JOURNAL OF COMPUTATIONAL PHYSICS 78, 405–423 (1988)

Molecular Dynamics of Clusters of Particles Interacting with Pairwise Forces Using a Massively Parallel Computer

L. L. BOYER*

Condensed Matter Physics Branch, Naval Research Laboratory, Washington, D. C. 20375-5000

AND

G. S. PAWLEY

Department of Physics, University of Edinburgh, Kings Buildings, Edinburgh EH9 3JZ, United Kingdom

Received April 10, 1987; revised December 7, 1987

An algorithm is presented for computing the forces on atoms in a free cluster using a massively parallel computer with processing elements connected in a square grid. The method is applied in molecular dynamics simulations of the melting transition in NaF and the plastic phase transformations, as well as melting, in sulphur hexafluoride.

INTRODUCTION

The dynamics of particles interacting with pairwise forces is a problem ideally suited for simulation using highly parallel computers of simple architecture. The ICL distributed array processor (DAP) [1], consisting of 4096 single-bit processing elements (PEs) connected in a square cyclic grid, is one such machine. The periodic arrangement of the PEs can be exploited with great efficiency on systems governed by periodic boundary conditions if the particles of the system do not diffuse or change places too quickly. For such problems the PEs are assigned to specific particles of the system, giving a mapping which allows the easy transfer of data between interacting molecules. Although neighboring molecules may not be on neighboring PEs [2] there is very little loss in computer time as the DAP is very efficient in its communications. This approach has been successful in studying the plastic phase transformations in SF_6 [3–6] and *n*-butane [7] and melting in naphthalene [8].

In the present paper we introduce another approach applicable to any finite system governed by pairwise interactions. Specifically, we consider a finite cluster in

* Work performed as visiting scientist at the University of Edinburgh.

0021-9991/88 $3.00

free space. For this problem our strategy is simply to compute the interactions between *all* possible pairs in the system, 4096, or as near as possible to 4096, at a time. Implementing this "brute force" approach then amounts to finding an efficient method for *accounting* for all the interactions. In the following section we present a method which accomplishes this for the DAP architecture with remarkable efficiency.

The method has a few advantages over other methods. As we have dispensed with neighbor lists for choosing interactions we do not have to include a section of software which updates such lists. Neighbor list searches are most efficiently implemented in different ways for solids and gases, presenting problems in the regime of low-density liquids: All these systems can be treated now with exactly the same software. Another problem which usually has to be faced is how to implement zero-pressure simulations, but this again is a problem which does not concern us because our system is a cluster in free space. Finally, we note that the Coulomb interaction is easy to handle for the finite cluster, whereas special techniques, such as the Ewald method, must be employed for the infinite system.

The study of surface-related properties is currently an important area for research in condensed matter physics. Molecular dynamics (MD) of finite systems, or clusters, offers the possibility of studying both surface and bulk properties by the same method. We illustrate this point with selected results from ongoing studies of melting and plastic phase transformations.

Computational Strategy

The most time consuming part of an MD calculation is the computation of the forces acting on the atoms or ions in the system. (Henceforth we use the term atoms to mean atoms or ions.) The feature of the force calculation exploited by a massively parallel computer is the fact that many pair interactions in the system are derived from a single potential function. A monatomic system, for example, has only one potential function, while a diatomic system has 3 different potentials, etc. The DAP is a single-instruction-multiple-data (SIMD) computer; which is to say that at a given time all PEs are acting on one and the same instruction. In a *single instruction* one can compute the forces due to as many pairs of a given type as there are PEs in the computer. The broadcast instruction acts on the relative coordinates of the pairs (*multiple data*) to determine the forces for each of the pairs. Parallel computers which allow PEs to act "independently," known as multiple-instruction-multiple-data (MIMD) machines are obviously more flexible, in general, than an SIMD machine. However, for the problem considered here an MIMD machine would have no real advantage over the SIMD machine. If one had a limited budget to design a computer solely to do molecular dynamics calculations, it could well be a massively parallel SIMD machine.

In order to simplify the discussion of the force calculation, we assume our computer is a 4×4 array of 16 PEs, and our cluster consists of 8 singly charged ions

interacting via the Coulomb potential. In this case, SIMD operations are carried out on the elements of the natural sized square matrices. By natural matrices we mean, in this case, 4×4 arrays. Natural matrices in the DAP are, or course, 64×64 arrays. Similarly, we refer to a sequence of 4 numbers of an arbitrary array as a natural vector. For example, a row or a column of a natural matrix is a natural vector.

The algorithm presented here requires four operations involving the natural matrices, in addition to the basic arithmetic operations needed to evaluate the potential and force functions: (1) an efficient method for copying a sequence of 4 numbers of an arbitrary array into either all the rows or all the colums of a natural matrix; (2) a method for summing the rows, or colums, of all the elements of a natural matrix; (3) the capability of logical masking, which specifies, using a logical matrix, whether or not operations on individual elements of natural matrices are actually carried out; and (4) the transpose operation.

Let X, Y, and Z, be arrays dimensioned $X(,2)$, $Y(,2)$, and $Z(,2)$, and let the elements of X, Y, and Z be the x, y, and z coordinates of the positions of the 8 atoms. In other words, the x components of atoms 1–4 form the first natural vector of X, specifically, $X(,1)$, and the x components of atoms 5–8 form the second natural vector of X, namely, $X(,2)$. We refer to pairs of atoms whose positions are stored in two vectors as two-vector interactions, and pairs arising from the positions in one vector as one-vector interactions.

First consider the two vector interactions. Let XR be the natural matrix whose rows are $X(,1)$, the x coordinates of the positions of atoms 1–4:

$$XR = \begin{matrix} x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4. \end{matrix}$$

And, let XC be the natural matrix whose columns are $X(,2)$, the x components of the positions of atoms 5–8:

$$XC = \begin{matrix} x_5 & x_5 & x_5 & x_5 \\ x_6 & x_6 & x_6 & x_6 \\ x_7 & x_7 & x_7 & x_7 \\ x_8 & x_8 & x_8 & x_8. \end{matrix}$$

The commands which create XR and XC in DAPFORTRAN are $XR =$ MATR($X(,1)$) and $XC =$ MATC($X(,2)$).

The relative coordinates for all possible pairs between the first and second vector atoms are given by $XR - XC$, $YR - YC$, and $ZR - ZC$ where YR, YC, ZR, and ZC

are the analogous matrices for the y and z components. The relative separations for each of these pairs are the elements of the matrix

$$R = SQRT((XR - XC)** 2 + (YR - YC)** 2 + (ZR - ZC)** 2)$$

and the x components of the forces associated with these pairs are the elements of

$$XX = (XR - XC)/R**3.$$

Let FX be an array, dimensioned $FX(,2)$, for storing the x components of the forces on the ions. It is updated to include the above contributions by two statements in DAPFORTRAN:

$$FX(,1) = FX(,1) + \text{SUMR}(XX)$$

and

$$FX(,2) = FX(,2) - \text{SUMC}(XX),$$

where SUMR (SUMC) creates a natural length row (column) vector from the sum of the rows (columns) of a natural matrix.

The remaining pairs are one-vector interactions. They are accounted for in a similar fashion by constructing the matrices

$$XR = \begin{matrix} x_5 & x_6 & x_7 & x_8 \\ x_1 & x_6 & x_7 & x_8 \\ x_1 & x_2 & x_7 & x_8 \\ x_1 & x_2 & x_3 & x_8 \end{matrix}$$

and

$$XC = \begin{matrix} x_1 & x_5 & x_5 & x_5 \\ x_2 & x_2 & x_6 & x_6 \\ x_3 & x_3 & x_3 & x_7 \\ x_4 & x_4 & x_4 & x_4 \end{matrix}$$

using the DAPFORTRAN statements

$$XR = \text{MERGE}(\text{MATR}(X(,1)), \text{MATR}(X(,2)), LT)$$

and

$$XC = \text{MERGE}(\text{MATC}(X(,1)), \text{MATC}(X(,2)), LTD),$$

where LT and LTD are, respectively, logical matrices with the lower triangle and the lower triangle plus the diagonal equal to true:

$$LT = \begin{matrix} F & F & F & F \\ T & F & F & F \\ T & T & F & F \\ T & T & T & F \end{matrix}$$

and

$$LTD = \begin{matrix} T & F & F & F \\ T & T & F & F \\ T & T & T & F \\ T & T & T & T. \end{matrix}$$

The MERGE statement constructs a matrix from the elements of two matrices (first two arguments) using the elements of the first (second) matrix if the corresponding elements of the logical matrix (third argument) are true (false).

Notice that the lower triangle of

$$XX = (XR - XC)/R**3$$

contains the x components of forces due to interactions between pairs of first-vector atoms, while the upper triangle of XX contains the x components of forces due to pairs of second-vector atoms. The diagonal of XX contains interactions which have already been counted, so they have to be discarded. The following statements update the force array from the elements of XX:

$$XX = \text{MERGE}(XX, 0.0, \text{LD})$$

$$XXT = -\text{TRAN}(XX)$$

$$FX(,1) = FX(,1) + \text{SUMR}(\text{MERGE}(XX, XXT, LT))$$

$$FX(,2) = FX(,2) + \text{SUMR}(\text{MERGE}(XX, XXT, .NOT.LT)),$$

where TRAN is the transpose operation and LD is a logical matrix with false on the diagonal.

$$LD = \begin{matrix} F & T & T & T \\ T & F & T & T \\ T & T & F & T \\ T & T & T & F \end{matrix}$$

The characteristic feature of the method just described is the expression of relative coordinates as differences between two matrices, one with identical rows and the other with identical columns. We shall refer to this as the row-column difference (RCD) method. It is easily generalized to include arbitrary numbers of atoms by (a) incorporating the above operations in suitable loops and (b) introducing "dummy" atoms if the number of atoms of one type is not a multiple of the rank (N) of the natural matrix. Let NN be the number of natural vectors required to store one of the position components of one type of atom in the system. For example, the DAP calculations discussed in the next section pertain to a cluster of 216 ions, 108 Na^+ and 108 F^- ions. Here $N=64$ and $NN=2$: The second group of Na^+ ions contain 20 dummies, and likewise for the F^- ions. Forces due to dummy ions are excluded from the force array updates using logical masking techniques similar to those employed above in calculating the one-vector interactions. A single loop from 1 to NN*2, in incremeents of 2, accounts for all one-vector interactions. Another double loop over I and J, $I=2$ to NN*2 and $J=1$ to $I-1$, accounts for two-vector interactions.

Before proceeding to a discussion of results, we consider briefly another class of algorithms for parallel computation of forces in a cluster. A baxically different strategy from the RCD method is to *store* position components in *two* different natural arrays and then account for all pairs by suitably *shifting* the elements of one array relative to the other. We call this the store two and shift (STS) strategy. Within the STS approach two different situations can arise: The number of PEs can be either greater than or less than the number of atoms. We briefly consider two examples which illustrate both situations as well as two different architectures.

In the first example we consider a system of 64 atoms using a 16×16 array of PEs. In one natural matrix ($X1$) four copies of the x components of position vectors are stored in columnwise order as follows: atoms 1–16 in the 1st, 5th, 9th, and 13th columns; atoms 17–32 in the 2nd, 6th, 10th and 14th columns; atoms 33–48 in the 3rd, 7th, 11th, and 15th columns; and atoms 49–64 in the 4th, 8th, 12th, and 16th columns. A second natural matrix ($X2$) is filled with x components of positions for atoms 1–16 in the 1st, 8th, 11th, and 14th columns; atoms 17–32 in the 2nd, 5th, 12th, and 15th columns; atoms 33–48 in the 3rd, 6th, 9th, and 16th columns; and atoms 49–64 in the 4th, 7th, 10th, and 13th columns. All pairs are accounted for by repeatedly (8 times) subtracting the two matrices ($X1$ and $X2$) and cyclically shifting the elements of $X2$ south by one row. The first such subtraction contains self "pairs" in the first four columns, and thus must be excluded. Repeatedly subtracting and shifting 16 times, double counts each pair. This method was programmed for the DAP and found to be slightly faster than the RCD method when pairs are double counted in both methods. We did not consider the modest improvement in efficiency offered by this method sufficient, for this initial study, to offset the convenience of the simpler and more flexible RCD method.

Now consider a system of 16 atoms and assume we have a parallel computer consisting of 8 PEs connected in a ring. Our natural array for this machine is a vector with 8 components. First consider atoms 1–8 interacting with atoms 9–16. As

before, we illustrate the accounting of all pairs by looking at the difference between x components of atom positions. Let $X1$ be a natural vector containing the x components for atoms 1–8, $X1(1)$ the x component for atom 1, $X1(2)$ the x component for atom 2, etc., and similarly let $X2$ be a natural vector containing the x components for atoms 9–16. The relative components of all pairs between the two groups of atoms are obtained by repeatedly (8 times) subtracting $X2$ from $X1$ and shifting the elements of $X2$ clockwise. Interactions involving atoms in one group with themselves are obtained by making $X2$ a copy of $X1$, shifting the elements of $X2$ one unit clockwise, and then repeatedly (4 times) subtracting $X2$ from $X1$ and shifting $X2$ clockwise. The last such subtraction must be given special consideration, otherwise 4 pairs will be double counted. This would not occur if our ring consisted of an odd number of PEs.

Thermal Expansion and Melting of Alkali Halides

Molecular dynamics calculations for NaF clusters containing 216 ions and 512 ions have been carried out on the DAP for the purpose of studying properties related to the melting transition. Pair potentials derived by the method of Gordon and Kim [9], which require no experimental input other than universal constants, were used to compute the forces. To a good approximation, the numerical values for the Gordon–Kim potential energy of two interacting ions can be expressed as the sum of a long range part plus four exponential terms, where the short range exponential terms account for the short range part of the electrostatic interaction and the kinetic, exchange, and correlation energy of the electrons [10]. Of all the alkali halides we chose to study NaF because the Gordon–Kim potentials were found to give best agreement with experiment for the thermal expansion coefficient and elastic constants, at room temperature, for this material [10].

A 216-ion cluster was prepared by starting with a block of 6 (100) planes, 6×6, or 36 ions per plane, in their perfect lattice positions. The lattice constant used was 8.6 Bohr, which is close to the value for minimum energy of the bulk crystal. In addition, the ions were given small random displacements and velocities, but these were constrained to give a net zero linear and angular momentum for the cluster and to have the center of mass at the origin. The starting cofiguration for the 512-ion cluster was similarly prepared from 8 (100) planes, each containing an 8×8 array of 64 ions.

The time step used in these calculations was 0.0025 ps, which is roughly $\frac{1}{40}$ of the period of the highest frequency mode of the bulk crystal. The amount of computer time required for the 512 ions cluster was slightly less than 0.5 s per time step. For comparison, the same calculation carried out on a Cray XMP 1–2 was found to take ~ 0.3 s per time step.

The 216-ion cluster was initially given a small amount of kinetic energy, equivalent to a temperature $T = 200$ K. The simulation was then allowed to proceed at constant energy for 25 ps. Apparent thermal equilibrium was achieved after

several picoseconds. This was checked by least squares fitting a straight line to the kinetic energy as a function of time t in the range $13 < t < 25$ ps, and observing that the resultant line had essentially zero slope. (In future discussion we use $T(t)$ to denote the instantaneous kinetic energy, in equivalent temperature units, and reserve T to denote the equilibrium temperature with zero slope.) The total energy of the system was then increased by scaling up the velocities to give an increase equivalent to 200 K. The simulation was again allowed to proceed at constant energy until a new thermal equilibrium was attained. The resultant temperature increase was found to be almost 100 K. (Note that if the system were perfectly harmonic the equipartition theorem would require exactly 100 K increase.) This procedure was then repeated a total of 16 times. For the first 13 energies the equilibrium temperature increased as a function of total energy E, but the departure from equipartition increased dramatically. After increasing the total energy 2600 K from the value of the initial simulation the ions began to diffuse, the volume of the cluster increased and T decreased. In other words, the cluster melted. The simultaneous drop in temperature and increase in volume corresponds to a reapportionment of the total energy; specifically, a transfer of kinetic energy to potential energy. Further increase in total energy resulted in an increase in the temperature of the liquid. The equilibrium temperature of the cluster is plotted as a function of half the total energy ($E/2$) in Fig. 1. Our simulation for the 512-ion cluster also showed this general behavior, although slightly more energy (~ 100 K) was required to melt the 512-ion cluster.

It is desirable to have a method for calculating the density of selected regions of both liquid and solid clusters. For convenience we select cube-shaped regions centered at the origin, which, in our case correspond to the center of mass. Let r_c be the cube "radius"; i.e., $2r_c$ is the cube width. Next we center Gaussians of width $1/\alpha$

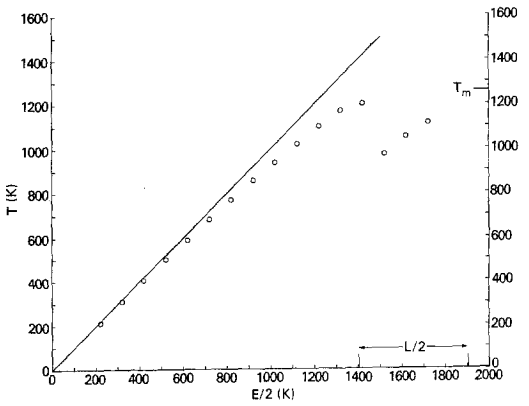

FIG. 1. Plot of equilibrium temperature T vs half the total energy. The energy scale is shifted to be zero at $T=0$ and converted to equivalent temperature units. Half the experimental latent heat ($L/2$), for comparison with $E/2$, is indicated on the abscissa. T_m denotes the experimental melting temperature.

at the instantaneous positions of the ions in the cluster. The fraction of a Gaussian centered at $\mathbf{r}=(x, y, z)$ in the selected cube is $f=f_x f_y f_z$, where $f_x=2-g_x-h_x$ for $x^2<r_c^2$, $f_x=|g_x-h_x|$ for $x^2>r_c^2$, $g_x=\mathrm{erfc}(|x+r_c|\alpha)$, $h_x=\mathrm{erfc}(|x-r_c|\alpha)$ (with similar expressions for y and z) and erfc is the complimentary error function. The mass density is then given by $\sum m_j f_j/8r_c$, where m_j is the mass of the jth ion, f_j is given above for the ion at $\mathbf{r}_j=(x_j, y_j, z_j)$ and $\sum$ is a summation over all the ions in the cluster. This part of the analysis was easily programmed for the DAP. For convenience we plot, instead the density, the corresponding lattice constant (a) of the rock salt structure, regardless of whether the cluster is in the liquid or solid phase. In particular, the plots of thermal expansion in Fig. 2 show results labeled

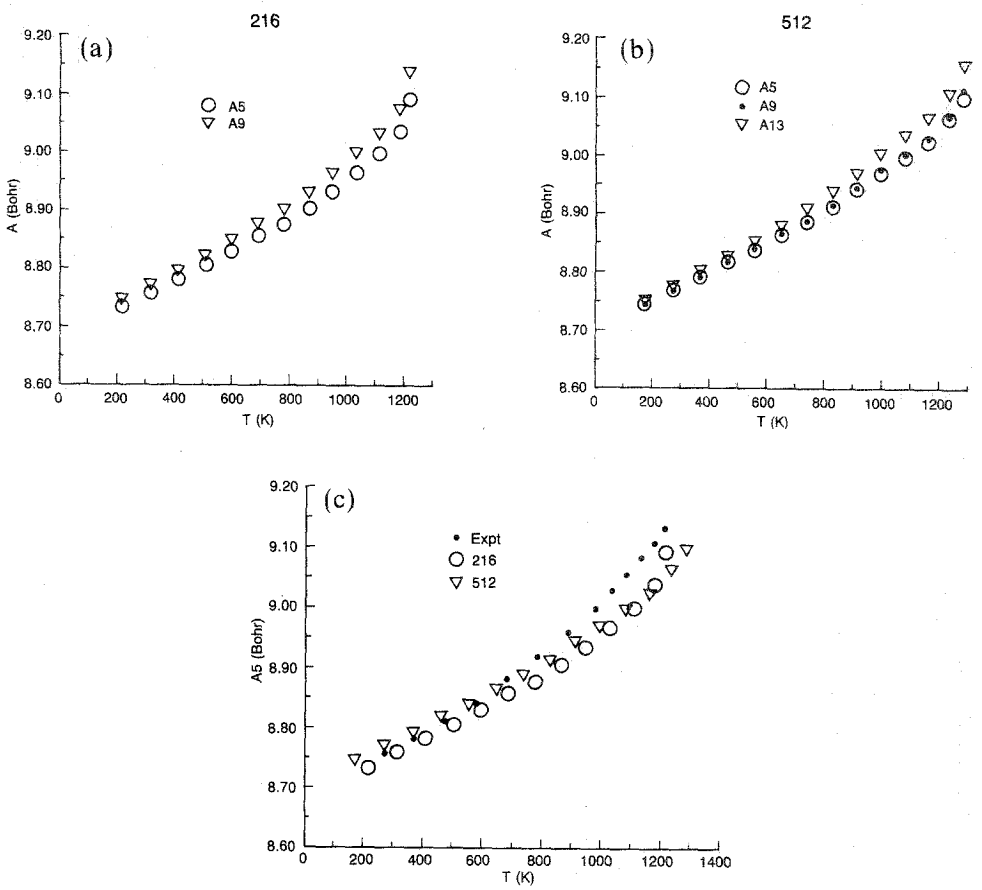

FIG. 2. Plot of the effective lattice constant, averaged over 16 ps, as a function of T for selected regions in 216-ion and 512-ion clusters, with A5, A9, and A13 corresponding respectively to cube shaped regions centered at the center of mass with widths of 10, 18, and 27 Bohr: (a) comparison of A5 and A9 for the 216-ion cluster; (b) comparison of A5, A9, and A13 for the 512-ion cluster; (c) comparison of A5 for the 216-ion and 512-ion clusters with experimental results (Ref. [11]).

A5, A9, and A13, which correspond respectively to $r_c = 9$, and 13.5 Bohr. These values were selected to enclose 8, 64, and 216 point ions, respectively, in the cube when its interior is a perfect microcrystal of NaF. An optimum value of $\alpha \sim 0.3$ $(\text{Bohr})^{-1}$ was determined by examining the results for A5, A9, and A13 derived when the $\mathbf{r}_j$ were selected to be at perfect lattice sites with a ranging from 8.4 to 10 Bohr. With $\alpha = 0.3$ the density was given correctly to $\sim 0.1\%$ by the above expressions. The error was then least-squares fitted to a quadratic function in a. Applying the resultant correction, a was then given correctly to within 0.005% everywhere in the range $8.4 < a < 10$ using this procedure.

The results in Fig. 2 were obtained by averaging 320 instantaneous values of A in a 16 ps run after thermal equilibrium was achieved. There are several noteworthy features in Fig. 2. First of all, the procedure described above appears to be a reliable method for calculating the density of selected regions in the interior of a cluster: Little noise is seen in the temperature dependence and it does produce accurate values for perfect crystals. There is substantial difference between A5 and A9 for the 216-ion cluster, as one might expect, but little difference between the two for the 512-ion cluster; indicating that A5 or A9 of the 512-ion cluster is probably close to that of a bulk crystal. To be certain of this, further results are needed for larger clusters. Results for A5 for both the 512- and 216-ion clusters are in reasonable agreement with experiment. In fact, the good agreement in the magnitude of a is somewhat fortuitous. The accuracy of predicted lattice constants using Gordon–Kim potentials is generally not better than a few percent: And zero-point motion, not included in our classical simulation, would shift the theoretical curves up by $\sim 0.5\%$. Both theory and experiment show a considerable increase in thermal expansion in the last ~ 200 K before melting. While this is considerably more pronounced in the calculations, it is evident in the experimental results by plotting the thermal expansion coefficient: see Fig. 1 of Pathak *et al.* [11]. They attribute the increase in thermal expansion just prior to melting to thermally generated Scottkey defects. In the simulation, the clusters remained perfect microcrystals up to the point where they began to melt. Clearly, much longer simulation would be required to generate point defects. The precursor is somewhat less pronounced in the 512 ion cluster than the 216 ion cluster.

cluster corresponding to the 14th point in Fig. 1 are plotted as a function of time. In the first 33 ps the displacements are computed from the instantaneous positions at 5, 9, 13, 17, 21, 25, and 29 ps. From $t = 33$ to 105 ps the displacements are computed from snapshots taken every 8 ps. The results for R2A are decomposed according to ion type (Na or F) and according to whether they were inside or outside the cube with $r_c = 9$ Bohr at the time of the snapshot. The walls of this cube lie approximately midway between the 64 "bulk" ions and the outer 152 "surface" ions the 216-ion microcrystal, which was still more or less intact for times less than ~ 30 ps. Straight-line fits to R2A are also plotted in Fig. 3. For $t > \sim 80$ ps, the cluster achieved an equilibrium temperature. In this region the slope of the linear

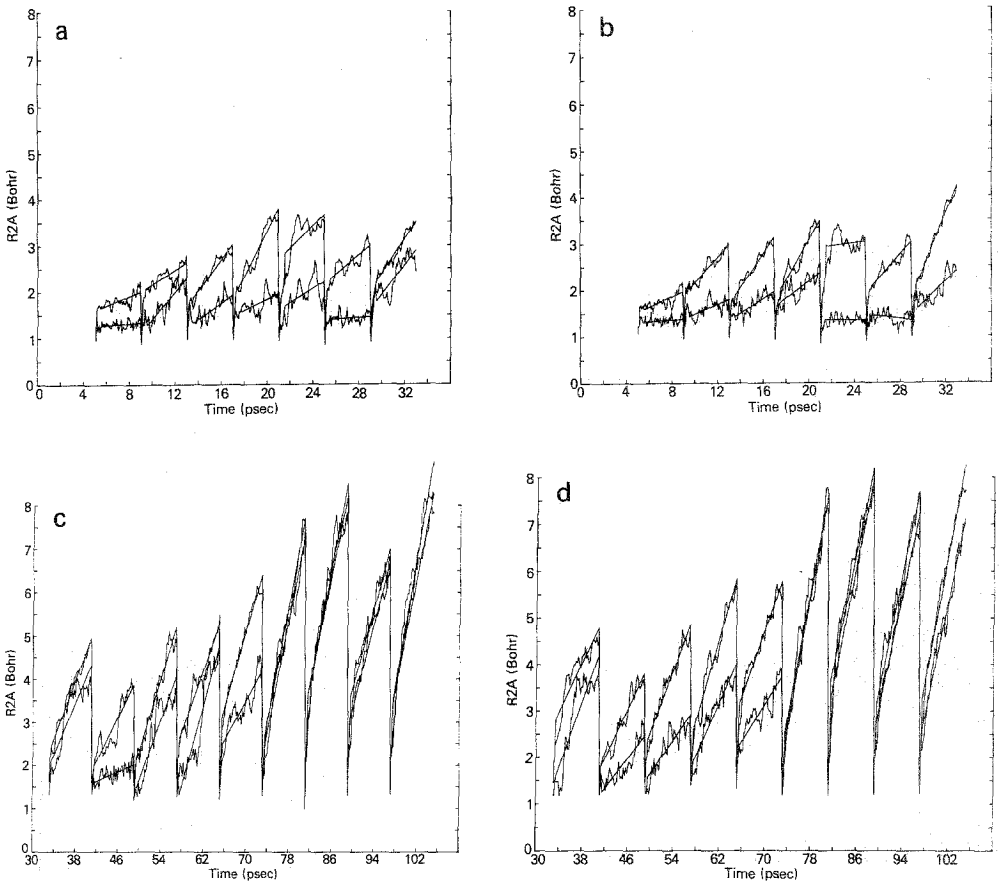

FIG. 3. (a) Plot of the root mean square displacements (R2A) of Na ions from snapshots (of the cluster corresponding to the 14th point in Fig. 1) taken every 4 ps from $t = 5$ to 27 ps. Upper (lower) curves are for ions that were outside (inside) a cube of width 18 Bohr at the time of the most recent snapshot. Straight lines correspond to linear fits to the results over all but the first 0.5 ps of the 4 ps intervals, except for vertical lines, which are to be ignored. (b) Same as (a) except for F ions. (c) Same as (a) except snapshots taken every 8 ps from $t = 33$ to 97 ps. (d) Same as (c) except for F ions.

fits to R2A could be used to obtain a diffusion coefficient. The vertical straight lines are to be ignored. The results in Fig. 3 show there is no essential difference between the diffusion of the two different types of ions. Diffusion begins first for the surface ions, but ultimately diffusion spreads throughout the cluster. The phenomenon of surface melting [12], in which a portion of the surface melts and comes to equilibrium with the solid interior, was *not* observed. During melting the average kinetic energy gradually decreased, from $T \sim 1250$ K at $t \sim 8$ ps to $T \sim 950$ K at $t \sim 80$ ps. Beyond $t = 80$ ps, to $t = 105$ ps, no further decrease in T was observed.

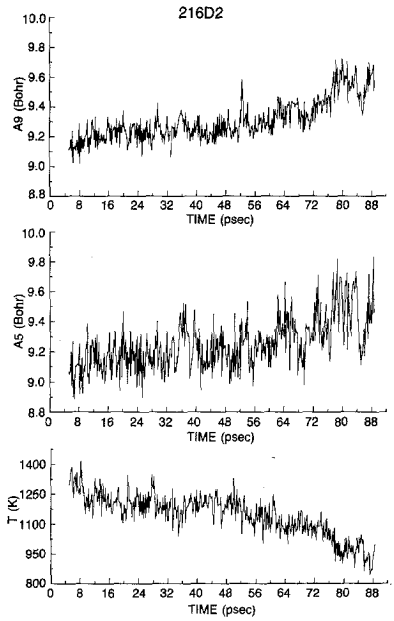

FIG. 4. Plot of the temperature $T(t)$ and the effective lattice constants A5 and A9 (see text for a complete description) during the melting for the cluster corresponding to the 14th point in Fig. 1.

Notice that beyond $t \sim 80$ ps diffusion persists throughout the sample at essentially a constant rate, which is the same for both Na^+ and F^- ions. Also during melting ($t < \sim 80$ ps) the density of the interior of the cluster decreased by about 15%, indicating a net increase in potential energy to compensate for the loss of kinetic energy. This is illustrated in Fig. 4 where A5, A9, and $T(t)$ are plotted as a function of time.

The cluster corresponding to the 13th point in Fig. 1 could possibly melt given a sufficiently long simulation. However, it remained a diffusionless ordered microcrystal after running for over 100 ps. The cluster corresponding to the 12th point can definitely be said to prefer the solid crystalline form. Energy was taken gradually (12.5 K/ps) out of the 14th-point cluster to reduce it to that of the 12th-point cluster. Further simulation at constant energy for 40 ps produced an increase in T and a corresponding increase in density to values near those for the 12th-point unmelted cluster. During this time the diffusion gradually ceased, except for a small amount of diffusion on the surface, and the bulk of the sample had evidently (based on computations of diffraction) recrystallized.

Crystallization from a supercooled liquid has previously been simulated using molecular dynamics with constant-volume periodic boundary conditions [13, 14]. The constant volume/shape constraint evidently does not impede crystallization because the resultant material (crystalline solid) prefers a smaller volume. On the

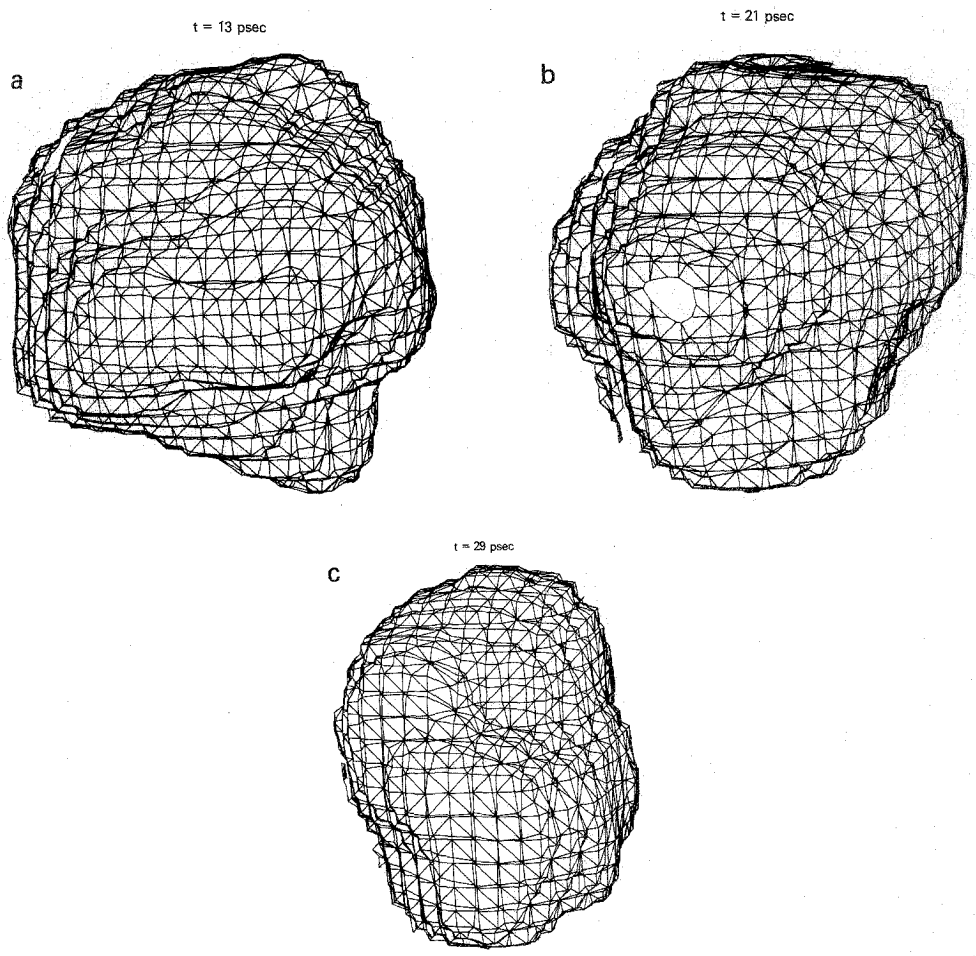

FIG. 5. Shape of the cluster for the 16th point in Fig. 1 (liquid at $T \sim 1120$ K) for three times (a) $t = 13$, (b) $t = 21$ and (c) $t = 29$ ps. The three-dimensional surfaces plotted are surfaces of constant density obtained by replacing point ion masses with suitable gaussians. The plotting routine was adapted from the tetrahedron method for plotting Fermi surfaces (Ref. [15]).

other hand, in melting, the resulting material (liquid) prefers a larger volume. Moreover, the shape of the material must be allowed to change for the system to be completely unconstrained.

We have examined changes in the shape of the cluster by monitoring the moment of inertia tensor and by generating 3 dimensional plots of the cluster at selected times. The shape of the cluster appears to be constantly changing in the liquid phase. This is illustrated in Fig. 5, where "snapshots" of the 16th-point cluster are shown at three times, 8 ps apart. The dramatic changes in shape are indicative of

the fact that liquids have essentially no resistance to shear stress. The plots in Fig. 5 were drawn by first smearing out the point ion masses with Gaussians of suitable width and then plotting a selected surface of constant density. The surface plotting program was adapted from a routine originally developed for plotting Fermi surfaces [15].

From Fig. 1 we see $T_m \sim 1200$ K for the 216-ion cluster, which is reasonably close to the experimental value for a truly macroscopic sample, 1260 K. The calculated value of T_m is probably too high: It is known, for example, that the melting point of small metal clusters (diameter of 20–30 Å) can be depressed by 20 to 30% from the bulk value [16]. And, our calculated value for the 512-ion cluster is $T_m \sim 1250$ K. Calculations are underway from the $10 \times 10 \times 10$ 1000-ion cluster. If isothermal conditions were maintained at the point where melting begins (13th point in Fig. 1) then energy would be taken in by the cluster until it transformed to a liquid at T_m. The amount of this energy is in good agreement with the experimental value for the latent heat (see Fig. 1).

These results indicate that above a certain critical energy, microcrystals of a few hundred ions, in a perfect vacuum, would transform to a homogeneous liquid at a lower temperature. This liquid state could only be achieved, under isothermal conditions, by supercooling the liquid below the melting temperature.

While this article was in preparation we became aware of similar MD calculations for alkali halide clusters carried out by Juo *et al.* [17]. They also observe a sharp kink, or "Van der Waals loop," in the E vs T curve, and find a narrow region, which would correspond to energies between the 13th- and 14th-point clusters in our calculations, where solid and liquid coexist. They identify this region, which falls midway between the maximum and minimum, with the melting temperature. As mentioned above, we cannot be sure that the 13th-point cluster would not melt, or possibly, partially melt, given a sufficiently long simulation. However, the 12th-point cluster would presumably not melt because, according to our calculations, the liquid solidifies at that energy. For this reason we do not identify the midpoint of the kink as the melting temperature. Our simulations for the 512-ion cluster exhibit a possible region of solid–liquid coexistence, but, like the 216-ion cluster, *this region is very narrow compared to the latent heat.* This is the intriguing aspect of our results on melting. For truly macroscopic systems solid and liquid phases coexist indefinitely over a large energy range which, in fact, defines the latent heat. Our results show we are a long way from achieving this state, for systems with several hundred atoms. Firm resolution of these issues will require longer simulations with larger clusters.

Sulphur Hexafluoride

Molecules of sulphyr hexafluoride, SF_6, consist of a central S atom surrounded by the six F atoms at the vertices of a perfect octahedron. When liquid SF_6 is cooled, it forms what is known as a plastic crystalline phase, intermediate between

the liquid and the truly crystalline phases. The molecules become ordered on a body-centered cubic lattice, but have considerable dynamic orientational disorder. On further cooling another phase occurs where two-thirds of the molecules become orientationally ordered while the other third remain disordered. This has been observed by low-energy electron diffraction [18], but not by neutron diffraction [19]. Further cooling produces the lowest temperature phase which is truly crystalline.

All these transitions except that from the liquid to the plastic phase have been simulated in molecular dynamics [3, 4], the transition from the plastic to the lower phases being the first such transition to be modelled on a computer [3]. The computer model for these transitions was based on lattice statics and lattice dynamics calculations [5], and involved a very simple intermolecular force model. A Lennard–Jones atom–atom potential was assumed to govern the interaction between any pair of F atoms, and by summing the forces that this potential gave, intermolecular forces and torques can easily be found. The potential function contains two parameters and these were fixed by using the cubic lattice parameter and the sublimation energy, thus giving no adjustable parameters for the model. Nevertheless the phase transitions were duly reproduced, with transition temperatures very close to those for the natural SF_6.

The model has been the basis of extensive work on the dynamics of the transition [6]. We wish to report here the evidence for these transitions observable in a droplet, or cluster, of SF_6. The size of the droplet here used is very small indeed, containing only 128 molecules, as compared with the 4096 molecules in the "infinite" sample of the earlier publications.

A spherical sample was made by taking a plastic phase configuration from the earlier work and carving out a spherical cluster containing the required number of molecules. Each F atom of each molecule of the cluster is assumed to interact with all the other F atoms in the sample, following the procedure outlined earlier. Each time step therefore involves the calculation of $(6 \times 128)^2/2$ or about 300,000 interactions, and takes about one second in the DAP. This is to be compared with the one million near-neighbor interactions per time step used for the calculations of [3, 4, 6], which took about three seconds per time step in the DAP in the matrix mode. That these are very comparable is a consequence of the fact that most of the calculation for the cluster takes place in matrix rather than vector mode.

The configuration chosen to start the cluster was at a temperature where the bulk system was in the plastic phase. Within a few minutes of computer time running the simulation at 25 K it was apparent that a single crystal of the lowest temperature phase had formed. This can be seen from the first part of Fig. 6 which is constructed as follows. Each molecule gives rise to three dots in the diagram, found by projecting the six S–F vectors onto a unit sphere. The intersections in the lower hemisphere are discarded, and those in the upper hemisphere are projected onto the equatorial plane using the equal-area projection. The dot density has been increased by taking a number of configurations separated by small time intervals.

At the time of cluster construction the mean S–F bond directions lay along the

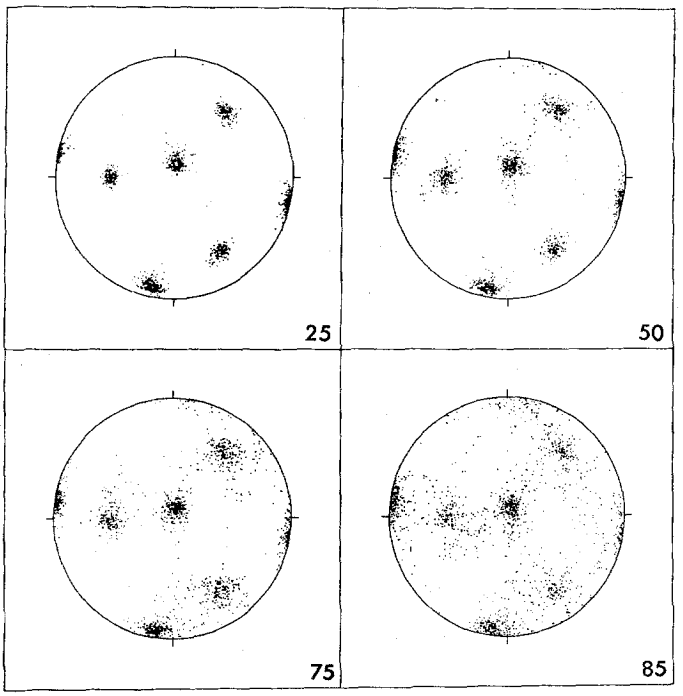

FIG. 6. Dot-plots showing the orientation of all the $S-F$ bonds throughout the whole 128 molecule cluster of SF_6. The upper hemisphere is shown, and the equal-area projection is used. The sample temperatures were 25, 50, 75 and 85 K, and the ordered structure is still apparent at 85 K.

cartesian axes, the equatorial axes being represented on the diagrams. The mean bond directions have shifted quite considerably on forming the crystal, the cluster at all times having zero angular momentum. It is apparent from the figure for 25 K that there is one set of octahedral directions with a third of the molecular population, and another set with two-thirds of the population. This is in agreement with the result where this transition was first observed [3], the crystal phase being identified as triclinic with three molecules in the primitive unit cell, two of which are related by a center of symmetry. A similar conclusion can be found from these diagrams as follows.

The centers of density on the dot-plot for 25 K can be pinpointed to within one degree and the angles between pairs of these points found by measuring along common great circles. The angles so found are shown in Fig. 7, the 90° angles between points of the same octahedra not being shown numerically. It is clear that the two octahedra are unrelated by any symmetry but that there is a point, denoted by an open circle in Fig. 7, which is the direction of a two-fold (110) axis of both octahedra. A two-fold symmetry would have to be reflected in the symmetry of the underlying crystal lattice, which is not the case.

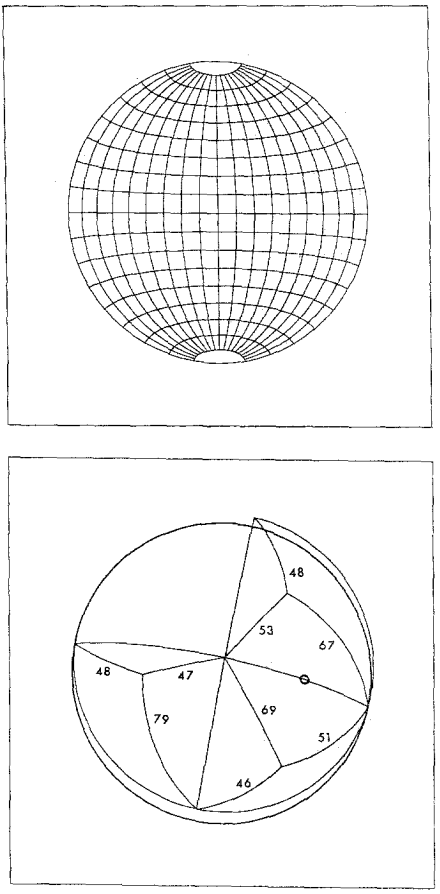

FIG. 7. The great-circle angles between the distribution centers found from Fig. 6 at 25 K. Triclinic symmetry can be deduced as there is no symmetry in the measured angles.

Following the development of the dot-plots as a function of temperature we see in Fig. 6 for 25, 50, 75, and 85 K that the molecules are experiencing increased thermal motion, but that the single crystal is persisting. Transitions are observable in Fig. 8. The clarity of the 90 K plot is not sufficient to indicate a change of phase to the intermediate structure, but it is clear that by 95 K the one-third molecular component is disordered as it must be in the intermediate phase. By 100 K the ordering is concentrated about the cartesian directions, showing reversion to the plastic cubic phase, and by 125 K all ordering has disappeared and the cluster is liquid. One solid state transition is therefore clear, and also the melting transition. The identification of the range (if any) of the analysis of these phase transitions as a function of droplet size is now being sttudied in detail.

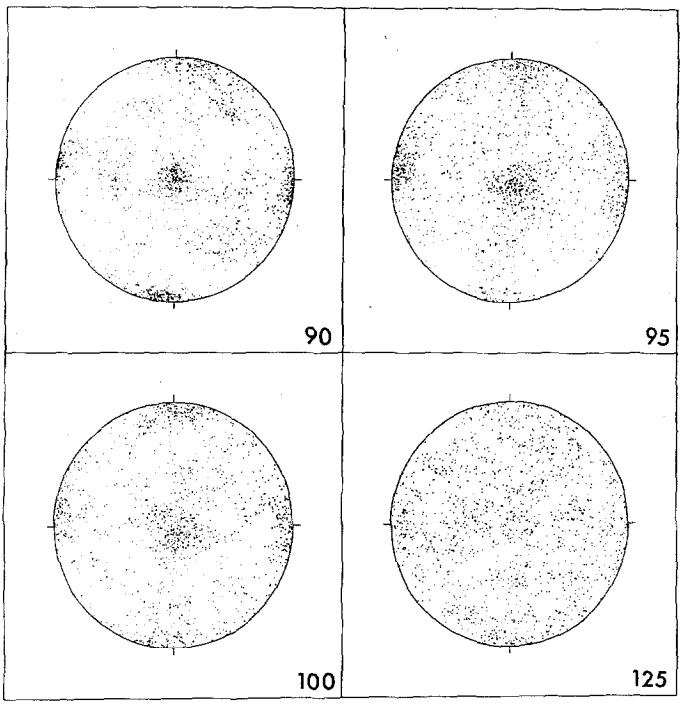

FIG. 8. Dot-plots as in Fig. 6 but for a sample heated to 90, then 95, 100, and 125 K. The ordering at 95 and 100 K corresponds to the plastic crystalline phase, and at 125 K the sample is liquid.

ACKNOWLEDGMENTS

We wish to acknowledge many helpful discussions with Dr. Andrew Brass pertaining to the algorithms discussed in Section 2, and specifically, for calling our attention to the ease with which a ring-type architecture could be applied to MD of free clusters. Support for the two DAP computers in Edinburgh from the Science and Engineering Research Council (UK) is gratefully acknowledged. One of us (LLB) would like to thank the Department of Physics at the University of Edinburgh for their kind hospitality and for providing the excellent facilities for carrying out this research.

REFERENCES

1. R. W. HOCKNEY AND C. R. JESSHOPE, *Parallel Computers* (Hilger, Bristol, UK, 1981).
2. G. S. PAWLEY AND G. W. THOMAS, *J. Comput. Phys.* **47**, 165 (1982).
3. G. S. PAWLEY AND G. W. THOMAS, *Phys. Rev. Lett.* **48**, 410 (1982).
4. G. S. PAWLEY AND M. T. DOVE, *Chem. Phys. Lett.* **99**, 45 (1983).
5. G. S. PAWLEY, *Mol. Phys.* **43**, 1321 (1981).
6. G. S. PAWLEY AND M. T. DOVE, *Helv. Phys. Acta* **56**, 583 (1983); M. T. DOVE AND G. S. PAWLEY, *J. Phys. C* **16**, 5969 (1983); M. T. DOVE AND G. S. PAWLEY, *J. Phys. C* **17**, 6581 (1984); G. S. PAWLEY AND M. T. DOVE, *Mol. Phys.* **55**, 1147 (1985); M. T. DOVE, G. S. PAWLEY, G. DOLLING, AND B. M. POWELL, *Mol. Phys.* **57**, 865 (1986).

7. K. Refson and G. S. Pawley, *Mol. Phys.*, in press; *ibid.*, in press.
8. G. S. Pawley, *Solid State Commun.* **53**, 817 (1985).
9. R. G. Gordon and Y. S. Kim, *J. Chem. Phys.* **56**, 3122 (1972).
10. L. L. Boyer, *Phys. Rev. B* **23**, 3673 (1981).
11. P. D. Pathak, J. M. Trivedi, and N. G. Vasavada, *Acta. Crystallogrr. A* **29**, 477 (1973).
12. J. Q. Broughton and G. H. Gilmer, *J. Phys. Chem.*, in press.
13. R. D. Mountain and P. K. Basu, *J. Chem. Phys.* **78**, 7318 (1983).
14. R. D. Mountain and P. K. Basu, *Phys. Rev. A* **28**, 370 (1983).
15. L. L. Boyer, *Phys. Rev. B* **19**, 2824 (1979).
16. Ph. Buffat and J-P. Borel, *Phys. Rev. A* **13**, 2287 (1976).
17. J. Luo, U. Landman, and J. Jortner, "Isomerization and Lelting of Small Alkali-Halide Clusters," Intl Symp. on the Phys. and Chem. of Small Clusters, Richmond, VA, Oct. 27–31, 1986.
18. G. Raynerd, G. T. Tatlock, and J. A. Venables, *Acta. Crystallogr. B* **38**, 1896 (1982).
19. B. M. Powell, Chalk River Nuclear Laboratories, Chalk River, Canada 1986 (unpublished).