has gone so far as to suggest that it is possible to omit the reciprocal-space sum entirely in the simulation of large systems. However, this is unlikely, since equation (9) shows that, optimally, the "integer cut-off" in reciprocal space increases as $N^{1/6}$, and the number of points within the cutoff therefore *increases* as $N^{1/2}$. It is amusing to note that there is, in fact, a valid argument for dropping the *real-space* term from the Ewald sum. By taking a large enough value of α, and hence a very short cutoff, it is possible to ensure that there are *no* neighbours within the cutoff, because of the finite size of the ions. This is of no practical use in molecular simulation, because we need a real-space sum anyway for the non-Coulombic interactions.

To explore these issues a test configuration of 4096 ions was created by distributing equal numbers of positive and negative charges randomly throughout the box. Though this technique will observe approximate local charge neutrality, the density of course will be far from uniform. This can therefore be regarded as a "worse case" test. On the HP 750 using the optimised program as described in the previous section the value of t_F was 2.0 μs. For t_R the best value which could be achieved was 2.6 μs, though this is rather variable depending on the efficiency of neighbour search which can be achieved with the actual link cell size used. Taking our example accuracy parameter $p = \pi^2$, equation (9c) then suggests the optimal choice $\bar{R} = 0.4244$. However, this immediately gives a problem as this value of $\bar{R}$ is too close to 0.5 for the link-cell algorithm to gain over a simple all-pairs search. $\bar{R}$ was therefore reduced to $0.5/(1 + 0.2) = 0.4167$ which, according to equation (10), is the largest value gaining some advantage when the value of the parameter M is taken to be 5. With these values of $\bar{R}$ and M there are 12 link cells along the side of the box, and an average of 2.37 particles per cell. From (9a) and (9b) the appropriate choice of the other parameters is $\bar{\alpha} = \bar{K} = 7.541$. The execution time for the program was then 12.8 s, made up of 5.5 s in the real-space sum, and 7.3 s in the reciprocal space sum. The reciprocal-space sum contributed about 11% of the total energy. The desired accuracy of the ES, corresponding to the chosen value of p, was 5.2×10^{-5}: the actual relative error in the energy was much better, 1.7×10^{-7}.

The next step was to investigate the scope for reducing the cutoffs and so improve the performance of the program. This was done in a series of stages:

at each stage changes were made until the relative error in the energy approached 5×10^{-5}.

(a) $\bar{K}$ was reduced keeping $\bar{R}$ and $\bar{\alpha}$ fixed. The lowest acceptable value of $\bar{K}$ was 5.7: with this value the reciprocal-space time had been reduced from 7.3 s to 3.3 s.

(b) $\bar{R}$ was reduced, keeping $\bar{\alpha}$ and $\bar{K}$ fixed. Reducing $\bar{R}$ gave scope for varying M, and this was done for each value of $\bar{R}$ in order to optimise the execution time. Surprisingly small values of $\bar{R}$ still gave acceptable errors: the final value was $\bar{R} = 0.29$, which gave a real-space time of 2.5 s (compared with the original value of 5.5 s) with $M = 3$. The optimisation of link-cell side is worthwhile, since the times with $M = 2$ and $M = 4$ were 3.8 s and 2.9 s respectively. With these values of $\bar{R}$ and M there are 10 link cells along the box side, and hence an average of 4.096 particles per cell.

(c) The smaller value of $\bar{R}$ should give scope for a reduction in $\bar{\alpha}$ and a consequent further reduction in $\bar{K}$. It was found that $\bar{\alpha}$ could be reduced to 6.6, and $\bar{K}$ to 5.1. With this value of $\bar{K}$ there was a further reduction in the reciprocal-space time to 2.7 s.

These results show that the analysis of Section 2 is too pessimistic about the cutoffs to be used for optimal performance, when real configurations are being studied. By reducing cutoffs it was possible, without introducing significant errors, to reduce the overall execution time from 12.8 s to 5.2 s. A rather similar situation arises in FMM. To be sure of achieving an accuracy of 5×10^{-5} for an arbitrary configuration requires taking the multipole expansion to order 14, whereas in practice order 4 (octupole) seems to be adequate.

References

[1] D. Fincham, "Programs for the dynamic simulation of liquids and solids: II", *Comput. Phys. Commun.*, **25** 159 (1982).

[2] J.W. Eastwood, R.W. Hockney and D. Lawrence, "P3M3D – the three dimensional periodic particle-particle/particle-mesh program", *Comput. Phys. Commun.*, **19** 215 (1980).

[3] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations", *J. Computat. Phys.*, **73** 325 (1987).
L. Grengard and V. Rokhlin, "The rapid evaluation of potential fields in three dimensions", *Lecture Notes in Mathematics*, **1360** 121 (1988).

[4] H-Q Ding, N. Karasawa and W.A Goddard, "Atomic level simulations on a million particles: the cell-multipole method for Coulomb and London non-bond interactions", *J. Chem. Phys.*, **97** 4309 (1992).

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

[6] H-Q Ding, N. Karasawa and W.A Goddard, "The reduced-cell multipole method for Coulomb interactions in periodic systems with million-atom unit cells", *Chem. Phys. Lett.*, **196** 6 (1992).

[7] D. Solvason, J. Kolafa, J.G. Petersen and J.W. Perram, "A rigorous comparison of the Ewald method and the Greengard-Rokhlin method", *J. Computat. Phys.* (submitted).

[8] Z.A. Rycerz, "Calculation of the Coulomb interactions in condensed matter simulations", *Molec. Simulation* **9** 327–349 (1992).

[9] J. Kolafa and J.W. Perram, "Cutoff errors in the Ewald summation formulae for point charge systems", *Molec. Simulation* **9** 351–368 (1992).

[10] R.W. Hockney, S.P. Goel and J.W. Eastwood, "A 10000 particle molecular dynamics model with long range forces", *Chem. Phys. Lett.* **21** 589 (1973).

[11] B. Quentrec and C. Brot, "New method for searching for neighbours in molecular dynamics computations", *J. Computat. Phys.* **13** 430 (1973).

[12] P. Schofield, "Computer simulation studies of the liquid state", *Comput. Phys. Commun.* **5** 17 (1973).