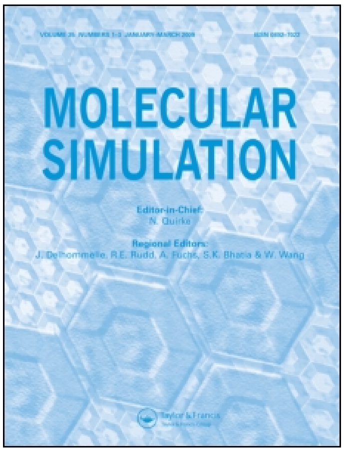

Molecular Simulation

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

On Multiple Time-step Algorithms and the Ewald Sum

T. Forester[a]; W. Smith[a]
[a] S.E.R.C. Daresbury Laboratory, Warrington, England

To cite this Article Forester, T. and Smith, W.(1994) 'On Multiple Time-step Algorithms and the Ewald Sum', Molecular Simulation, 13: 3, 195 — 204

To link to this Article: DOI: 10.1080/08927029408021983

URL: http://dx.doi.org/10.1080/08927029408021983

Molecular Simulation, 1994, Vol. 13, pp. 195–204
Reprints available directly from the publisher
Photocopying permitted by license only

ON MULTIPLE TIME-STEP ALGORITHMS AND THE EWALD SUM

T. FORESTER and W. SMITH

S.E.R.C. Daresbury Laboratory, Daresbury, Warrington WA4 4AD, England

(Received June 1993, accepted February 1994)

We show how standard multiple time-step algorithms devised for systems with short-range potentials can be used successfully in simulations of periodic systems with long-range (Coulombic) potentials. Three strategies for incorporating the Ewald sum into a multiple time-step algorithm are considered. These are (i) evaluation of reciprocal space terms every time-step (ii) evaluating reciprocal space terms once every n time-steps and placing these terms in with the slowly varying forces and energies (iii) a modified form of the second strategy in which primary shell (close) electrostatic interactions are evaluated directly and the more distant interactions handled by the Ewald sum (once every n time-steps). Only the first and third approaches give satisfactory thermodynamic results. The third strategy is much more efficient than the first. With the third strategy substantial savings in cpu time are acheived in both the real space and, most importantly, the reciprocal space terms of the Ewald sum. This is achieved without significant loss of accuracy or stability. Overall execution time is decreased by a factor of between 2 and 3.

KEY WORDS: Molecular Dynamics Simulation, Ewald sum, multiple time-step

INTRODUCTION

The most computationally demanding part of a molecular dynamics (MD) simulation is the evaluation of the interaction forces and energies. Normally, when dealing with non-bonded forces, the potential form is restricted to pairwise additive terms. All pair interactions whose separations are less than some cutoff radius, r_{cut}, are evaluated every time-step. Typically this evaluation of energies and forces accounts for over 90% of the cpu time. This ultimately places a limit on the size of system tractable for simulation. There is thus great interest in schemes which increase the efficiency of energy and force evaluation. One way to do this is with a "multiple time-step" algorithm. In such a scheme the interactions to be evaluated are separated into two classes: those within a "primary" shell (with separation less than $r_{prim} < r_{cut}$), and those outside the primary shell but with separations less than r_{cut}. The primary interactions are evaluated every time-step while the "secondary" interactions are evaluated only once every n time-steps. The rationale behind this is that on a short timescale, the size of the fluctuations in the secondary forces and energy are considerably smaller than those arising from the primary interactions. Various levels of sophistication exist for adding the secondary contributions to the primary ones: these include treating the secondary contributions as constant in between evaluations [1], self-correcting schemes [2], and using spatial derivatives of secondary force and energy to improve estimates during the primary time-steps [3]. However the discussions that follow are not specific to any particular multiple time-step algorithm.

Multiple time-step algorithms work well when the interaction potential is short-ranged. As for example when a Lennard-Jones potential is used:

$$E_{ij} = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \tag{1}$$

r_{ij} is the separation between sites i and j and the parameters ε and σ have their usual meanings. With this sort of potential r_{prim} is usually set around 1.2σ, and r_{cut} around 2.5σ. The secondary interactions are typically evaluated once every 5 to 10 time-steps and energy conservation over the course of a simulation is comparable to that obtained with a single time-step approach. When a Verlet neighbourhood list is used, the multiple time-step approach can speed up overall execution time by a factor of 3 to 5 [4].

However, if a *long*-range potential is used, such as the Coulombic potential:

$$E_{ij} = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{r_{ij}} \tag{2}$$

(where ε_0 is the permittivity of free space, and q_j the charge on the jth site) the multiple time-step method is not so readily applicable because the fluctuations in the forces and energy from secondary interactions are comparable with those arising from primary interactions. If we choose a suitably large primary cutoff, we expect that fluctuations from the secondary interactions will be small enough to allow a multiple time-step algorithm to be used. However, it should be apparent that the primary cutoff will need to be larger than for a short-range potential. Consequently some of the anticipated gain in efficiency will be lost.

The situation becomes more complex when the Coulombic potential is to be evaluated in a periodic system. In such systems the Coulombic potential is only conditionally convergent (ie. the interaction energy ultimately depends upon the order in which we add up the pair interactions). However, the evaluation can be efficiently and accurately handled by the Ewald summation [5]. The main point of this communication is to show how the separation of primary and secondary terms can take place within the framework of the Ewald sum and to investigate the accuracy and efficiency of this approach in MD simulations.

The Ewald sum splits contributions to the Coulombic potential into three parts. First there is a damped real space term

$$E_{real} = \frac{1}{4\pi\varepsilon_0}\sum_{r_{ij}<r_{cut}}\frac{q_i q_j}{r_{ij}}\,erfc(\zeta r_{ij}) \tag{3}$$

where *erfc* is the complementary error function: $erfc(x) = 1 - erf(x)$. The convergence parameter ζ is chosen so that pairs separated by more than r_{cut} contribute a negligible amount to the sum and are thus ignored.

Second is a reciprocal space contribution

$$E_k = \frac{1}{2V\varepsilon_0}\sum_{k'}^{\infty}\exp(-k^2/4\zeta^2)/k^2\left|\sum_{j}^{N} q_{ij}\exp(-i\mathbf{k}.\mathbf{r}_j)\right|^2 \tag{4}$$

where $\mathbf{k}$ are reciprocal space lattice vectors and the prime indicates that the $\mathbf{k} = \mathbf{0}$ term is omitted from the sum. V is the volume of the replicated MD cell and N is the number of sites in the cell. In practice, the series is summed spherically and truncated after sufficient **k** vectors have been included in the sum.

The third contribution is a "self-interaction" term. It is a correction term to the reciprocal space term. The reciprocal space sum includes all pair interactions in the system. This implicitly includes the interactions of a site with itself and pairs of charge sites on the same molecule that are mutually involved in chemical bonds or valence angle interactions. All these terms need to be subtracted from E_k if the correct interaction energy is to be obtained. These terms are collected together in the "self-interaction energy", E_{self}:

$$E_{self} = \frac{\zeta}{4\pi^{3/2}\varepsilon_0}\sum_j^N q_j^2 + \frac{1}{4\pi\varepsilon_0}\sum_{molecules}\sum_{\alpha<\beta}\frac{q_\alpha q_\beta}{r_{\alpha\beta}}\, erf(\zeta r_{\alpha\beta}) \tag{5}$$

where α and β label sites on the same molecule involved in a mutual chemical bond or valence angle. The correct reciprocal space contribution can then be written as

$$E_{recip} = E_k - E_{self} \tag{6}$$

and the total Coulombic energy for the periodic system is given by

$$E_{Coul} = E_{real} + E_{recip} \tag{7}$$

To incorporate the evaluation of Coulombic energy and forces into a multiple time-step algorithm there are at least three strategies that we can adopt. The most straightforward strategy (S1) is to view the real space term as short-ranged and treat it by standard methods. The consequence is that reciprocal space terms still need to be evaluated every time-step. While perfectly correct this approach is not entirely satisfactory. Evaluation of E_{recip} can take a comparable amount of time to, if not more than, evaluation of E_{real}. (The exact proportioning is machine dependent and also depends on the choice of real and reciprocal space cutoffs). In having to calculate E_{recip} at each time-step the speed up in overall execution time is still expected to be significant but it is not expected to be dramatic.

In the second and third strategies we aim for schemes that require evaluation of the reciprocal space contributions only once every n time-steps. The "naive" strategy (S2) is to partition the real space terms between primary and secondary interactions and treat all reciprocal space contributions as secondary terms (evaluated once every n time-steps). Thus for the primary Coulombic energy we have

$$E_{prim} = \frac{1}{4\pi\varepsilon_0}\sum_{r_{ij}<r_{prim}}\frac{q_i q_j}{r_{ij}}\, erfc(\zeta r_{ij}) \tag{8}$$

As shall be seen this approach is not satisfactory because a non-negligible contribution to the actual primary interactions is contained in the reciprocal space terms. Consequently, classifying these primary-shell reciprocal-space terms as "secondary" results in poor energy conservation.

The third strategy, S3, is to seek a rigorous separation of "primary" interactions from all others in the system. The primary interactions are evaluated directly and the remaining interactions evaluated via the Ewald sum. To do this we proceed as follows: the Coulombic energy arising from primary interactions, E_{prim}, is evaluated every time-step from

$$E_{prim} = \frac{1}{4\pi\varepsilon_0} \sum_{r_{ij} < r_{prim}} \frac{q_i q_j}{r_{ij}} \tag{9}$$

where the sum includes all interactions with $r_{ij} < r_{prim}$ at the time the interactions are classified as "primary" or "secondary". Note that once an interaction has been classified as "primary" or "secondary" it must maintain that classification until the next time the lists are updated (n time-steps later).

The secondary interactions are evaluated once every n time-steps by the Ewald sum. Only the secondary interactions contribute to the real space term of the Ewald sum. We label this term as E^{sec}_{real}:

$$E^{sec}_{real} = \frac{1}{4\pi\varepsilon_0} \sum_{r_{prim} \le r_{ij} < r_{cut}} \frac{q_i q_j}{r_{ij}} erfc(\zeta r_{ij}) \tag{10}$$

The reciprocal space term now consists of E_{recip} (6) less a term, E^{prim}_{recip}, that takes out the contributions to E_{recip} that arise from primary interactions. These primary interactions are implicitly present in the **k** space sum of (4). The addition term is given by:

$$E^{prim}_{recip} = \frac{1}{4\pi\varepsilon_0} \sum_{r_{ij} < r_{prim}} \frac{q_i q_j}{r_{ij}} erfc(\zeta r_{ij}). \tag{11}$$

The modified reciprocal space contribution to the total Coulombic energy is:

$$E^{sec}_{recip} = E_{recip} - E^{prim}_{recip}. \tag{12}$$

The secondary contributions to the Coulombic energy can then be written as

$$E_{sec} = E^{sec}_{real} + E^{sec}_{recip} \tag{13}$$

and the total Coulombic energy as

$$E_{Coul} = E_{prim} + E_{sec}. \tag{14}$$

In implementing S3 we need to evaluate E_{prim} every time-step but E^{prim}_{recip} only when E^{sec}_{recip} is evaluated. On time-steps when both E_{prim} and E^{prim}_{recip} are required they can be most efficiently evaluated within the same segment of code.

Strategies S1 and S2 are easy to implement in an MD code, S3 is only slightly more complex. We now investigate the performance of these schemes in MD simulations.

THE TEST SYSTEMS

Liquid water provides a stringent test of the multiple-time-step strategies. It has a low rotational inertia and a large dipole moment. For oxygen-oxygen separations greater than about 4 Å the intermolecular potential is dominated by the electrostatic terms. The first test system was 256 SPC water molecules in a cubic box of length L = 19.6615 Å, r_{cut} = 8 Å, Δr (width of Verlet shell) = 1.83 Å and r_{prim} = 6 Å. The Ewald convergence parameter was $\zeta = 0.40$ Å^{-1} and the reciprocal space cut off was $k_{max} = 8(2\pi)/L$. Bond-lengths where held constant using SHAKE with a tolerance in d^2 of 10^{-8}.

Table 1 Thermodynamic results for Water (1 ps and 10 ps† simulations): $r_{cut} = 8\,\text{Å}$, $r_{prim} = 6\,\text{Å}$.

Algorithm	*n*	*cpu time (s step^{-1})*	*$\langle E_{Total} \rangle$ (kJ mol^{-1})*	*S.D.(E_{Total}) (kJ mol^{-1})*	*drift(E_{Total}) (kJ mol^{-1} ps^{-1})*	*$\langle T \rangle$ (K)*	*$\langle E_{Coul} \rangle$ (kJ mol^{-1})*
single step	1	3.18	−35.740	0.001	−0.004	300.0	−49.879
S1	4	3.12	−35.740	0.001	−0.004	301.2	−49.984
	6	3.02	−35.739	0.002	−0.002	299.4	−49.895
	8	2.98	−35.737	0.003	0.002	300.9	−49.820
	10	2.95	−35.734	0.004	0.008	299.8	−49.891
S2	4	2.16	−35.708	0.019	0.060	302.0	−50.012
	6	1.78	−35.559	0.100	0.358	304.9	−49.914
	8	1.60	−35.071	0.432	1.334	309.5	−49.940
	10	1.53	−34.021	1.055	3.433	326.6	−48.488
S3	4	2.20	−35.741	0.004	−0.007	299.6	−49.875
	6	1.69	−35.743	0.013	−0.010	302.2	−50.222
	8	1.61	−35.740	0.022	−0.004	299.4	−49.855
	10	1.51	−35.683	0.056	0.028	304.2	−50.043
S3†	6	1.69	−35.696	0.038	0.008	298.6	−49.770
	8	1.60	−35.327	0.194	0.082	302.2	−50.480

The second test case was an antifreeze protein immersed in 1234 SPC waters. The MD cell was a cube of length L = 35 Å with periodic boundary conditions. The real space cutoff was $r_{cut} = 12\,\text{Å}$ and the reciprocal space cutoff $k_{max} = 7(2\pi/L)$. The Ewald convergence parameter, ζ, was assigned a value of $0.26\,\text{Å}^{-1}$. In both test cases the electrostatic energy was converged to the fifth significant figure.

All simulations were carried out in the microcanonical (NVE) ensemble in which the total energy (E_{Total}) of the system is conserved. The Verlet leapfrog integration scheme was used in conjunction with the Teleman-Jönsson multiple time-step algorithm [1]. The reliability of a simulation can be gauged from the standard deviation (S.D.) and drift in E_{Total} over the course of the simulation. For the single time-step algorithm we found it necessary to use a time-step of 1 fs to get acceptable energy conservation. The single time-step simulation serves as the standard by which the multiple time-step simulations should be judged.

The program used was DL_POLY [6], a parallelized macromoleculer simulation package being developed at the Daresbury Laboratory. Calculations were performed on the Daresbury Laboratory Intel iPSC/860 hypercube operating on 8 nodes.

RESULTS AND DISCUSSION

The reliability of the multiple time-step strategies can be assessed via thermodynamic quantities obtained from the simulations. Results for the first test case are given in table 1.

S1 is expected to be the most accurate (and expensive) of the three multiple-time-step strategies. It is well behaved (table 1) but for small *n* it is not significantly better than S3. Both these strategies produce good agreement with the single time-step simulation (1 ps) for values of *n* up to about 8 after which degradation in energy conservation becomes apparent. The differences between results for S1 and S3

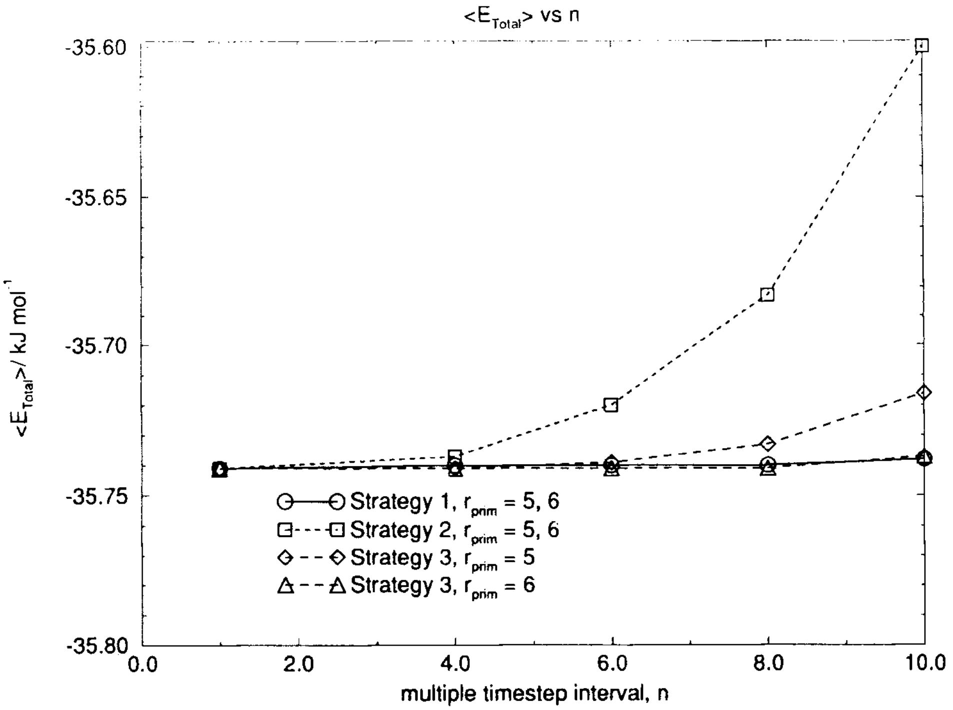

Figure 1 Average total energy in the water test case. The data are from runs of 100 time-steps at 300 K. See text for details.

are that for S3 the trends are less systematic and the standard deviation in E_{Total} is noticeably greater (but still acceptable). These results indicate that while S3 generates somewhat larger random errors in the instantaneous forces and energies, the systematic errors seem to be on a par with those from S1.

In contrast, simulations using S2 showed severe energy drifts, systematic heating of the system and large standard deviations in the total energy. The results are unsatisfactory even at the $n = 4$ level and become systematically worse as n increases. The "naive" strategy, S2, is thus unsuitable for use in simulation.

We also investigated how sensitive the stability of the strategies were to choice of r_{prim}. Because S1 and S2 both make use of the rapidly decaying complementary error function in the primary electrostatic terms they should be relatively insensitive to the choice of r_{prim}. (Provided, of course, that the value of $erfc(\zeta r_{prim})$ is suitably small.) Figure 1 shows results obtained for $\langle E_{Total} \rangle$ for short runs on the water system (0.1 ps, 100 time-steps). For both S1 and S2 results obtained using $r_{prim} = 5$ Å ($erfc(\zeta r_{prim}) = 4.7 \times 10^{-3}$) were virtually indistinguishable from those obtained with $r_{prim} = 6$ Å ($erfc(\zeta r_{prim}) = 8.0 \times 10^{-4}$). The figure also indicates just how quickly S2 goes astray. S3 is more sensitive to choice of r_{prim}. Results for the short runs were considerably improved by increasing r_{prim} from 5 Å to 6 Å (figure 1). The $r_{prim} = 6$ Å results being almost identical to those obtained using S1. Extending the 6 Å runs to 1 ps (table 1) shows significant differences between S1 and S3 only at $n = 10$. Because the r.m.s. fluctuations for S3 are quite large it is necessary to further extend the runs in order to get a reliable estimate for the energy drift. We extended the $n = 6$ and $n = 8$ runs out to 10 ps (table 1). Only in the $n = 6$ run did the drift remain at an acceptable level. As there is only a 6% difference in execution time between the two runs, use of $n = 6$ is clearly preferrable to $n = 8$.

There seems little point in extending table 1 beyond its current range: S2 is obviously flawed; S1 will never give comparable performance to S3 (in terms of cpu time) and S3 shows instability at $n = 10$ and above.

The observation that S3 is well behaved while S2 is not points to the fact that *primary* interactions can make a considerable contribution to the *reciprocal* space terms in the Ewald sum. In the antifreeze protein system ($r_{prim} = 8\,\text{Å}$) we found primary-shell reciprocal-space terms to account for 4% of the *total* Coulombic energy of the system. Hence use of equation (8) to generate E_{prim} is erroneous unless E_{prim} is calculated every MD time-step. Contrary to recent claims that reciprocal space terms can be completely ignored [7], we find that proper treatment of them is crucial for reliable simulation.

Ultimately the factor that determines the stability of the multiple time-step strategies is how rapidly the secondary forces change with time. The time dependence of the secondary forces, $\mathbf{F}_{sec}$, can be assessed using the auto-correlation function formalism

$$C(t) = \langle \mathbf{F}_{sec}(0).\mathbf{F}_{sec}(t) \rangle / \langle \mathbf{F}_{sec}(0).\mathbf{F}_{sec}(0) \rangle \tag{15}$$

where the $\langle \ldots \rangle$ indicates an ensemble average over all particles in the system and all time origins in the simulation. A slow decay rate of the correlation function indicates the $\mathbf{F}_{sec}$ is relatively time invariant and that a correspondingly larger value of n may be used for multiple time-steps. Figure 2 shows these correlation functions for S2 and S3 in the water system. Data were generated using $r_{prim} = 6\,\text{Å}$ and a fixed primary and secondary neighbour list. The function obtained using S3 decays much more slowly than that obtained using S2. The S2 correlation function decays to 0.98 after just 4 fs. It takes four times longer for the S3 function to decay by the same amount. The rapid loss in time correlation of the secondary forces accounts for the instability of S2 and reinforces the necessity of properly accounting for the primary shell terms. It is worth noting that the r.m.s. magnitude of the secondary forces for S2 is about 10% *smaller* than that for S3. This highlights the point that it is the fluctuations in the secondary forces, and not their absolute magnitudes, that are of principal concern.

Having considered accuracy we now turn our attention to the efficiency of the algorithms. Given that S2 is unsuccessful we focus only on S1 and S3. The overall increase in efficiency of the program can be measured by the speed-up factor defined by:

$$f = \frac{\langle \textit{time for one MD step using single time step algorithm} \rangle}{\langle \textit{time for one MD step using multiple time step algorithm} \rangle} \tag{16}$$

where the angular brackets denote an average over a large number of time-steps. For the water test system S1 and S3 (at $n = 6$) have speed-up factors of $f = 1.05$ and $f = 1.88$ respectively. In the antifreeze protein simulation ($n = 5$), we obtained values of $f = 2.0$ and $f = 2.6$ for the two strategies. The greater efficiencies in the second series of simulations reflecting the larger ratio of r_{cut} to r_{prim} and the fewer

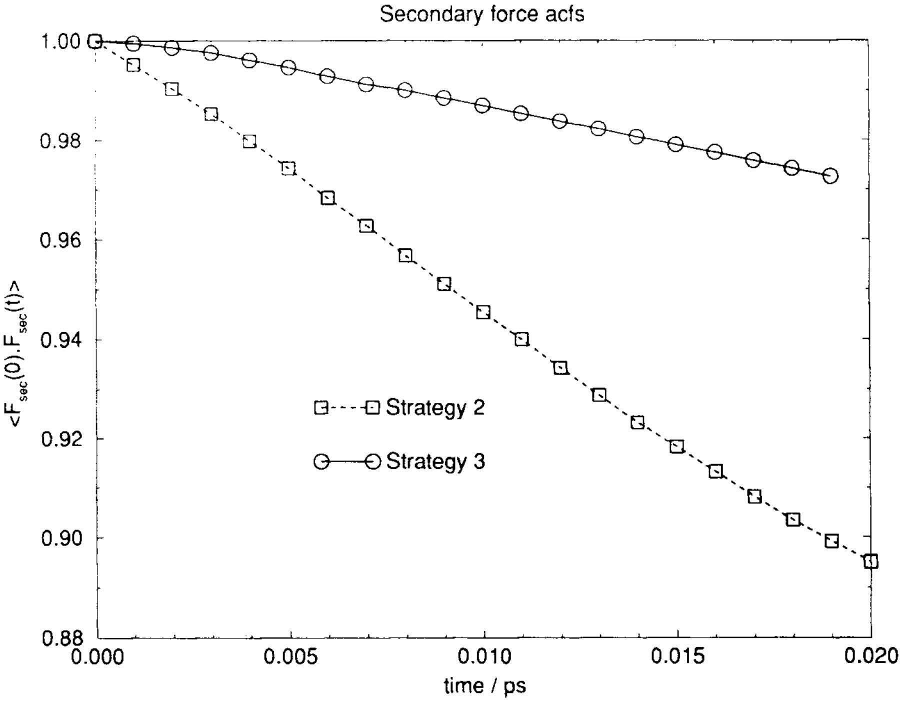

Figure 2 Normalised auto-correlation functions of the secondary forces for strategy 2 (S2) and strategy 3 (S3). Data obtained from the water test case with fixed secondary and primary neighbour lists.

reciprocal space vectors we needed to use. The speed up factors are somewhat lower than the figures quoted in the introduction for short-range potentials. This is principally because, the fraction r_{prim}/σ can be as low as 1.2 for a short-range potential. When electrostatic interactions are considered we found that, for water, this figure needed to be at least 1.8. In addition, in all the multiple time-step strategies the overall increase in efficiency is limited by computational tasks common to both multiple time-step and single time-step algorithms - notably the evaluation of bond constraints which took an average of 0.42 seconds per step in the water test case. None-the-less the overall increase in efficiency we have obtained with S3 represents a very significant saving in cpu time.

In general the speed-up factors may be enhanced by using a larger secondary time-step (ie, increasing n) and/or increasing (r_{cut}/r_{prim}). Optimal values are system dependent but it does appear necessary to use a larger primary cutoff and smaller n when Coulombic rather than short-ranged potentials are used. In general, a larger value of r_{prim} should permit a larger value of n to be employed but will mean primary interactions will take longer to evaluate since the number of such interactions is proportional to r^3_{prim}. A smaller value of r_{prim} will require a smaller value of n and incur computational overheads in the assignment of which interactions are primary and which are secondary. At one extreme one could set $r_{prim} = r_{cut}$ in the hope that a much larger value of n could be used. In this case all the saving in cpu would be due to the reduced number of evaluations of the reciprocal space

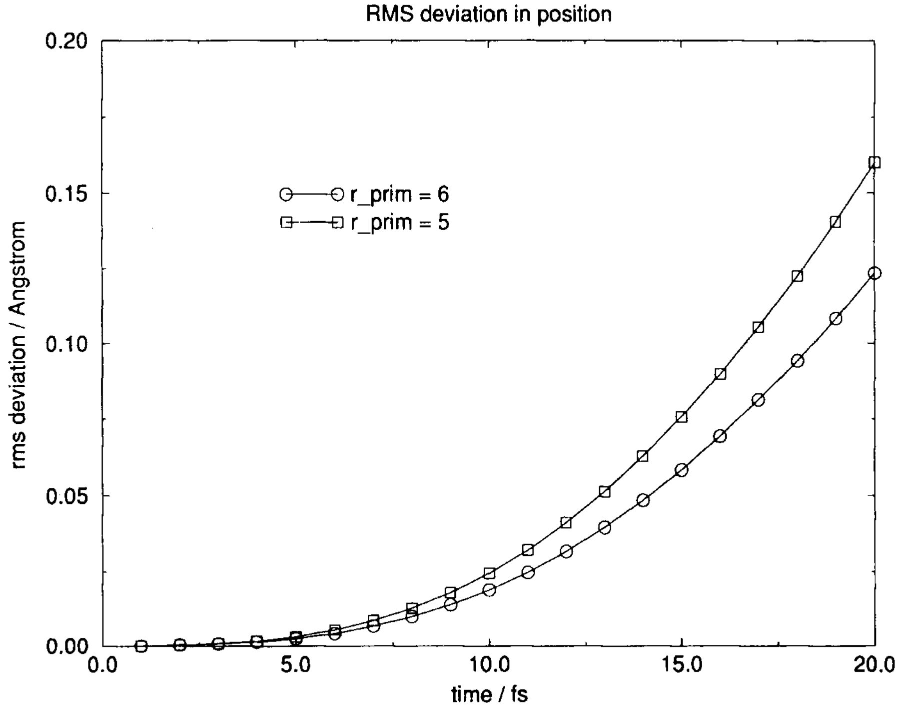

Figure 3 Root mean square deviation of atomic coordinates in the antifreeze protein test case. Data obtained for fixed secondary and primary neighbour lists for r_{prim} = 5 Å and r_{prim} = 6 Å.

terms. In the test cases this extreme did not turn out to be optimal. In the water test case use of r_{prim} = 8 Å did reduce the r.m.s fluctuations in the energy by about 30% compared to the values listed in table 1. However the associated energy drift was not significantly reduced so that one is still limited to $n = 8$ or less. Figure 3 shows the root mean square (r.m.s.) deviation, in units of Angstrom, of atomic coordinates in the antifreeze protein test system between a multiple time-step simulation (S3) and a single time-step simulation. The data generated in this figure were for $n = 20$ and r_{prim} = 5 Å and 6 Å. The figure indicates that the smaller r_{prim} the more rapid the divergence of the MD trajectories. Use of ($n = 5$, r_{prim} = 6 Å) corresponds to a r.m.s. deviation in positions of less than 0.005 Å at the update of secondary interactions. Increasing n to 8 doubles the r.m.s. deviation. This is still acceptable given the essentially chaotic nature of individual MD trajectories and that most of the r.m.s. deviation is due to hydrogen sites in the solvent. In this case a thermostat could be used to suppress the energy drift associated with S3. With $n = 8$ and r_{prim} = 6 Å the speed-up factors for the antifreeze protein simulation increase to $f = 2.3$ and $f = 3.0$ for S1 and S3 respectively.

CONCLUSIONS

We have demonstrated how the Ewald sum can be incorporated into a multiple time-step algorithm in an efficient and accurate way. Of the three strategies studied S1 is the most accurate (and the most expensive) for a given value of *n*. S3 is an excellent compromise between accuracy and efficiency. The "naive" strategy, S2, is of the same efficiency as S3 but is seriously flawed. The use of S3 can decrease the overall execution time by a factor of 2 to 3 compared with a single time-step algorithm. A saving in cpu time is achieved in both the real space and the reciprocal space parts of the Ewald sum without significant loss of accuracy or stability. The long-range nature of the electrostatic potential and large magnitude of the forces involved make it necessary to use a larger primary cutoff and smaller multiple time-step interval, *n*, than with purely short-range potentials.

Acknowledgements

We gratefully acknowledge the UK S.E.R.C. for financial support of this work under the Computational Science Initiative and we thank the referee for constructive comments on this work.

References

[1] O. Teleman and B. Jönsson, "Vectorizing a general purpose Molecular Dynaimcs simulation package", *J. Comp Chem*, 7, 58 (1986).
[2] M.E. Tuckerman, B.J. Berne and A. Rossi, "Molecular dynamics algorithm for multiple time scales: Systems with disparate masses", *J. Chem Phys.*, **94**, 1465 (1991).
[3] R.D. Swindoll and J.M. Haile, "A multiple time-step method for Molecular Dynamics simulations of fluids of chain molecules", *J. Comp Phys.*, **53**, 289 (1984).
[4] M.P. Allen and D.J. Tildeseley, *Computer Simulation of Liquids*, Oxford University Press, Oxford, 1989, pp. 152–155.
[5] *ibid.* pp. 156–162.
[6] The macromoleculer simulation package DL_POLY is currently being developed at this laboratory for release to the academic community in 1994. Interested readers should contact the authors.
[7] Z.A. Rycerz, "Calculation of the Coulomb interactions in condensed matter simulation", *Molecular Simulation*, **9**, 327 (1992).