OPTIMISATION OF THE EWALD SUM

David Fincham, SERC Daresbury laboratory and Keele University

Introduction

It is part of the folk law of simulation that the execution time for the Ewald sum scales with N, the number of ions, as $N^{3/2}$. This result has appeared in print in various forms, e.g [1]. However, in view of a recent paper [2] which claims it is order N, I thought it worthwhile to go over the argument here, and bring out some points of practical importance.

We will consider 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 $N^{1/3}$.

Specification of parameters

The real space part of the system energy involves a sum over ion pairs of the form

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

which depends on the complementary error function $\mathrm{erfc}(\alpha r)$ 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 Ewald sum 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

$$\sum_{\mathbf{k}} k^{-2} \exp(-k^2/4\alpha^2)\ \Big|\sum_i q_i \exp(i\,\mathbf{k.r})\Big|^2$$

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).

Thus these two equations 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.

Evaluation of execution time

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

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

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

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

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 $\mathbf{k} = \frac{2\pi}{L}(l, m, n)$ 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 dependance 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

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

The above derivation is rough and ready. We have considered only the energy, whereas in molecular dynamics one is more 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 factor k^{-2} in the reciprocal space terms, which increases the rate of convergence somewhat. A more thorough analysis can be found in [3], 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 determine the N dependence of the sum when optimised.

Optimisation

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.

Discussion

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

$$\underline{R} = R/L \qquad (8a)$$

$$\underline{\alpha} = \alpha L \qquad (8b)$$

$$K = KL/2\pi \qquad (8c)$$

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

$$\underline{\alpha} = p^{1/2}/\underline{R} \qquad (9a)$$

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

and

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

The factor of 2π in the definition of K, (8c), is included since K then corresponds to the "integer cutoff" i.e the maximum value of $(l^2+m^2+n^2)^{½}$.

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 [3] and assume the value $p = \pi^2$. This gives an accuracy of $\exp[-p] = 5.2\times10^{-5}$, which should be adequate for most purposes. With this choice we have, from (9a) and (9b), $\alpha = \pi/R$ and $K = \alpha$.

On the basis of operation counts alone we would expect $t_F \approx 2t_R$, though this is obviously very hardware and software dependent. However, we also need to take into account the efficiency with which we can locate neighbours in the real-space part of the calculation. In the case of a solid system, and with the use of a neighbour list, this efficiency will be very high. For a liquid system we obviously need to use an order N neighbour-search algorithm, such as the link-cell method, but this is still not particularly efficient in locating neighbours, unless it is also used in conjunction with a neighbour list. Again, in order to have a concrete example, let us assume $t_F = t_R$. Combining with the suggested value for p we then have

$$R_{OPT} = \pi^{1/2}N^{-1/6} \qquad \bar{\alpha} = \bar{K} = \pi^{1/2}N^{1/6}$$

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 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 R. With the example parameters this corresponds to $\alpha = K = 2\pi = 6.28$. These values are close to the values which have been adopted in most simulations to date.

There is another effect to be considered in the simulation of small systems. Conventional link-cell programs use cells whose side is equal to or greater than the cutoff. They only become more efficient than a simple all-pairs neighbour search when there are at least four link cells along each side of the computational box. This implies an R of 1/4 or less. With our parameters this condition is reached only when $N \approx 127000$! The situation can be improved by using a variant of the linkcell method which uses smaller cells, as discussed below. However, the fact remains that with smaller systems one may be forced to use a less efficient neighbour search method, or else a smaller and hence non-optimal value of R.

We may also need to consider memory requirements. Most Ewald sum programs, in their reciprocal space routine, precompute and store factors of the form $\exp(i 2\pi l x_i / L)$, which can involve substantial amounts of memory. The number of these factors is proportional to NK, and hence varies as $N^{7/6}$ when performance is optimised. A compact version of the Ewald sum which computes these factors as they are needed can be several times slower. Furthermore, efficient neighbour location really needs a neighbour list, which again uses a large amount of memory. In this case the number of neighbours of each particle depends on R^3, and the memory requirement scales as $N^{3/2}$.

Size of link cells

In the link-cell method the computational box is divided into sub-cells, and the search for neighbours of particles in a particular cell is 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 [4], the cells have side equal to R, and 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, $\frac{4}{3}\pi R^3$, 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 [5] 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. In experiments to minimise the computer time by varying the link cell side I found that the best size of link cell to use is one which contains about 4 particles. Only after doing these tests did I find another 1973 paper, this time by Schofield [6], which came to exactly the same conclusion. Truly a vintage year for neighbour searching!

Do we need the reciprocal space sum?

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 $N^{1/2}$. In practice, as we saw in the previous section, for all but the very largest of systems we may wish to reduce the real space cutoff below the optimal value to enable the use of an efficient neighbour search algorithm. This will require a compensating increase in α and hence even more terms in the reciprocal-space sum. The suggestion in [2] that we can omit the reciprocal-space sum in the simulation of large systems thus seems unlikely. However, most simulators have noticed that the reciprocal-space sum is usually small, particularly in the case of large systems, and there are reasons for thinking it may be less important than the above analysis suggests.

The arguments above have been 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. It is because of cancellations arising from local charge neutrality that the sum tends to be rather small. 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 well be possible to take fewer terms than the general analysis would suggest. If reducing cutoffs to save computer time, the best technique is still to confirm the accuracy of the Ewald sum by checking that energies are independent of α. When doing this, it is essential to use a configuration that is typical of the simulation to be performed: do not rely on reproducing the rocksalt Madelung constant!

It is amusing to note that there is 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 dynamics, because we need a real-space sum anyway for the non-Coulombic interactions.

Example timings

My program for these timing tests creates a configuration 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. The program calculates energies only, not forces. The real space sum uses a link cell method, but based on link cells which have sides which can be chosen to be various submultiples of the cutoff. I chose to test on a system of 4096 ions. Running on an HP 750 workstation I found the value of t_F to be 1.9μs. For t_R, by optimising the link cell side, I achieved a value of 5.2μs. Taking the above values, and $p = \pi^2$, we find $R = 0.377$, and hence $\alpha = K = 8.33$.

The execution time for the program was then 19.6 s, made up of 9.9 s in the real space sum, and 9.7 s in the reciprocal space sum. The link cell side was just over $0.25R$, giving an average of 4.1 particles per cell. The reciprocal space sum provided about 15% of the energy, and certainly cannot be neglected entirely. However, reducing K to 6 reduced the reciprocal space time to 3.8 s, and gave an error in the total energy of about 0.05%. This demonstrates that in real problems it may be possible to make considerable savings in computer time, without significant loss in accuracy, by reducing the reciprocal space cut off.

Achieving the real-space times above took some effort. The reason lies in the treatment of the nearest-image transformation. My original code used the popular coding trick

```
rx = rx - anint (rx)
```

etc., but this gave an execution time of 61.6 s. Changing this to

```
rx = rx - nint (rx)
```

and so allowing the compiler rather than the intrinsic function to convert from integer to real, reduced the time to 33.7 s, and almost the same time resulted if the "nint trick" was abandoned and a double IF statement used instead. The much faster time of 9.9 s was obtained by using the link cell information to determine when the nearest-image transformation needs to be applied. The "nint trick" works better on other computers.

By removing the nearest-image transformation and spherical cutoff I was able to determine that the actual evaluation time for a single real-space interaction is about 4.0μs. This shows that the neighbour search and nearest-image transformation still impose a 30% overhead.

REFERENCES

[1] 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 (1988) 875-893.

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

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

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

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

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