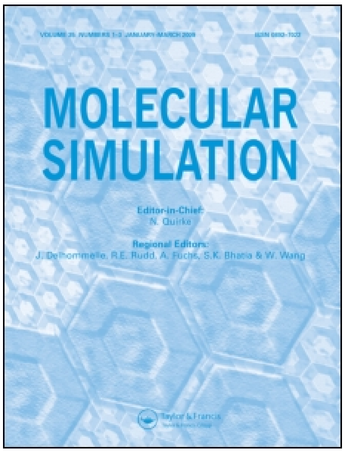

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

A Data-Parallel Molecular Dynamics Method for Liquids with Coulombic Interactions

Fredrik Hedman[a,b]; Aatto Laaksonen[a,b]
[a] Center for Parallel Computers (PDC), Stockholm, Sweden [b] Division of Physical Chemistry Arrhenius Laboratory, Stockhom, Sweden

To cite this Article Hedman, Fredrik and Laaksonen, Aatto(1995) 'A Data-Parallel Molecular Dynamics Method for Liquids with Coulombic Interactions', Molecular Simulation, 14: 4, 235 — 244

To link to this Article: DOI: 10.1080/08927029508022020

URL: http://dx.doi.org/10.1080/08927029508022020

Molecular Simulation, 1995, Vol. 14, pp. 235-244
Reprints available directly from the publisher
Photocopying permitted by license only

A DATA-PARALLEL MOLECULAR DYNAMICS METHOD FOR LIQUIDS WITH COULOMBIC INTERACTIONS

FREDRIK HEDMAN* and AATTO LAAKSONEN†

Center for Parallel Computers (PDC) KTH S-100 44 Stockholm, Sweden
Division of Physical Chemistry Arrhenius Laboratory, SU S-106 91 Stockhom, Sweden

(Received September 1993, accepted January 1995)

An approach to large-scale data-parallel molecular dynamics for simulations of systems which include Coulombic interactions is investigated. Short-range interactions are calculated with a method based on the use coarse-grained cells. In this method the simulation cell is decomposed into equally sized subcells, with the shortest side being larger than the cut-off radius of the short-range interaction. Electrostatic interactions are calculated using a data-parallel version of the Ewald summation method. Calculations of long- and short-range interactions are merged by a suitable choice of the size of the subcells and the Ewald sum convergence parameter.

The method, which should be easy to implement on most massively parallel computers of SIMD or MIMD type has been developed on a 8K CM200 using CM Fortran, and so far it has been tested on systems of over 2 million particles. Timings of systems up to 2 million particles are presented. The time per time-step for a full CM 200 is estimated to be around 15 seconds for a system of 256 thousand particles.

KEY WORDS: Data-parallel Molecules Dynamics, Coulombic interactions.

1 INTRODUCTION

The computer simulation of matter is an established area of science and a large and steadily expanding discipline of its own [1]. Methods, such as Monte Carlo (MC) and Molecular Dynamics (MD) can now routinely be employed on systems like pure liquids, solids and not too complex mixturs and solutions [1, 2]. Molecular computer simulations provide a link between theory and experiment and in many cases it is also the only way to study complex many-body systems, either because of non-existing experimental techniques or simply because too complex mathematical machinery often leads to a theoretical impasse.

Ever since the mid-sixties, when Rahman [3] reported an extensive MD simulation of argon, the power of classical MD techniques in studies of real systems has been apparent. Astonishingly simple molecular models together with no more than a couple of hundred particles have shown to give good descriptions of structure and dynamic behaviour in e.g. liquid systems. During the last thirty years computer simulations of

* Supported by the Swedish National Board for Industrial and Technical Development (NUTEK)
† Supported by the Swedish Natural Research Council (NFR)

increasingly complex many-body problems have been applied to numerous areas in chemistry, physics, biology and material sciences [4–7]. Several excellent textbooks have appeared guiding students and researchers through the world of statistical mechanical theories, numerical algorithms and applications of computer simulation methods [8–11].

In most MD simulations, pairwise interactions are assumed to be sufficient. This implies that the time per time-step in a simulation of N particles scales as $O(\mathrm{N}^2)$. Now, depending on the range of the interactions between the particles the required computing time can be considerably reduced by applying a suitable cut-off. This effectively sets interactions between particles separated by distances larger than the cut-off to zero. Calculating these interactions can thus be avoided as soon as one knows the particle separation. In fact, by employing the linked-cell method [12], where the simulation cell is decomposed into sub-cells with dimensions chosen to match the cut-off radius of the pairwise interaction, the search for interacting neighboring particles can be limited to the cell in which the particle is currently located and the touching cells. In a d-dimensional simulation the number of touching cells is $3^d - 1$. Assuming small variations in the particle density, the linked-cell method transforms the problem into an $O(N)$ problem.

Nevertheless, despite this result MD simulations can easily use up very large amounts of CPU-time. It is clear that the pace of progress towards simulations of more complex and larger molecular models is set by advances in microprocessors technology and computer architecture. In the last couple of years performance of microprocessors has increased by 60%-per-year and this quadrupling of performance every three years is believed to be possible at least also for the next couple of years [13].

MD simulatiions of today can be performed on many different kinds of computer resources, ranging from workstations, cluster of workstations, shared-memory machines, vector machines and massively parallel machines. These different types of computers are complementary. The smaller problems runs very cost-effectively on workstations and clusters of workstations, and the very largest and most compute intensive problems can be handled on the large vector machines and massively parallel machines. The vector machines are excellent at providing throughout capacity for very demanding applications and the massively parallel machines offer very high peak-performance on applications that are inherently parallel.

A wide spectrum of parallel architectures and computer systems have been designed from loosely coupled master-slave configurations [14] to shared memory multi-processor systems. For a short and enlightening overview see for example the review by Bell [13].

Computer simulations of many particle systems are very well suited for all kinds of parallel computer systems. As an example of pioneering work in massively parallel computer simulations of matter we give the work performed at the Edinburgh school by Pawley and coworkers on lattice simulations on the ICL DAP [15]. The geometric mapping of particles on processors in the straight-forward method in simulations of solid state systems. In liquids, however, one major complication comes from the continuous, seemingly chaotic movement of particles which calls for completely different mapping schemes. In general, the most optimal parallel scheme for a particular problem will depend on the hardware at hand, the size of the simulation and on the type of interparticle interactions. Excellent summaries of different parallel strategies in molecular simulations is given for example by Fincham [16] and Smith [17].

Examples of real molecular systems which have short-range interactions are the noble gases and nonpolar molecules. A large number of simulations have been reported where the efficiency of the linked-cell method is maximally exploited. Very efficient vectorization schemes were introduced by Rapaport [18] and Grest *et al.* [19], while parallelization aspects are discussed by Smith [17]. Several of the recent parallel large-scale calculations are summarized in our previous paper [20]. These parallel large-scale simulations, of up to five million Lennard-Jones particles, have mostly dealt with conceptual and methodological questions of efficient implementation on different parallel machines. However, the parallel methods developed for these problems is a natural first step when developing parallel methods to treat more complicated systems.

In most real molecular systems the interactions are more complicated than just the simple Lennard-Jones potential and also include long-range electrostatic interactions. The Coulombic interactions due to charge distributions on molecules require a special techniques because of the extremely slow and conditional convergence of the electrostatic potential. The particle-particle-particle-mesh (P^3M) method, the Ewald summation method and the reaction field method are three commonly used techniques for calculating Coulombic interactions in systems of polar molecules or ionic liquids [1].

The shifted cut-off method is a simple way to force the interactions to decay to zero at the edge of the cut-off radius. It and other related techniques were popular in the early computer simulations of systems including electrostatic interactions. However, they have recently been shown to give some serious artifacts in the structural simulation results [21] as well as a considerable degree of arbitrariness to dynamic properties [22]. Both the P^3M method and the Ewald summation method split up the total force into a long-range and short-range component. In both these methods the short-range part is often treated with a linked-cell scheme, but they treat the long-range part differently. In the case of the P^3M method the long-range force is found by smearing out the charges on a mesh, solving Poisson's equation on this mesh to find the potential, numerically differentiating the potential to find the force on the mesh-points and finally interpolating the mesh-force to find the force on each particle. The Ewald summation method treats the long-range part of the force by a sum in reciprocal space.

The choice of efficient computational algorithms for calculations of long-range (electrostatic) interactions to be used in parallel computer simulations is somewhat restricted. The longer cut-off radius and more complicated computational schemes make the simulations memory demanding and may often also demand unstructured, and hence costly, general communication.

For shared memory parallel computers, the replicated data (RD) scheme has become a quite natural framework [17, 23–25] in which to implement more general, all-purpose molecular simulation packages. These packages include RD versions of the Ewald summation method.

To make use of the peak-performance offered by the highly parallel computer systems on large molecular systems, such as, proteins and DNA embedded in membranes and solutions, the serial algorithms normally used to solve these problems often have to be reformulated and even changed completely. The first step is to parallelize the computationally heavy intermolecular force calculation. The next step is to look over the calculation of bond and angle constraints and also the intramolecular force calculation. Parallel versions of simulation packages e.g. Amber [26] have been reported running on *i*PSC/860.

A major part of the development of the parallel computational simulation schemes for liquids and solutions have been done for MIMD [27] type of computers. The full generality offered by the MIMD-machines can both be an asset as well as a burden. In contrast, when developing programs on SIMD-machines one is morerestricted but have the advantage of easier programming and debugging. In the case of MD simulations, SIMD-machines offer a very attractive programming model which will be portable to future generations of both MIMD and SIMD machines.

Our long-term goal is to devise an instrument which can be used to investigate system-size dependencies, e.g. effects of periodic boundary conditions. To achieve this, we intend to continue the development of our data-parallel molecular dynamics simulation code and include a proper treatment of Coulombic interactions. We have decided to use a parallel formation of the linked-cell method often called the coarse-grained cell (CGC) method [21, 28] as a base, and investigate how the Ewald summation method performs when implemented in this data-parallel context.

In this communication we apply the CGC method on the ionic melt of sodium chloride and present timing results of systems up to two million particles. The paper is organizd as follows. In Section 2, we give a description of the model salt to be simulated. The data-prallel Ewald summation method is introduced in Section 3. The algorithm is presented in Section 4 and timing results and discussion follow in 5. Finally conclusions, are drawn in Section 6.

2 COMPUTER SIMULATIONS OF FUSED SALTS

Historically the first computer simulation was the Monte Carlo work by Woodcock and Singer [29], published 1971 (same year as Rahman and Stillinger reported the first MD simulation of liquid water [30]). They used a rigid ion empirical Born- Huggins-Mayer model parameterized by Fumi and Tosi (BHMFT) [31]. The work by Woodcock and Singer initiated an intensive period of computer simulations of molten salts found in a comprehensive review by Sangster and Dixon [32]. These early simulations of liquid salts gave many new and earlier inaccessible data about the structure and dynamics of alkali halids and rare earth halides. However, the BHMFT model gave identical radial distribution functions for like ions of KCl and NaCl, while in neutron scattering experiments they were resolved. Laaksonen and Clementi [33] carried out extensive ab initio calculations at the CI level of approximation for sodium chloride system to construct an accurate potential model for NaCl. The ab initio pair potential was able to resolve g + + and g − − functions. It also gave subsidiary peaks (invisible in BHMFT curves) for the like ion radial distribution functions. The potential energy obtained from the simulations with ab initio potentials was too low compared to the experiment. It was attributed to the lack of many-body interactions, since the potential energy function was fitted from the points of dimer configurations.

It was also suggested [32] that polarizable ion models should be used instead of rigid ion models to improve the simulation results. At least for heavy ions and when the cations and anions are of different size [34,35]. Polarizability can be implemented for example by using so called shell models. A serious drawback with shell models have been that they are expensive to use in computer simulations. Most recent developments of shell models are by Fincham and Mitchell [36] and Smith [37].

Parallel computer simulations of ionic systems have been reported by e.g. Boyer [38] using the ICL DAP, Boltjes and de Leeuw [39] using the CM2 and Rajiv *et al.* [40] using the *i*PSC/860. In this paper we present a data-parallel computational scheme to treat ionic systems by means of molecular dynamics simulations. The software is based on the sequential version of McMoldyn [41]. For the basic strategy to implement simulation codes on SIMD type of computers see [20] and references cited therein.

As a model potential for our investigations we have chosen the BHMFT potential [31] to study systems of fused salts. The potential V between two particles i and j, with respective charges q_i and q_j separated by a distance r_{ij} is given by

$$V(r_{ij}) = q_i q_j \frac{e^2}{r_{ij}} + A_{ij}\exp(B(\alpha_{ij} - r_{ij})) + \frac{C_{ij}}{r_{ij}^6} + \frac{D_{ij}}{r_{ij}^8} \tag{1}$$

The first term in equation (1) is the electrostatic interaction, the exponential term is the Born-Huggins repulsion and the third and fourth term represent the dipole-dipole and dipole quadrupole dispersion energies. The parameters have been obtained from [42].

3 EWALD SUMMATION METHOD

In periodic systems the Coulombic interaction is conditionally convergent. This has as a consequence that direct summation of the interactions depend on the exact order in which it is performed. The Ewald summation method transforms the conditionally convergent sum into three parts, the real-space term, the Fourier-space term and the self-interaction term [43]. By choosing a sufficiently large cut-off, r_c, the Ewald summation for an electrically neutral system of N charges with no external field can be written

$$V_{Coul} = V_r + V_f - V_s \tag{2}$$

$$V_r = \frac{1}{4\pi\varepsilon_0} \sum_{r_{ij} < r_c} \frac{q_i q_j}{r_{ij}} erfc(\alpha, r_{ij}) \tag{3}$$

$$V_f = \frac{1}{2V\varepsilon_0} \sum_{k \neq 0} \frac{\exp(-k^2/4\alpha^2)}{k^2} Q_k Q_k^* \tag{4}$$

$$V_s = \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N} q_i^2 \tag{5}$$

Table 1 Parameters of the BHMFT potential used in the simulation.

	$A[10^{-19}J]$	$B[10^{10}m^{-1}]$	$\sigma_{ij}[10^{-10}m]$	$C[10^{-79}Jm^6]$	$D[10^{-99}Jm^8]$
+ +	0.4225	3.15	2.34	− 1.68	− 0.80
+ −	0.3380	3.15	2.75	− 11.20	− 13.90
− −	0.2535	3.15	3.17	− 116.00	− 233.00

where *erfc* in equation (3) is the complementary error function. In equation (4), Q_k denotes the charge density at wave vector $k = 2\pi(k_1, k_2, k_3)/\mathrm{L}$ where k_1, k_2 and k_3 are integers and $V = L^3$ is the volume of the simulation box,

$$Q_k = \sum_{i=1}^{N} q_i \exp(ik \cdot r_i), \tag{6}$$

r_i and q_i the positiin and charge respectively of particle i. The parameter α[1] in equation (3) is chosen so that the complementary error function is negligible when $r_{ij} \geqslant r_c$. This in turn governs the number of terms in the sum of equation (4) in order to assure sufficient accuracy. Effectively one uses a cut-off in the reciprocalspace too, k_c, and continue summing for $k^2 < k_c^2$. For a chosen accuracy optimal values of r_c, α, and k_c the Ewald summation method scales as $O(N^{3/2})$ [44].

4 DATA-PARALLEL EWALD SUMMATION

The coarse-grained cell method is a parallel formulation of the linked-cell method where the simulation box is divided into a number of equally sized cubic cells. The size of the cell side is conveniently chosen to be the same as the potential cut-off radius, thereby limiting the search region of possible interacting particles to a total of 27 cells. Each one of the cells are mapped to a processor. If there are more cells than processors then each processor takes care of sveral cells. For best performance, cells that are logical neighbors should be mapped to processors that are connected by direct communication links. In the case of the CM-system this is taken care of automatically by the compiler.

The Ewald summation method splits the calculation of the Coulombic interaction into the real-space part, the Fourier-space part and the self-interaction part. By a suitable choice of the convergence parameter α in equation (3) one can calculate this part together with the short-range terms of the BHMFT potential, equation (1). The Fourier-space part does not require any exchange of information between cells except for the global summation necessary to find Q_k in equation (6). The self-interaction term is a constant during a simulation and can be calculated at initialization. In practice, this means that when one starts from a program that simulates a system with only short-range interactions one has to add code to take care the real-space part and two subroutines which will take care of calculating the Fourier-space part and the self-interaction part.

4.1 Real-space contribution

This calculation loops over the 13[2] neighboring cells and the 'home-cell'. It first selects the particles which have a chance of interacting with particles in the neighbor cell, then

[1] *Often called the* Ewald convergence parameter.

[2] Symmetry of force gives $(27 - 1)/2 = 13$.

zeros out force accumulators and changes origin of the particles which have a chance of interacting and consequently have to be shifted to the neighboring cell. This is needed because particles in all cells have their coordinates in relation to their current 'home-cell'. With this arrangement it is possible to write more unifor malgo- rithms since particles in the interior will not have to be treated differently than particles belonging to a cell at the edge of the box. With the coordinates properly shifted the particles are sent away to the neighbor cell and the interactions are calculated. The forces which have been found are sent back and accumulated. This is needed because we only loop over half the 26 neighbor cells. Adding in the real-space part of the Ewald summation only changes the routine which contains the force calculations.

4.2 Fourier-space contribution

The Fourier-space term first needs to adjust to global coordinates and then continues by calculating the first term used in the iterative scheme employed to calculate the *exp* term in equation (6). The scheme is based on the following formulas

$$\sin nx = 2\sin(n-1)x\cos x - \sin(n-2)x \tag{7}$$

$$\cos nx = 2\cos(n-1)x\cos x - \cos(n-2)x \tag{8}$$

We have elected an iterative scheme because the traditional way of calculating expression (6) becomes very memory demanding when one has to include many terms in the sum. Furthermore, we make use of the symmetries and calculate only in an octant of k-space. At the end we change from a global to a local coordinate system.

5 TIMING RESULTS AND DISCUSSION

All the runs have used a cut-off of 12Å and both the real-space and Fourier-space sums have been truncated when the terms have been less than $\exp(-\pi^2)$. Times have been measured with the available timers; the 'elapsed time' has been used.

The runs have been performed on a 8K CM200 with 1 GByte of memory. The machine can also be split in two equal halves. That is why there are also timings from a 4K machine. The compiler version used was CM Fortran version 2.1 Beta II; the compiler model has in all cases been the 'slicewise' model. For more information concerning the CM system see [45].

In the cases of $64 - 8 \times 8 \times 8$ and $125 - 8 \times 8 \times 8$, Table (2), we can clearly see the effect of running too small a problem on a parallel machine, or in other terms when the problem is not big enough to allow for all processors have something to do. When the problems get large enough we see that we have not quite a linear scaling in the number of processors. One can also note that when the cell-size becomes too largethen t_p goes up. We note also that the small 8K CM 200 will actually harbor a simulation of over 2 million particles. It is however not recommended for practical MD simulations.

In Table (3) the time per time-step is split between the real-space part and the reciprocal-space part. We see that when the relation between real-space part and Fourier-space part time is approximately 1/4 that is when we have the smallest t_p.

Table 2 Internal units are in relation to the length of the cell-side of the whole simulation box. The cut-off r_c and α are given in internal units. k is the cut-off in Fourier-space, N the size of the system, T the time per time-step and $t_p = T/N$.

8 K	r_c	α	k	N	$T[s]$	$t_p[\mu s]$
64–8 × 8 × 8	0.114	27.6	11	32768	51	1562
125–8 × 8 × 8	0.091	34.5	13	64000	125	1962
64–16 × 16 × 16	0.056	55.2	16	262144	104	395
125–16 × 16 × 16	0.045	68.9	17	512000	203	396
64–32 × 32 × 32	0.028	110.3	20	2097152	1284	612
4 K	r_c	α	k	N	$T[s]$	$t_p[\mu s]$
64–8 × 8 × 8	0.114	27.6	11	32768	28	872
125–8 × 8 × 8	0.091	34.5	13	64000	62	966
64–16 × 16 × 16	0.056	55.2	16	262144	195	742
125–16 × 16 × 16	0.045	68.9	17	512000	381	744

Table 3 The time per time-step is split between the real-space part and the Fourier-spcae part.

8 K	N	*real-space*	*recip-space*
64–8 × 8 × 8	32768	35	16
125–8 × 8 × 8	64000	90	35
64–16 × 16 × 16	262144	22	82
125–16 × 16 × 16	512000	44	159
64–32 × 32 × 32	2097152	121	1163
4 K	N	*real-space*	*recip-space*
64–8 × 8 × 8	32768	14	14
125–8 × 8 × 8	64000	32	30
64–16 × 16 × 16	262144	37	157
125–16 × 16 × 16	512000	72	309

From the times presented here one can surmise that on a full CM 200 one can run a problem containing 256 K particles with a time-step of just above 10 seconds. An interesting line of investigation would be to see if one could neglect most of the Fourier-space part of the calculation. This would mean great improvements in time per time-step but hopefully only at a small loss in accuracy.

6 CONCLUSION

For appropriate choices of systems sizes we can conclude that for a full CM 200 one would have a times-step between 10 and 15 seconds for a system of 256 K particles. Larger systems than that should be run on the new MPP-machines which are now available.

The Ewald summation method scales as $O(N^{3/2})$ when one chooses the parameters r_c, α and k_c [46, 47]. From these papers we have, in internal units and with an accuracy of $\exp(-\pi^2)$, $r_{\mathrm{opt}} \approx \sqrt{\pi}N^{-1/6}$ which results in $\alpha_{\mathrm{opt}} = k_{\mathrm{opt}} \approx \sqrt{\pi}N^{1/6}$. Take a concrete example. Suppose we want to simulate 2 M particles. Then according to the above formulas we should choose $r_c \approx 1/6$. If one uses the coarse-grained cell method this implies that one should use a system which is 6x6x6 in the cell subdivision and has at least 9700 particles in each cell. This means that we have to choose a very large cut-off and we will spend a lot of unnecessary calculations when evaluating the short-range forces.

The conclusion is then that for very large simulations the Ewald summation technique and the coarse-grained cell method do not match very well. However, if one could neglect the Fourier-space part of the calculation at only a small loss of accuracy then the method may still turn out to be an interesting alternative. To resolve this question more conceptual work is necessary in order to decide weather this is an acceptable approximation.

References

[1] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1987.

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

[3] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review,* **136A,** 405–411 (1964).

[4] J.A. McGammon and S.C. Harvey. *Dynamics of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, 1987.

[5] D. Nicholson and N.G. Parsonage. *Computer Simulation and Statistical Mechanics of Adsorption.* Academic Press, New York, 1982.

[6] M. Ronchetti and G. Jacucci, editors. *Simulation Approach to Solids.* Kluwer Academic Publishers, Milano, 1990.

[7] G. Cicotti, D. Frenkel and I.R. McDonald, editors. *Simulation of Solids and Liquids.* North Holland, Amsterdam, 1987.

[8] Wm. G. Hoover. *Computational Statistical Mechanics.* Elsevier, Amsterdam, 1991.

[9] J.M. Haile. *Molecular Dynamics Simulation, Elementary Methods.* John Wiley & Sons. New York, 1992.

[10] D.W. Heermann. *Computer Simulation Methods in Theoretical Physics.* Springer-Verlag, Berlin, second edition, 1990.

[11] H. Gould and J. Tobochnik. *An Introduction to Computer Simulation Methods. Applications to Physics,* volume I and II. Addison Wesley New York, 1989.

[12] R.W. Hockney and C.R. Eastwood. A 10000 particle molecular dynamics model with long range forces. *Chemical Physics Letters,* **21,** 589– (1973).

[13] Gordon Bell. Ultracomputers, a teraflop before its time. *Communications of the ACM,* **35** (8), 27–47 (1992).

[14] E. Clementi, G. Corongiu, J.H. Detrich, H. Khanmohammadbaigi, S. Chin, L. Domingo, A. Laaksonen, and H.L. Nguyen. Parallelism in computational chemistry: Applications in quantum and statistical mechanics. *Physica,* **131B,** 74–102 (1985).

[15] G.S. Pawley and G.W. Thomas. The Implementation of Lattice Calculations on the DAP. *Journal of Computational Physics,* **47(2),** 165–178 (1982).

[16] David Fincham. Parallel computer and molecular simulation. *Molecular Simulation,* **1,** 1–45 (1987).

[17] W. Smith. Molecular dynamics on hybercube parallel computers. *Computer Physics Communications,* **62,** 229–248 (1991).

[18] D.C. Rapaport, Large-scale molecular dynamics simulation using vector and parallel computers. *Comp. Phys. Rep,* **9,** 1–53, (1988).

[19] G.S. Grest, B. Dünweg, and K. Kremer. Vectorized link cell fortran code for molecular dynamics for a large number of particles. *Computer Physics Communications,* **55,** 269–286 (1989).

[20] Fredrik Hedman and Aatto Laaksonen. Data parallel large-scale molecular dynamics for liquids. *International Journal of Quantum Chemistry,* **46**(1), 27–38 (1993).

[21] C.L. Brooks III. The influence of long-range force truncation on the thermodynamics a aqueous ionic solutions. *Journal of Chemical Physics*, **86**, 5156 (1986).

[22] O. Teleman. An efficient way to conserve the total energy in molecular dynamics simulations; boundary effects on energy conservation and dynamic properties. *Molecular Simulation*, **1**, 345 (1988).

[23] W. Smith. A replicated data molecular dynamics strategy for the parallel Ewald sum. *Computer Physics Communications*, **67**, 392 (1992).

[24] W. Smith, C. Dean, D. Fincham, and D. Tildesley. DL_POLY: a macromolecular simulation package. *CCP5 Information Quarterly*, **35**, 15 (1992).

[25] W. Smith and T. Forrester. DL_POLY: Progress report. *CCP5 Information Quarterly*, **37**, 37 (1993).

[26] Stephen E. DeBolt and Peter A. Kolman. AMBERCUBE MD, Parallelization of AMBER's Molecular Dynamics Module for Distributed-Memory Hypercube Computers. *Journal of Computational Chemistry*, **14**(3), 312–329 (1993).

[27] W. Smith. Molecular dynamics on distributed memory (MIMD) parallel computers. *Theoretica Chimica Acta*, **84**, 385–398 (1993).

[28] Fredrik Hedman and Aatto Laaksonen. An approach to data parallel molecular dynamics for liquids. *International Journal of Modern Physics C*, **4**(1), 41–48 (1993).

[29] L.V. Woodcock and K. Singer. Thermodynamic and structural properties of liquid ionic salts obtained by Monte Carlo computation. *Trans. Faraday. Soc.*, **67**, 12–30 (1971).

[30] A. Rahman and F.H. Stillinger. Molecular dynamics study of liquid water. *J. Chem. Phys.*, **55**, 3336–3359 (1971).

[31] F.G. Fumi and M.P. Tosi. Ionic sizes and Born repulsive parameters in the NaCl-type alkali halides I. The Huggins-Mayer and Pauling forms. *J. Phys. Chem. Solids*, **25**, 31–43 (1964).

[32] M.J.L. Sangster and M. Dixon. Interionic potentials in alkali halides and their use in simulations of molten salts. *Adv. Phys*, **25**, 247 (1976).

[33] A. Laaksonen and E. Clementi. Theoretical study of some gas, liquid and crystal properties of sodium chloride using *ab initio* potentials. *Mol. Phys*, **56**, 495–524 (1985).

[34] M. Dixon and M.J. Gillan. Structure of molten alkali chlorides. I. Molecular dynamics study. *Philosophical Magazine B*, **43**, 1099 (1981).

[35] M. Dixon. Molecular dynamics studies of molten NaI. I. Quasi-elastic neutron scattering. *Philosophical Magazine B*, **47**(5), 509–530 (1983).

[36] P.J. Mitchell and D. Fincham. Shell model simulations by adiabatic dynamics. *Journal of Physics: Condensed matter*, **5**, 1031–1038 (1993).

[37] W. Smith. to be published.

[38] L.L. Boyer and G.S. Pawley. Molecular dynamics of clusters of particles interacting with pairwise forces using a massively parallel computer. *Journal of Computational Physics*, **78**, 405 (1991).

[39] B. Boltjes and S.W. De Leeuw. Molecular dynamics simulation on the connection machine using fortran. *Molecular Simulation*, **7**, 1–42 (1991).

[40] R.K. Kalia, S. De Leeuw, A. Nakano, and P. Vashishta. Molecular-dynamics simulations of Coulombic systems on distributed-memory MIMD machines. *Computer Physics Communications*, **74**, 316–326 (1993).

[41] A. Laaksonen. Computer simulation package for liquids and solids with polar interactions. I. McMOLDYN/H20: Aqueous systems. *Computer Physics Communications*, **42**, 271 (1986).

[42] John W.E. Lewis and K. Singer. Thermodynamic properties and self-diffusion of molten sodium chloride. *J. Chem. Soc. Faraday II*, **71**, 41–53 (1975).

[43] S.W. De Leeuw, J.W. Perram and E.R. Smith. Simulation of electrostatic systems in periodic boundary conditions I. Lattice sums and dielectric constants. *Proc. R. Soc. Lond.*, **A 373**, 27–56 (1980).

[44] 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. *Molecular Physics*, **65**, 875–893 (1988).

[45] TMC, Thinking Machines Corporation, Cambridge, Massachusetts. *Connection Machine CM–200 Technical Summary*, 1992.

[46] J. Kolafa and J.W. Perram. Cutoff Errors in the Ewald Summation Formulae for Point Charge Systems. *Molecular Simulation*, **9**, 351–368 (1992).

[47] D. Fincham. Optimisation of the Ewald Sum. *CCP5 Information Quarterly*, **38**, 17–24 (1993).