ARMY RESEARCH LABORATORY

Near-Neighbor Calculations Using a Modified Cell-Linked List Method

by William Mattson
and Betsy M. Rice

ARL-TR-1956

June 1999

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-1956 **June 1999**

Near-Neighbor Calculations Using a Modified Cell-Linked List Method

William Mattson
University of Illinois at Urbana-Champaign

Betsy M. Rice
Weapons and Materials Research Directorate, ARL

Abstract

We have modified the conventional cell-linked list method to reduce the number of unnecessary internuclear distance calculations in molecular simulations of systems containing many particles. In the conventional method, the simulation space is partitioned into cells with edge lengths no less than the cutoff distance of the interaction potential (r_{cut}). The atoms are assigned to cells according to their spatial positions, and all internuclear distances for atoms within a cell and atoms in the same and nearest neighbor cells are evaluated. While this method ensures that the internuclear separation between all atom pairs within r_{cut} is calculated, it allows for unnecessary internuclear distance calculations between pairs that are within the volume encompassing the neighbor cells but that are separated by more than r_{cut}. The modified method presented here allows for reductions in the cell sizes and the number of atoms within the volume encompassing the neighbor cells. These reductions decrease the number of atoms that are outside of the interaction range and the number of unnecessary internuclear distance calculations while ensuring that all internuclear distances within the cutoff range are evaluated. We present algorithms to determine the volume with the minimum number of neighbor cells as a function of cell size and the identities of the neighboring cells. We also evaluate the serial performance using the modified form as functions of cell size and particle density for comparison with the performance using the conventional cell-linked list method.

Acknowledgment

This work was supported under the DOD High Performance Computing Software Support Initiative (CHSSI) for Computational Chemistry and Material Science.

INTENTIONALLY LEFT BLANK.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Popular molecular simulation techniques such as molecular dynamics or Monte Carlo are used to study the physical and chemical processes occurring in systems containing large numbers of atoms at the atomic level (Thompson 1998). These methods require evaluation of either the total potential energy of a system of N atoms (V_{Tot}) or the gradients of the potential energy. The total potential energy consists of terms that describe the various interactions among the atoms in the system. These terms are usually functions of internal coordinates, such as internuclear distances between two atoms, bond angles among three atoms, or torsional angles among four atoms. For condensed phase modeling, the total potential energy is often described as a sum of two-body interactions over all atom pairs. The interaction terms are typically simple functions of the internuclear distance r_{ij} between atoms i and j:

$$V_{Tot}=\sum_{i=1}^{N-1}\sum_{j>i}^{N}V(r_{ij}). \tag{1}$$

The evaluation of equation (1) and the gradients are usually the most computationally demanding steps in a simulation, even if the functional forms for $V(r_{ij})$ are extremely simple. Brute force evaluation of equation (1) requires the calculation of at least N (N−1)/2 internuclear distances. In a molecular dynamics simulation, each integration step often requires the evaluation of equation (1) and its gradients more than once depending on the integration scheme that is chosen (Allen and Tildesley 1990). It is clear that methods to reduce the computational burdens associated with numerous evaluations of equation (1) are required. The most obvious recent approaches are to modify the codes for scalable platforms. However, modifications of existing algorithms designed to reduce the computational burdens associated with evaluation of equation (1) can be made to increase the serial performance and exploit scalable architectures to achieve enhanced performance. In this work, we present a modification of existing algorithms that were developed to reduce unnecessary computations of the internuclear distances for atom pairs used in the evaluation of equation (1).

Common strategies to reduce the computational demands associated with equation (1) include the use of simple functions to describe the pair interaction potentials and the assumption that the interaction between two particles is negligible beyond a certain cutoff distance, r_{cut}. The assumption of a cutoff distance in the interaction potential allows for a reduction in computational time, since the interaction between atoms separated by distances exceeding r_{cut} does not need to be calculated. Unfortunately, the easiest and most direct way to determine the set of internuclear distances that are within r_{cut} is to evaluate all distances between all pairs and eliminate those that exceed r_{cut}. This step requires a potentially large number of unnecessary calculations and might be the most costly computational step in such a simulation.

The order N^2 method described in the preceding paragraph is due to the assumption of pair interaction potentials in equation (1). However, commonly used functions (such as Lennard-Jones or exp-6 forms) are too simple to correctly model all of the anisotropies that exist in many systems. Also, if chemical reactions in the condensed phase are being simulated, more sophisticated potential energy functions are required. Increasingly complex potential energy functions often use many of the internuclear distances evaluated for equation (1) more than once per evaluation of potential energy or force. An example is seen in the potential energy function used in the simulation of detonation (Rice et al. 1996). In this example, the function that describes the interaction for all atoms in the system is

$$V_{Tot} = \sum_{i=1}^{N} \sum_{j>i}^{N} \left\{ f_c(r_{ij}) \left[\left(2 - \overline{B}_{ij}\right) V_R(r_{ij}) - \overline{B}_{ij} V_A(r_{ij}) \right] + V_{vdW} \right\}, \tag{2}$$

where the first set of terms on the right-hand side of equation (2) (within the square brackets) contains the intramolecular interaction terms and includes many-body effects. The V_{vdW} term in equation (2) is a modified Lennard-Jones potential that describes the intermolecular interactions. The intra- and intermolecular interaction terms have different interaction ranges and thus sample different sets of internuclear distances out of the total set in the system. The many-body term in the intramolecular interaction portion of equation (2) has the form

$$\bar{B}_{ij} = \frac{1}{2}\left(\left\{1+G\sum_{\substack{k=1\\k\neq i,j}}^{N} f_c(r_{ik})\exp\left[m\left(r_{ij}-r_{ik}\right)\right]\right\}^{-n} + \left\{1+G\sum_{\substack{k=1\\k\neq i,j}}^{N} f_c(r_{jk})\exp\left[m\left(r_{ij}-r_{jk}\right)\right]\right\}^{-n}\right). \tag{3}$$

Evaluation of this term for a single i-j atom pair in equation (1) requires knowledge about the remaining (N−2) internuclear distances. If a brute force calculation of the entire set of internuclear distances is performed for each evaluation of the intramolecular interaction between all atom pairs during evaluation of equation (1) using a potential of the form of equation (2), this simulation becomes order N^3.

A reduction of unnecessary calculations of internuclear distances can be accomplished through the use of the Verlet neighbor list (Verlet 1967). This method requires the construction of a list of neighbors for each atom. An atom's neighbors are usually defined to be all of the atoms that are within a distance slightly greater than the range of the interaction potential. Information about the neighbors is stored in arrays. For the duration of the simulation or until the lists are updated, each atom is assumed to interact only with the atoms on its neighbor list. The internuclear distances, interaction potentials, and forces are evaluated for each atom and its neighbors only. The list may be periodically updated to allow for the movement of atoms into or out of the interaction range. Brute force construction or update of the list requires the evaluation of all N(N−1)/2 internuclear distances. The method has been shown to be efficient when the system contains a relatively small number of atoms (Allen and Tildesley 1990; Morales, Rull, and Toxvaerd 1989). However, as the system becomes larger, the memory requirements for maintaining the neighbor lists become prohibitive. Also, as the mobility of the atoms becomes greater, either the frequency of lists updates must increase or the cutoff distance used in the definition of the neighbors must increase. Either of these requirements increases the computational demands of the Verlet neighbor list method. The example of the detonation simulation is one such case in which the mass flow (moving at supersonic speeds) would require large neighbor cutoff distances and frequent neighbor list updates (Rice et al. 1996).

Alternative methods for the efficient determination of the interacting neighbors for each atom include grid or cell approaches (Allen and Tildesley 1990; Boris 1986; Lambrakos and Boris 1987; Brugè 1993). These approaches partition the simulation space into grids or cells, to which the atoms are assigned by virtue of their positions relative to the cells. Since each cell has an unchanging set of neighboring cells that contain the volume within the distance r_{cut} of that cell, an atom associated with one of the cells has as its neighbors those atoms assigned to the same or neighboring cells. The implementations of these methods usually assign the atoms to the cells at each integration step. However, the same considerations used for the frequency of updating the Verlet neighbor lists are applicable here. There is some overhead associated with these methods, and they are preferable only for systems that contain more than 1,000 atoms (Allen and Tildesley 1990). These methods substantially reduce the number of unnecessary internuclear distance calculations in evaluating equation (1) but do not completely eliminate unnecessary computations.

In this work, we report modifications to grid-cell methods to further reduce the number of internuclear distance calculations in systems containing larger numbers of atoms. The approach we present is a modification of the conventional method of cell-linked lists as described in detail by Allen and Tildesley (1990). The results show a dramatic decrease in CPU requirements and are amenable to parallelization.

Brugè and coworkers have already provided geometric and systolic parallelization schemes for conventional implementations of Verlet neighbor lists and conventional cell-linked lists (Brugè 1993; Brugè and Fornili 1990a, 1990b). These have shown significant decreases in computation times, and we refer the readers to such information. Our intent here is to modify the algorithms to accelerate both serial and scalable performance. We describe the modifications and demonstrate the performance on serial platforms in this work. Future work will focus on scalability and further modifications to enhance performance. We are confident that some of the scalable methods set forth by Brugè and coworkers (Brugè 1993; Brugè and Fornili 1990a, 1990b) will be applicable to these algorithms.

2. Method of Cell-Linked Lists

2.1 Conventional Method. The conventional method of cell-linked lists is well described by Allen and Tildesley (1990). We, like they, describe our variation of the method in two dimensions, but the method can be generalized to include three dimensions. The modification we present is similar to one suggested by Allen and Tildesley (1990)—that the cell size be reduced so that no more than one atom can occupy a cell.

In the conventional method of cell-linked lists, the simulation space is partitioned into cells, the edges of which being no smaller than the cutoff distance of the interaction potential. The atoms are then assigned to the various cells by virtue of their position in the simulation space. A linked list of the atom indices is created during the sorting procedure. Also, at the beginning of a simulation, an array that contains a list of cell neighbors for each cell is created. The list remains fixed unless the simulation space changes during the simulation (see, for example, Rice et al. [1996]).

A cell i_{cell} has as its neighbors any cell that contains at least one point that is within the distance r_{cut} of any point within i_{cell}. Since the conventional method requires that the edges of each cell be no smaller than r_{cut}, each cell has eight nearest neighbors (we are assuming periodic boundary conditions in both dimensions of our two-dimensional example). These requirements ensure that all atoms that are within the interaction range of any atom within i_{cell} are assigned to the eight nearest-neighbor cells of i_{cell} or i_{cell} itself. All atoms occupying cells other than these are outside the interaction range of any atom located within i_{cell}. Figure 1 illustrates the division of a region of the simulation space into cells. In this figure, both the x and y cell edges (denoted as l_x and l_y hereafter) equal r_{cut}. Evaluation of equation (1) occurs through looping over the cells using the linked list of atoms rather than accessing the atom indices sequentially as written in equation (1).

This method dramatically reduces the number of unnecessary internuclear distance calculations that would result from a brute force calculation of all N(N−1)/2 internuclear distances. However, modifications can be made to further reduce the number of unnecessary distance calculations. In the conventional method, the distances between all atom pairs located within the rectangular area of $9l_xl_y$ are calculated. Assuming the limiting case $l_x = l_y = r_{cut,}$ the area within which all distances are calculated is $9r_{cut}^2$. The area within the cutoff radius for a single atom is only $\pi^*r_{cut}{}^2$. Thus, the traditional cell-linked list method calculates distances between all atom pairs within an area that is almost three times larger (or more, since $l_i \geq r_{cut}$, where i = x or y) than that actually required for an atom. This dramatic difference is illustrated by comparing the area within the shaded circle centered on the atom labeled "T" with the area for the cell containing T and its neighboring cells in Figure 1. The shaded circular area illustrates the range of interaction for atom T. Implementation of the conventional cell-linked list for this example would result in nine unnecessary internuclear distance calculations.

2.2 Modified Method of Cell-Linked Lists. The main modification of the method is in the definition of the sizes of the cells. By dividing the simulation space into smaller rectangular cells, each atom is surrounded by a group of cells that better approximates the area of interaction for that atom. For example, in Figure 2, we have divided the original rectangular cells from Figure 1 into fourths. The length of each cell is now $1/2r_{cut}$. The neighboring cells to that containing the labeled atom T are the surrounding first and second shells of cells. The area for this set of neighboring cells is $6.25r_{cut}^2$, which is approximately one-third smaller than the rectangular area that would be considered in the conventional approach (see Figure 1). Also, the number of unnecessary internuclear distance calculations has been reduced to four. However, the number of neighboring cells to that containing atom T has increased from 8 to 24. Thus, the area has been substantially reduced, but the number of neighboring cells has increased by a factor of 3. There is an increase in memory requirements associated with the linked lists upon increasing the number of neighboring cells.

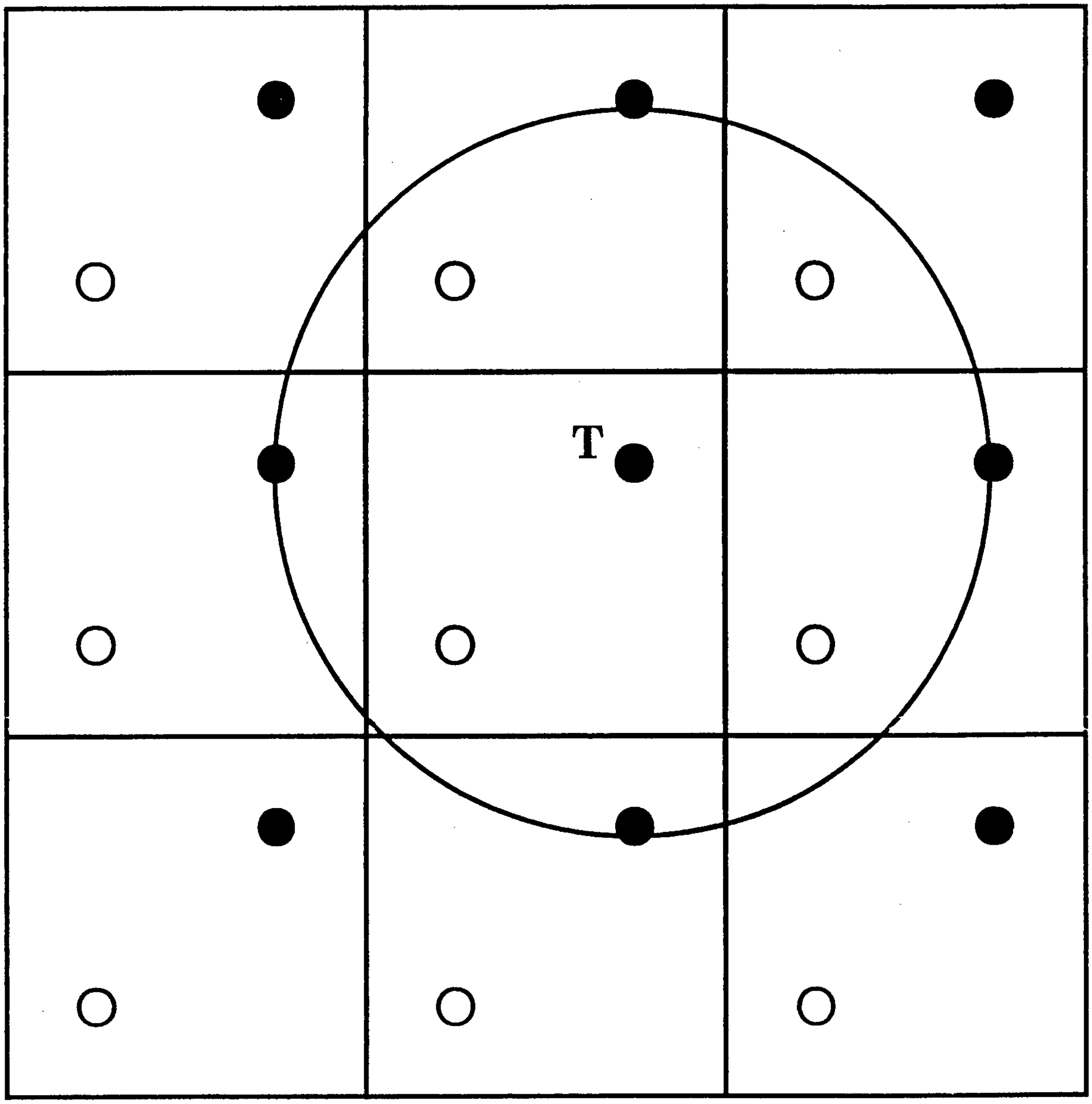

Figure 1. Illustration of the Conventional Cell Method in Two Dimensions; Simulation Box Is Partitioned Into 3×3 Square Cells; Edge Length of Each Cell Is r_{cut}. The Shaded Circle Centered on Atom T Has Radius r_{cut} and Denotes the Range of Interaction for Atom T. In This Method, the Eight Outer Cells Are Considered Neighbors of the Central Cell That Contains Atom T.

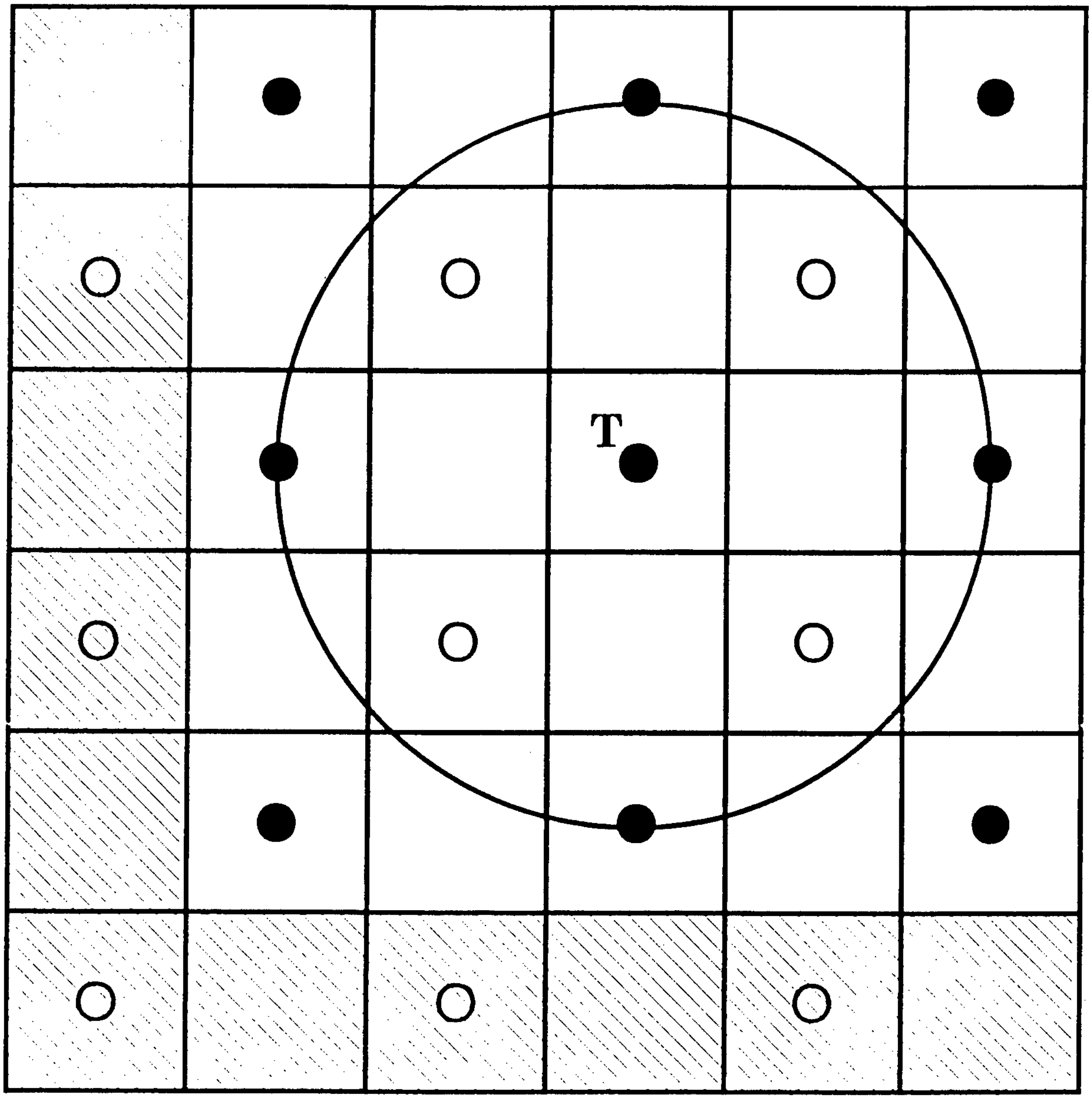

Figure 2. Illustration of the Conventional Cell Method in Two Dimensions; Simulation Box Is Partitioned Into 6×6 Square Cells; Edge Length of Each Cell Is r_{cut}. The Hatched Area Denotes the Cells That Are Not Considered to Be Neighbors of the Cell Containing Atom T in the Modified Cell-Linked List Method.

In the simple examples shown in Figures 1 and 2, the division of the original cell size into fourths has reduced the number of unnecessary internuclear distance calculations from nine to four, and a further reduction in cell size would probably not result in additional savings. A system whose atoms are arranged such that the density is not uniform might benefit from further reduction of the cell sizes to the point that the sphere of interaction of an atom is closely approximated by a set of small rectangular cells. Such an example is given in Figure 3, which has overlaid the positions of atoms behind a detonation wave (Rice et al. 1996) (a high dense region) onto a grid of cells with edge lengths $l_x = l_y = 1/20r_{cut}$. It is clear that use of cells with the sizes shown in Figures 1 or 2 would require many unnecessary internuclear distance calculations for the atomic arrangement in Figure 3.

Note that many of the neighboring cells to that containing the labeled atom T in Figure 3 are empty. There is overhead associated with determining whether a cell is occupied. Also, more memory is required to maintain the cell-linked list and the neighbor list, since as the cell sizes decrease the number of cells and the number of neighboring cells for each cell increase. While this method reduces the unnecessary distance calculations, there is a point at which the reduction in the size of the cell requires more computation in overhead than it saves in eliminating unnecessary distance calculations. The optimum cell size might vary from machine to machine and implementation to implementation. Therefore, it is desirable to use an algorithm that allows for the cell size to be changed easily to accommodate portability.

2.3 Offset Mapping Method. As cell sizes decrease, memory requirements for storage of neighbor information increase and are potentially a limitation on the use of the modified cell-linked list scheme. This problem can be reduced by the determination of neighboring cells through a list of relative cell index offsets, similar in spirit to that presented in the Monotonic Logical Grid (MLG) approach (Boris 1986; Lambrakos and Boris 1987). After partitioning the simulation space into cells, each cell is assigned a grid cell index (i,j,k) that corresponds to its location in a Cartesian reference frame (x,y,z). Figure 4 illustrates the two-dimensional grid overlaid on the simulation box shown in Figure 2. In this example, the grid indices are assigned

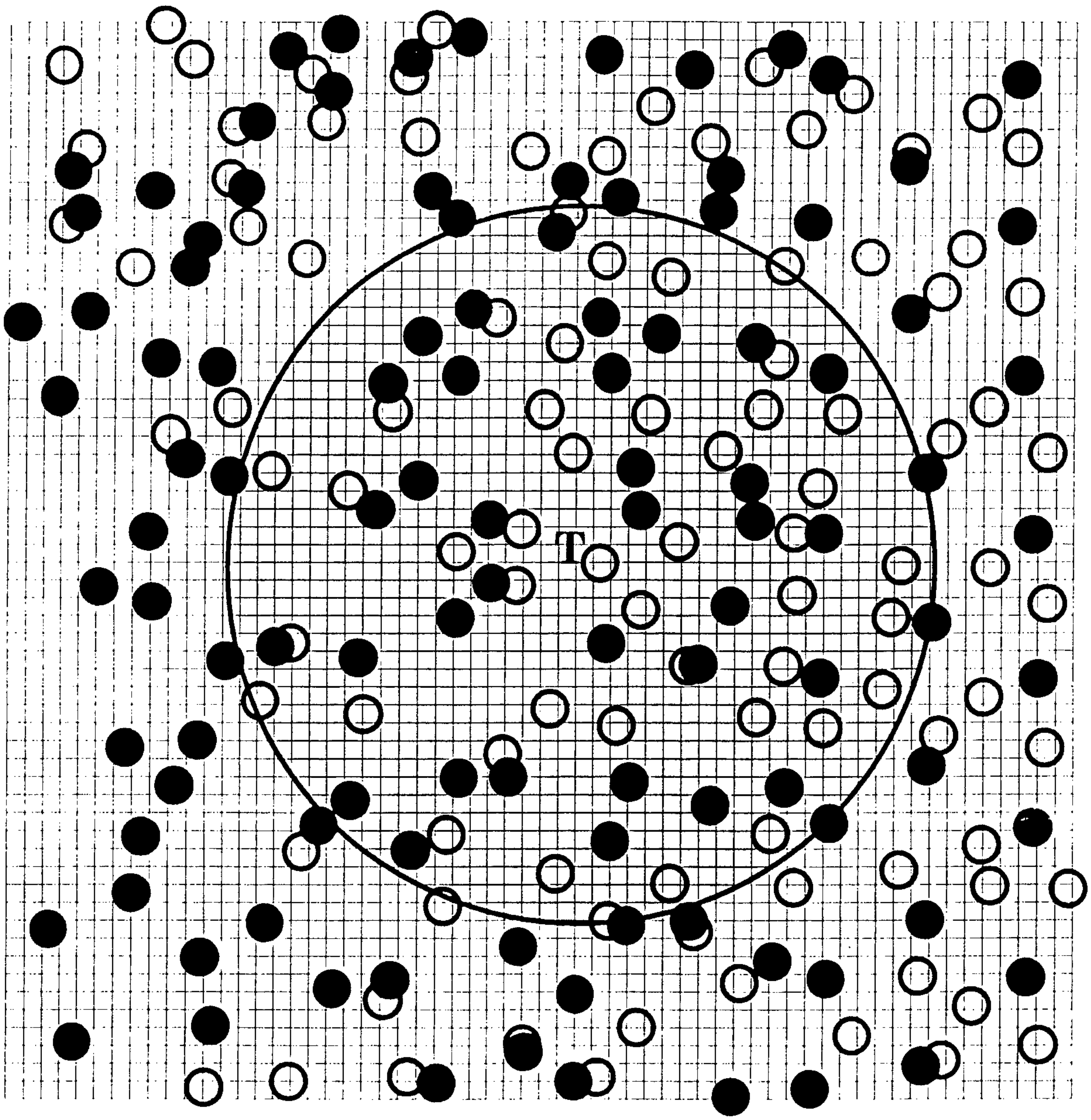

Figure 3. Simulation Box Partitioned Into 60×60 Square Cells; Edge Length of Each Cell Is $1/20r_{cut}$. The Atoms That Are Illustrated on This Grid Were Taken From Results of a Molecular Dynamics Simulation of Detonation and Correspond to the High Dense Region Behind the Detonation Front (Rice et al. 1996).

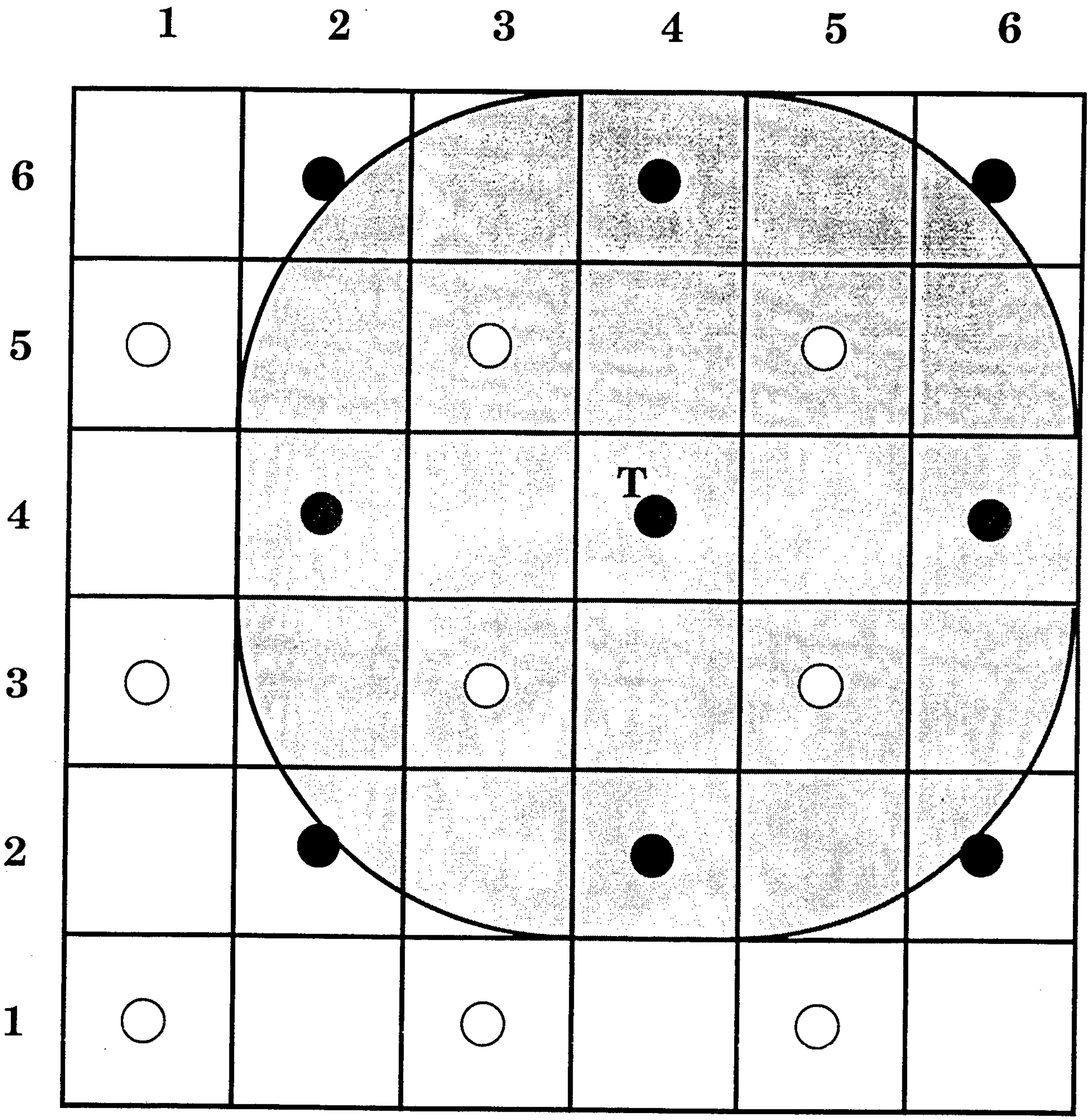

Figure 4. Illustration of the Conventional Cell Method in Two Dimensions; Simulation Box Is Partitioned Into 6×6 Square Cells; Edge Length of Each Cell Is 1/2r_{cut}; "Shaped" Neighbor Region (Shaded Area) Illustrated.

relative to cell (1,1), located at the lowermost cell on the left-hand side of the figure. The cell that contains the labeled atom T is located at the fourth column (x direction) and the fourth row (y direction). Thus, the grid index for this cell is (4,4). The set of cells that are within the interaction range (r_{cut}) for all points in cell (4,4) consists of the first and second nearest neighbors, each of which has a set of grid indices that can be described as relative offsets to (4,4). Each cell in the simulation box has the same set of relative grid index offsets as (4,4). This set can be stored in a relative offset array, which is illustrated for this example in the upper portion of Figure 5.

Determination of the relative cell index offsets of the neighbors is straightforward, particularly if the area encompassing the neighbors is rectangular. In this example, the rectangular area containing all neighboring cells has dimensions of $2r_{cut} + l_i$, where i = x or y. However, the shape of the area containing the neighboring cells is not limited to a rectangle. Further reductions in unnecessary distance calculations can result if the area containing the neighboring cells resembles a circle. Since the set of neighbors must contain all of the area within the interaction range of any point within the cell, we want the minimum set of cells that make up this "neighbor region." Rounding the corners of the rectangular neighbor region will shape the neighbor region to approximate a circle. Again, using our simple example, we illustrate this in the shaded portion of Figure 4. The rounded corners represent the portions of circles with radius r_{cut} that are centered on the corners of the cell that contains T. In this example, the number of cells containing the neighbor region is the same as that of the rectangular area. However, as the cell sizes are reduced, the number of cells containing the neighbor region will be less than those of the corresponding rectangular area, and the set of cells contained in the neighbor range will more closely approximate a circle.

To determine the minimum number of cells contained in the neighbor region, we first assume a rectangular simulation box that is larger than twice the cutoff radius in all dimensions. The box is then partitioned into cells of a desired size. At this point, assume that the central cell in this box has the grid index (0,0). Only the neighbor cells contained in one quadrant of this simulation

-2,2	-1,2	0,2	1,2	2,2
-2,1	-1,1	0,1	1,1	2,1
-2,0	-1,0	0,0	1,0	2,0
-2,-1	-1,-1	0,-1	1,-1	2,-1
-2,-2	-1,-2	0,-2	1,-2	2,-2

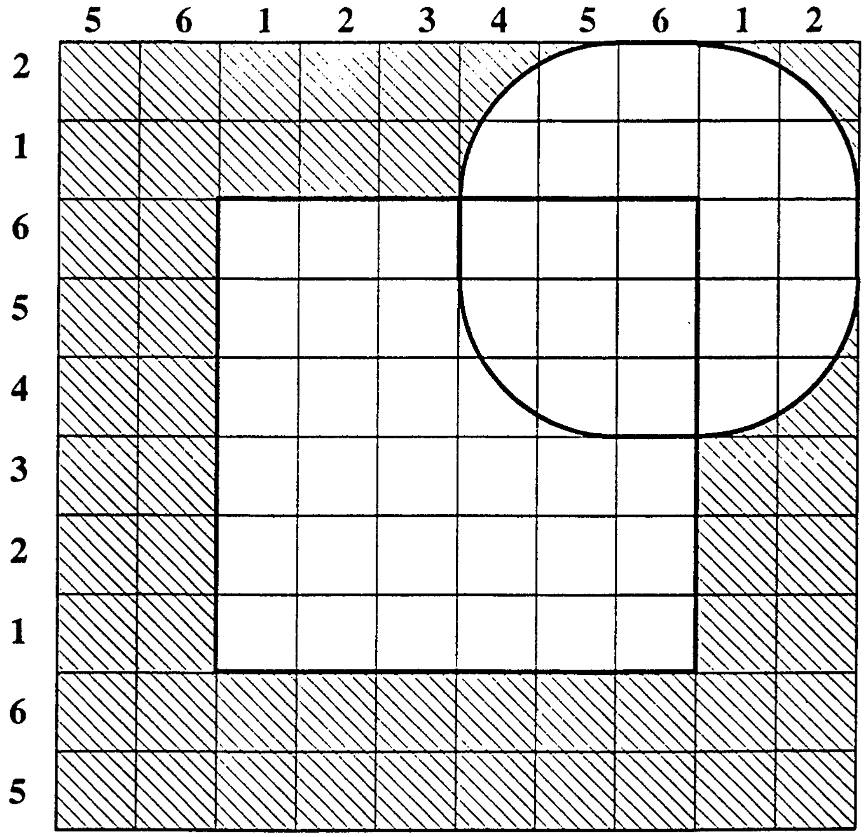

Figure 5. Geometric Representation of the Offset List, With the Relative Offset Numbers (Upper Frame), and an Illustration of the Same Simulation Box as in Figure 4, Surrounded by "Ghost" Cells (Hatched Area) (Lower Frame). This Is the Geometric Representation of the Mapping Array. The Numbers Along the Left-Hand Side and Top of the Figure Indicate the Packing of the Mapping Array.

box need to be identified, since the remaining neighboring cells that occupy the other quadrants can be generated from symmetry. We consider the top right-hand quadrant in our illustration.

The process of identifying the neighbor cells in this quadrant begins with the calculation of the range of the cells along the x axis. The grid index for the furthermost neighbor cell in the x direction, xlen, is defined as xlen = floor (r_{cut} / l_x+1). Each cell (ix,0) that is between cells (0,0) and (xlen,0) is a neighbor of (0,0). [Due to the symmetry of the system, all cells (jx,0) that include or are between (0,0) and (−xlen,0) are also neighbor cells.] We then determine the range in the y direction as follows. For each cell (ix,0) including or between (0,0) to (xlen,0), the grid index for the furthermost neighbor cell in the y direction from cell (ix,0) is defined as ylen(ix) = floor [sqrt(r_{cut}^2 - [(ix−1) * l_x]^2)/l_y + 1]. All cells that include or are between (ix,0) to [ix,ylen(ix)] are added to the list of neighbor cells. The process ensures that any cell whose lower left-hand corner is less than r_{cut} from the upper right corner of the cell (0,0) is a neighbor. A sample FORTRAN code for this process is given in the Appendix.

In a simulation, when a cell with grid index (i,j) is selected, the neighboring cells are identified by simply adding the relative cell index offsets that are determined at the beginning of the simulation to the cell grid index (see Figure 5 for the simple example presented in this work). If only half of the neighbors are required in the calculations, only the offsets in the first and second quadrants should be used, except those from (−xlen,0) to (−1,0). In our example here, in which we have found the neighbors in the upper right-most quadrant, we may just add the offsets from (−ix,1) to [−ix, ylen(ix)] with ix = −1 to xlen.

This method as described up to this point is sufficient for determining neighbors for cells that are far enough from the edges of the simulation box such that none of the neighbors should be minimum images. However, for cells on or near the edge of the simulation box, the method fails. Again, we use our simple example described in Figure 2. In Figure 5, we have reproduced the simulation box of Figure 2 and surrounded it with a shell of "ghost" cells (hatched area) that is two cells deep in both dimensions. Overlaying the offset list (upper portion of Figure 5) on cell (6,6), which is the geometric equivalent of simply adding the offset list indices to (6,6), would result in identifying as neighboring cells those with grid indices ranging from (4,4) to (8,8). This is clearly wrong because the cells with indices greater than 6 are not defined. To remedy this, a mapping array has been developed to correctly map the two-dimensional relative grid cell index offsets to the appropriate set of neighbor cells while properly taking into account the boundary conditions of the simulation. The mapping array for the simple example given in this report (assuming periodic boundary conditions in both directions) is shown in the bottom portion of Figure 5. It is constructed using the column and row designators that border the top and left-hand side of the two-dimensional array in the bottom portion of Figure 5. We illustrate its use as follows. In this simple example, one of the neighbor cells has relative cell index offset (0,2). Adding the relative cell index offset (0,2) to cell (6,6) addresses mapping array element (6,8). According to the mapping scheme, the element (6,8) of the mapping array contains the grid cell index (6,2). Cell (6,2), which corresponds to relative cell index offset (0,2), is the appropriate neighbor for cell (6,6) according to the periodic boundary conditions established for this example. As for the neighbor list, this map array increases in size with decreasing cell size.

By combining the neighbor offset list with the mapping array, the computation of and memory used for storing the neighbor information are kept at reasonable levels, even for very small cell sizes. There are four major arrays associated with this method. These are the list, the overlay, the listhead (which contains the index of the particle that is used to address the element of the linked-list array) (Allen and Tildesley 1990), and the map arrays. The size of the list array always equals the number of atoms. The size of the overlay array is proportional to the interaction range divided by the volume of the cell. The size of the listhead array equals the number of cells, and the size of the mapping array equals the total number of cells and ghost cells. When the number of cells is larger than the number of atoms, then the listhead and mapping arrays require the most memory. However, this method becomes inefficient before the number of cells equals the number of atoms. Therefore, in any reasonable use of this method, the list array has the largest memory requirement.

2.4 Distance Lists. As noted earlier, often more complex functions reuse information in the evaluation of equation (1), such as those systems that use potentials described in equations (2) and (3). This form of interaction potential requires that the internuclear distances be used many times in a single evaluation of the potential energy or forces. Recalculations within a single integration step significantly increase the computer time required for a molecular dynamics simulation. To overcome this problem, we implemented lists that contain information about atom pairs for reuse in the evaluation of the interaction potential and forces. This would be unnecessary for models that assume pair-additive interaction potentials such as the Lennard-Jones or exp-6 potentials, since the internuclear distances for each pair are used only one time per evaluation of forces. But for functions such as those presented in equations (2) and (3), there are several terms that could benefit from storage of the internuclear distances. These include the $\exp(-g_{ik})$ terms, the $f(r_{ik})$ terms, corresponding derivatives x_{ik} and y_{ik}, distance r_{ik}, and the atom index of the neighbor. This information can be generated before or during every call to the potential energy and force subroutine using the linked-list method and neighbor list. If the distance is within the intramolecular interaction range, all information that can be reused is calculated and stored. Given an atom pair ij, the stored information corresponding to that pair can easily be accessed during the evaluation of the potential energy and forces for that pair. Since the number of atom pairs can be large compared to the number of atoms, blocking techniques can be used for the storage of the atom pair information to minimize the memory required to store the atom pair information. In a blocking method, the atom pair information is calculated and stored for only a small number of cells in the simulation space at a time. The potential for these cells is calculated, and as the atom pair information is no longer needed, it is replaced by atom pair information for other nearby cells that are used next.

3. Results

Although the description of the procedure given in the preceding section is given in two-dimensional terms, the method is tested herein in a three-dimensional application. Six cubic simulation boxes that differ in size have been chosen to evaluate this methodology. The six simulation boxes consist of 27 (3×3×3), 64 (4× 4×4), 125 (5×5×5), 216 (6×6×6), 343 (7×7×7), and 512 (8×8×8) cubic cells. Each cell has edge lengths just greater than r_{cut}. The different simulation boxes are denoted hereafter as simulation box 3, 4, 5, 6, 7, and 8, respectively. All calculations were performed serially on an SGI Onyx with four 195-MHz R10000 processors with 1.5 GB of main memory and 4 MB of secondary cache per processor.

The CPU time used to evaluate the internuclear distances using this method as a function of system size and particle density is given in Table 1. For the evaluation using the cell-linked list methods, we report only the times for actual evaluation and do not include any initialization. The initialization, which includes setting up the mapping array and the relative cell offset list, is relatively fast and is only done once. The times reported are the averages for 20 separate evaluations of neighbors, and the timings include the construction of the linked lists for each evaluation. To check the method, all atom pairs were calculated and compared to those calculated through from the brute force method. In Table 1, the variable N_{div} denotes the number of divisions along an edge of the simulation box. For example, for simulation box 3, $N_{div} = 3$ partitions each of the three edges of the box into three sections. The simulation box has a total of 27 cells. This value of N_{div} corresponds to the conventional cell-linked list method. $N_{div} = 0$ indicates that the cell-linked list method has not been used, and all $N(N-1)/2$ internuclear distances are calculated. The calculations for $N_{div} = 0$ are denoted as "brute force" calculations.

It has been established that the conventional cell-linked list method is superior to the brute force approach for systems in which the dimensions are large compared to the cutoff radius of the potential (Brugè 1993). We have seen the same result in this study. Table 1 gives the times for evaluation of the internuclear distances as a function of particle density and N_{div} for the six simulation boxes and the different methods. For simulation box 3, the execution times of the conventional and modified cell-linked list methods for low densities are greater than that of the brute force method. At higher densities, there is a slight speed-up using the modified cell-linked list method over the brute force approach. Note that for all densities for simulation box 3, the conventional method is slower than the brute force method. For systems that are larger than simulation box 3, however, the performance of the conventional and modified cell-linked list methods given here are superior to that of the brute force method. For the largest simulation box (box 8), there is a 90–97% reduction in CPU time over the brute force method.

Table 1. Time (in Milliseconds) Required to Evaluate Internuclear Distances for Systems of Different Sizes and Particle Densities

N_{div}	No. of Atoms Per Cell											
	27		64		125		216		343		512	
	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]
Simulation Box consisting of 3×3×3 cubic cells												
0[b]	65	44.0	370	42.3	1,445	41.6	4,418	40.2	11,133	40.4	24,996	39.6
3[c]	116	0	641	0	2,475	0	7,386	0	18,684	0	41,406	0
6	71	38.8	403	37.1	1,438	41.9	4,401	40.4	10,741	42.5	23,920	42.2
9	82	29.3	336	47.6	1,073	56.6	3,473	53.0	8,577	54.1	18,608	55.1
12	98	15.5	407	36.5	1,294	47.7	3,324	55.0	7,760	58.5	16,626	59.8
15	146	-25.9	503	21.5	1,406	43.2	3,409	53.8	7,485	59.9	15,116	63.5
18	201	-73.3	615	4.1	1,619	34.6	3,877	47.5	7,931	57.6	15,863	61.7
21	258	-122.4	782	-22.0	1,972	20.3	4,451	39.7	9,268	50.4	17,833	56.9
Simulation Box consisting of 4×4×4 cubic cells												
0[b]	339	-23.7	1,926	-25.8	7,368	-25.9	22,238	-26.3	55,979	-26.6	125,085	-27.1
4[c]	274	0	1,531	0	5,853	0	17,614	0	44,220	0	98,443	0
8	193	29.6	959	37.4	3,504	40.1	10,424	40.8	25,934	41.4	57,450	41.6
12	203	25.9	799	47.8	2,807	52.0	8,371	52.5	20,405	53.9	44,077	55.2
16	271	1.1	1,064	30.5	3,061	47.7	7,869	55.3	18,126	59.0	40,565	58.8
20	364	-32.8	1,222	20.2	3,475	40.6	8,135	53.8	17,743	59.9	36,886	62.5
24	489	-78.5	1,454	5.0	3,890	33.5	9,500	46.1	19,406	56.1	38,539	60.9
28	667	-143.4	1,914	-25.0	4,704	19.6	11,194	36.4	23,138	47.7	44,388	54.9
Simulation Box consisting of 5×5×5 cubic cells												
0[b]	1,248	-136.4	7,056	-137.0	27,028	-136.2	80,968	-137.0	204,402	-137.0	457,627	-137.0
5[c]	528	0	2,977	0	11,442	0	34,159	0	86,245	0	193,099	0
10	374	29.2	1,874	37.1	6,865	40.0	20,249	40.7	50,833	41.1	112,994	41.5
15	397	24.8	1,735	41.7	5,627	50.8	16,318	52.2	39,879	53.8	86,837	55.0
20	552	-4.5	2,102	29.4	6,000	47.6	15,496	54.6	35,812	58.5	78,217	59.5
25	719	-36.2	2,382	20.0	6,768	40.8	16,008	53.1	35,042	59.4	71,033	63.2
30	964	-82.6	2,907	2.4	7,858	31.3	18,657	45.4	40,285	53.3	78,124	59.5
35	1,346	-154.9	3,798	-27.6	9,670	15.5	21,814	36.1	46,529	46.1	84,956	56.0

[a] Percent reduction of execution time relative to that using the conventional cell-linked list method.

[b] Brute Force method (see text).

[c] Conventional cell-linked list method (see text).

Table 1. Time (in Milliseconds) Required to Evaluate Internuclear Distances for Systems of Different Sizes and Particle Densities (continued)

N_{div}	No. of Atoms Per Cell											
	27		64		125		216		343		512	
	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]	Time	% Red.[a]
Simulation Box consisting of 6×6×6 cubic cells												
0[b]	3,688	-299.1	20,721	-304.6	79,091	-302.4	237,039	-299.0	601,276	-303.9	1,355,015	-305.0
6[c]	924	0	5,121	0	19,654	0	59,413	0	148,883	0	334,566	0
12	663	28.2	3,198	37.6	11,988	39.0	35,114	40.9	87,566	41.2	195,512	41.6
18	771	16.6	2,991	41.6	9,928	49.5	28,189	52.6	68,481	54.0	150,101	55.1
24	973	-5.3	3,618	29.3	10,518	46.5	26,943	54.7	62,025	58.3	135,247	59.6
30	1,278	-38.3	4,080	20.3	11,851	39.7	27,760	53.3	60,281	59.5	123,254	63.2
36	1,683	-82.1	5,214	-1.8	13,577	30.9	34,181	42.5	69,848	53.1	136,088	59.3
42	2,347	-154.0	6,876	-34.3	16,745	14.8	38,511	35.2	80,343	46.0	148,248	55.7
Simulation Box consisting of 7×7×7 cubic cells												
0[b]	9,174	-527.9	51,667	-537.5	197,044	-531.1	594,673	-538.8	1,515,307	-542.5	3,395,131	-540.3
7[c]	1,461	0	8,105	0	31,224	0	93,086	0	235,849	0	530,254	0
14	1,055	27.8	5,032	37.9	19,108	38.8	55,661	40.2	139,306	40.9	310,958	17.0
21	1,221	16.4	4,787	40.9	15,816	49.3	44,021	52.7	108,887	53.8	238,017	51.3
28	1,548	-6.0	5,770	28.8	16,736	46.4	42,627	54.2	98,612	58.2	210,469	62.7
35	2,017	-38.1	6,544	19.3	18,880	39.3	43,812	52.9	97,988	58.5	200,075	67.0
42	2,665	-82.4	8,243	-1.7	22,687	27.3	54,302	41.7	110,560	53.1	213,922	68.7
49	3,740	-156.0	10,937	-34.9	27,262	12.7	60,889	34.6	127,843	45.8	234,908	66.5
Simulation Box consisting of 8×8×8 cubic cells												
0[b]	20,360	-840.4	114,378	-823.6	438,915	-841.9	1,329,823	-848.9	3,403,575	-866.0	7,888,051	-905.7
8[c]	2,165	0	12,384	0	46,599	0	140,151	0	352,334	0	784,365	0
16	1,570	27.5	8,025	35.2	28,340	39.2	82,872	40.9	208,860	40.7	456,941	41.7
24	1,819	16.0	7,494	39.5	23,296	50.0	66,534	52.5	163,040	53.7	355,385	54.7
32	2,312	-6.8	9,027	27.1	24,763	46.9	63,588	54.6	147,941	58.0	322,611	58.9
40	3,010	-39.0	9,976	19.4	28,035	39.8	65,405	53.3	146,368	58.5	303,279	61.3
48	4,107	-89.7	12,579	-1.6	33,962	27.1	81,331	42.0	165,208	53.1	326,885	58.3
56	5,861	-170.7	16,931	-36.7	40,796	12.5	91,196	34.9	191,219	45.7	358,214	54.3

[a] Percent reduction of execution time relative to that using the conventional cell-linked list method.

[b] Brute Force method (see text).

[c] Conventional cell-linked list method (see text).

Further comparison of the modified method is made to the conventional cell-linked results rather than those using the brute force method. Table 1 provides a percent reduction in time using the modified cell-linked list method over the conventional method, and Figure 6 provides an illustration. Each curve in each frame of Figure 6 shows that the percent time reduction first increases with increasing N_{div}, then decreases as N_{div} becomes larger. The subsequent decrease in performance with increasing values of N_{div} becomes more pronounced for systems with low particle densities. For example, the curves for densities of 27 and 64 particles per cell show that the modified cell-linked list method is much slower than the conventional method at large values of N_{div}. Conversely, the percent time reduction at large values of N_{div} for high densities (>343 particles per cell) is only slightly less than the maximum value, indicating further time reduction does not necessarily occur with increased partitioning of the simulation space (reduction in cell size). This effect suggests that although the number of unnecessary internuclear distance calculations is decreasing with increasing N_{div} (see Table 2), the computational costs for the overhead associated with using a smaller cells are increasing and will eventually outweigh the savings realized from the reduced number of internuclear distance calculations.

4. Conclusions

It is clear that as advances in scalable architectures continue, more sophisticated molecular simulations requiring more atoms and more complex interaction potentials will be attempted. It is because of this expectation that we have modified the traditional cell-linked list method to reduce unnecessary internuclear distance calculations for larger and more complex systems. We have shown a significant increase in speed of the evaluation of information needed for a molecular simulation through the reduction of unnecessary internuclear distance calculations. Although we have developed this algorithm for acceleration on serial machines, future efforts will invoke strategies for further increased performance on scalable architectures.

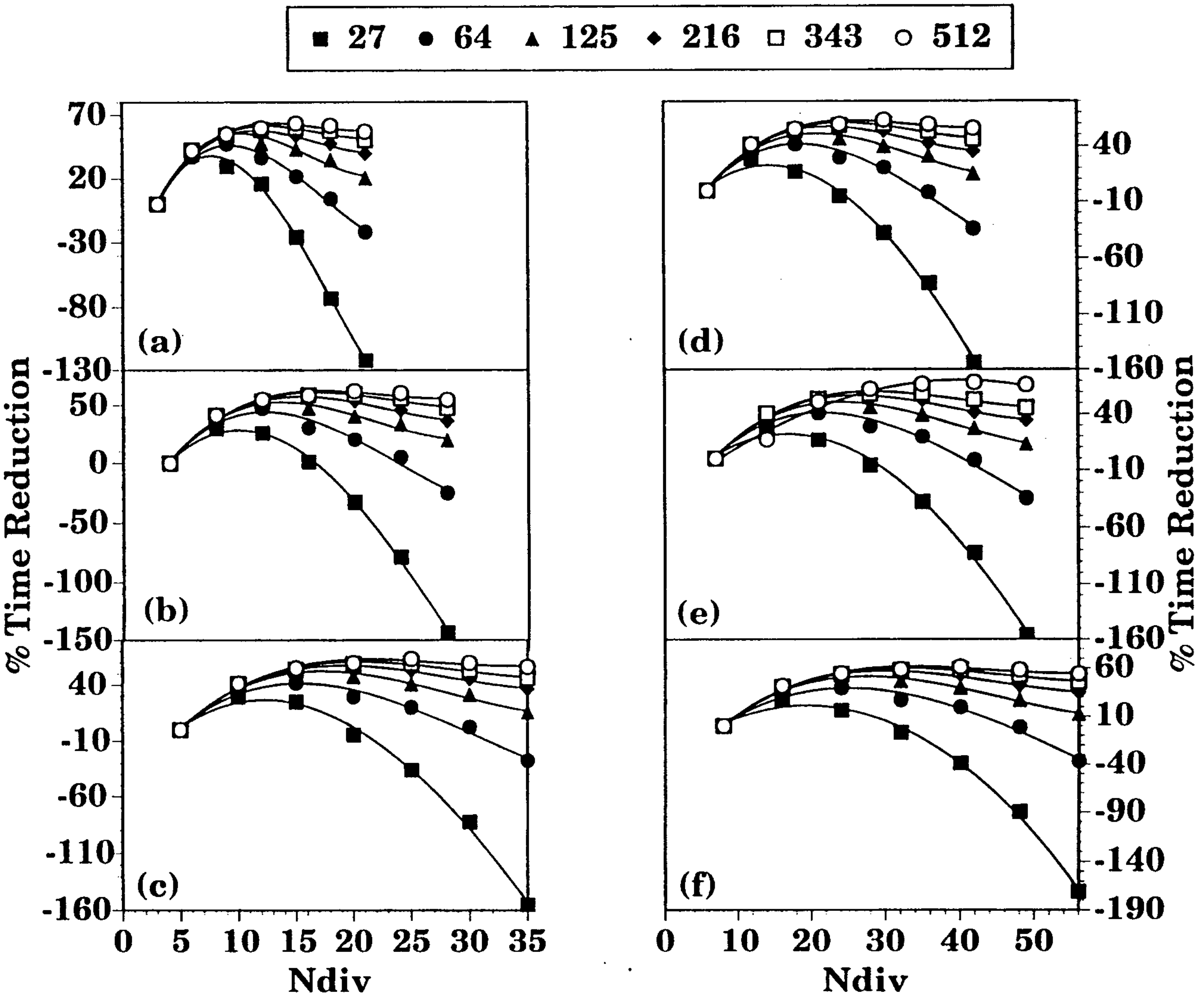

Figure 6. **Percent Time Reduction of the Modified Cell-Linked List Method Over the Conventional Method as a Function of the Number of Divisions (N_{div}) Along Each Edge of the Simulation Box.**

Table 2. Number of Unnecessary Internuclear Distance Calculations for Various System Sizes and Particle Number Densities

N_{div}	No. of Atoms per Cell					
	27	64	125	216	343	512
Simulation Box consisting of 3×3×3 cubic cells						
Req.[a]	26,432	174,460	684,208	2,257,896	5,646,287	12,940,352
0[b]	238,924	1,317,668	5,009,417	14,745,300	37,232,143	82,604,224
3[c]	238,924	1,317,668	5,009,417	14,745,300	37,232,143	82,604,224
6	110,516	688,676	2,479,917	7,580,688	18,662,645	42,348,736
9	75,223	411,940	1,449,075	4,994,196	12,701,118	27,410,252
12	49,109	316,868	1,388,325	3,800,304	9,516,071	21,179,584
15	45,128	252,972	968,867	2,834,820	7,208,556	15,228,564
18	40,100	203,688	764,841	2,169,240	5,614,910	13,097,784
21	37,120	176,216	683,984	1,848,988	4,804,972	11,715,904
Simulation Box consisting of 4×4×4 cubic cells						
Req.[a]	70,304	436,860	1,696,928	5,594,028	13,769,740	31,587,760
0[b]	1421,824	7,949,700	30,299,072	89,950,548	227,164,436	505,266,768
4[c]	558,688	3,100,036	11,799,072	34,709,844	87,868,020	194,888,272
8	293,332	1,609,092	6,111,572	17,727,060	45,043,784	99,467,856
12	170,464	948,712	3,858,284	11,596,116	29,810,540	63,704,884
16	132,992	825,732	3,224,804	8,838,660	21,984,672	49,283,664
20	115,276	603,780	2,315,872	6,565,260	16,714,388	35,911,728
24	93,200	463,576	1,770,764	5,227,860	13,790,044	30,060,984
28	82,340	427,036	1,533,444	4,626,168	12,218,292	27,882,672
Simulation Box consisting of 5×5×5 cubic cells						
Req.[a]	145,428	881,372	3,404,448	11,180,808	27,490,362	62,783,136
0[b]	5,548,197	31,114,628	118,658,052	353,305,692	891,621,013	1,985,184,864
5[c]	1,083,072	6,026,628	22,954,927	67,537,692	171,020,888	379,552,864
10	574,333	3,114,628	11,919,888	34,368,192	87,661,399	193,184,864
15	324,597	2,122,036	7,765,753	22,393,692	57,732,571	125,439,420
20	267,238	1,584,228	6,192,889	17,090,928	42,125,062	95,162,464
25	222,038	1,121,124	4,431,177	12,592,412	32,066,215	69,197,976
30	175,870	922,576	3,583,698	9,951,192	29,202,191	63,713,352
35	168,950	857,812	3,317,824	8,815,208	24,289,163	53,251,408
Simulation Box consisting of 6×6×63 cubic cells						
Req.[a]	259,752	1,555,996	5,988,268	19,614,300	48,193,916	109,755,152
0[b]	16,743,444	93,988,580	358,498,232	1,068,753,540	2,696,284,912	6,005,484,784
6[c]	1,863,096	10,381,028	39,560,732	116,411,268	294,833,524	654,601,456
12	995,060	5,349,092	20,576,920	59,094,372	151,113,168	332,557,552
18	644,208	3,646,756	13,406,952	38,402,436	99,192,204	216,103,172
24	476,948	2,704,100	10,576,944	29,333,220	73,346,640	163,167,472
30	391,924	1,870,044	7,549,532	21,371,448	54,740,844	118,474,024
36	299,248	1,687,464	6,011,796	18,939,108	50,250,076	108,856,920
42	282,396	1,579,524	5,666,484	15,456,412	41,272,636	90,945,600

[a] The number of internuclear distances that are within the cutoff distance and are required to be calculated in an evaluation of equation (1) and its derivatives.

[b] Corresponds to the brute force evaluation of the N*(N−1)/2 internuclear distances in a system of N particles.

[c] Corresponds to the conventional method of cell-linked lists.

Table 2. Number of Unnecessary Internuclear Distance Calculations for Various System Sizes and Particle Number Densities (continued)

N_{div}	No. of Atoms per Cell					
	27	64	125	216	343	512
Simulation Box consisting of 7×7×7 cubic cells						
Req.[a]	427,310	2,508,732	9,629,888	31,381,752	77,230,911	175,732,480
0[b]	42,451,120	238,425,444	909,481,487	2,713,097,076	6,843,353,865	15,244,669,440
7[c]	2,943,694	16,446,820	62,700,237	184,621,812	467,483,959	1,038,037,504
14	1,577,899	8,456,292	32,650,934	93,604,704	239,630,154	526,643,712
21	1,008,145	5,767,564	21,274,439	60,746,676	156,949,686	342,409,612
28	753,414	4,255,620	16,652,561	46,450,464	116,162,470	257,650,176
35	612,874	3,008,764	11,865,187	33,597,004	90,128,832	196,734,024
42	460,190	2,671,232	10,396,442	29,836,104	78,872,346	171,449,640
49	435,242	2,468,668	9,364,454	24,132,408	64,829,056	143,171,928
Simulation Box consisting of 8x8x8 cubic cells						
Req.[a]	639,300	3,787,580	14,510,808	47,190,180	116,163,092	263,943,792
0[b]	94,905,276	533,066,948	2,033,457,192	6,068,049,756	15,304,238,828	34,095,663,504
8[c]	4,392,636	24,507,588	93,457,192	275,240,796	696,938,988	1,547,864,464
16	2,368,072	12,580,036	48,710,196	139,378,524	357,232,184	784,501,136
24	1,503,420	8,582,740	31,738,324	90,330,972	233,554,028	510,266,172
32	1,134,968	6,309,060	24,695,332	69,107,820	172,954,384	382,962,064
40	919,568	4,477,816	17,572,392	49,688,288	133,800,800	292,228,740
48	693,232	3,979,096	15,616,800	44,186,460	116,607,292	254,340,984
56	695,396	3,785,284	13,906,096	35,478,860	95,880,684	211,718,836

[a] The number of internuclear distances that are within the cutoff distance and are required to be calculated in an evaluation of equation (1) and its derivatives.

[b] Corresponds to the brute force evaluation of the N*(N−1)/2 internuclear distances in a system of N particles.

[c] Corresponds to the conventional method of cell-linked lists.

INTENTIONALLY LEFT BLANK.

5. References

Allen, M. P., and D. J. Tildesley. *Computer Simulation of Liquids.* New York: Oxford University Press, 1990.

Boris, J. "A Vectorized 'Near Neighbors' Algorithm of Order N Using a Monotonic Logical Grid." *Journal of Computational Physics,* vol. 66, p. 1, 1986.

Brugè, F. "Systolic Calculation of Pair Interactions Using the Cell Linked-Lists Method on Multi-Processor Systems." *Journal of Computational Physics,* vol. 104, p. 263, 1993.

Brugè, F. and S. L. Fornili. "Concurrent Molecular Dynamics Simulation of Spinodal Phase Transition on Transputer Arrays." *Computer Physics Communication,* vol. 60, p. 31, 1990a.

Brugè, F. and S. L. Fornili. "A Distributed Dynamic Load Balancer and Its Implementation on Multi-Transputer Systems for Molecular Dynamics Simulations." *Computer Physics Communication,* vol. 60, p. 39, 1990b.

Lambrakos, S. G., and J. P. Boris. "Geometric Properties of the Monotonic Lagrangian Grid Algorithm for Near Neighbor Calculations." *Journal of Computational Physics,* vol. 73, p. 183, 1987.

Morales, J. J., L. F. Rull, and S. Toxvaerd. "Efficiency Test of the Traditional MD and the Link-Cell Methods." *Computer Physics Communication,* vol. 56, p. 129, 1989.

Rice, B. M., W. Mattson, J. Grosh, and S. F. Trevino. "A Molecular Dynamics Study of Detonation: I. A Comparison with Hydrodynamic Predictions." *Physical Review* E, vol. 53, p. 611, 1996.

Thompson, D. L. *Modern Methods for Multidimensional Dynamics Computations in Chemistry.* New Jersey: World Scientific, 1998.

Verlet, L. "Computer 'Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennary-Jones Molecules." *Physical Review*, vol. 159, p. 98, 1967.

INTENTIONALLY LEFT BLANK.

Appendix:

Sample FORTRAN Code

INTENTIONALLY LEFT BLANK.

```
! Cutoffr is the cut off radius.  Maxdim is the maximum coordinates for the simulations,
! mindim is the minimum.  Ndiv is the number of divisions that the simulation is divided into.
! All of these are arrays of length 2.
cr2 = cutoffr * cutoffr
clen = (maxdim - mindim) / ndiv
len = int(cutoffr / clen) + 1
maxlen = len(2)

! Iterate from the cell immediately next to the test cell to the last cell in the x direction.
! Since the height above the test cell is always the same as the height above the cell
immediately
! next to it we don't calculate it here.  We start at 2 just for array index reasons.
do i = 2,len(2) + 1

 ! Calculate the height above the current cell.
 lengths(i) = floor(sqrt(cr2 - ((i - 2) * clen(2))**2)/clen(1) + 1)

enddo

! Taking advantage of the above mentioned symmetry
lengths(1) = lengths(2)
n = 0

! Time to replicate the cells for all quadrants and create the offset list.
! Loop over every cell along the x dimension.
do i = -len(2),len(2)

  ai = abs(i) + 1

  ! Loop over every cell along the y axis for column i
  do j = -lengths(ai),lengths(ai)

   ! Don't include cell (0,0)
   if(i .ne. 0 .or. j .ne. 0) then

    n = n + 1
    overlay(n,1) = i
    overlay(n,2) = j

   endif
  enddo
enddo
```

INTENTIONALLY LEFT BLANK.

NO. OF
COPIES ORGANIZATION

2 DEFENSE TECHNICAL
INFORMATION CENTER
DTIC DDA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 HQDA
DAMO FDQ
D SCHMIDT
400 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 OSD
OUSD(A&T)/ODDDR&E(R)
R J TREW
THE PENTAGON
WASHINGTON DC 20301-7100

1 DPTY CG FOR RDE HQ
US ARMY MATERIEL CMD
AMCRD
MG CALDWELL
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS AT AUSTIN
PO BOX 202797
AUSTIN TX 78720-2797

1 DARPA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 NAVAL SURFACE WARFARE CTR
CODE B07 J PENNELLA
17320 DAHLGREN RD
BLDG 1470 RM 1101
DAHLGREN VA 22448-5100

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MAJ M D PHILLIPS
THAYER HALL
WEST POINT NY 10996-1786

NO. OF
COPIES ORGANIZATION

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL DD
J J ROCCHIO
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AS (RECORDS MGMT)
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

4 DIR USARL
AMSRL CI LP (305)

NO. OF COPIES	ORGANIZATION
1	HQDA SARD TT MR J APPEL WASH DC 20310-0103
1	HQDA OASA RDA DR C H CHURCH PENTAGON ROOM 3E486 WASH DC 20310-0103
4	COMMANDER US ARMY RESEARCH OFC R GHIRARDELLI D MANN R SINGLETON R SHAW P O BOX 12211 RESEARCH TRIANGLE PARK NC 27709-2211
1	DIRECTOR ARMY RESEARCH OFFICE AMXRO RT IP LIB SRVCS P O BOX 12211 RESEARCH TRIANGLE PARK NC 27709-2211
2	COMMANDER US ARMY ARDEC AMSTA AR AEE B D S DOWNS PICATINNY ARSENAL NJ 07806-5000
2	COMMANDER US ARMY ARDEC AMSTA AR AEE J A LANNON PICATINNY ARSENAL NJ 07806-5000
1	COMMANDER US ARMY ARDEC AMSTA AR AEE BR L HARRIS PICATINNY ARSENAL NJ 07806-5000

NO. OF COPIES	ORGANIZATION
2	COMMANDER US ARMY MISSILE COMMAND AMSMI RD PR E A R MAYKUT AMSMI RD PR P R BETTS REDSTONE ARSENAL AL 35898
1	OFFICE OF NAVAL RESEARCH DEPARTMENT OF THE NAVY R S MILLER CODE 432 800 N QUINCY STREET ARLINGTON VA 22217
2	COMMANDER NSWC R BERNECKER R-13 G B WILMOT R-16 SILVER SPRING MD 20903-5000
5	COMMANDER NAVAL RSRCH LAB M C LIN J MCDONALD E ORAN J SHNUR R J DOYLE CODE 6110 WASHINGTON DC 20375
2	COMMANDER NAVAL WEAPONS CENTER T BOGGS CODE 388 T PARR CODE 3895 CHINA LAKE CA 93555-6001
1	SUPERINTENDENT NAVAL POSTGRDTE SCHL DEPT OF AERONAUTICS D W NETZER MONTEREY CA 93940
3	AL LSCF R CORLEY R GEISLER J LEVINE EDWARDS AFB CA 93523-5000
1	AFOSR J M TISHKOFF BOLLING AIR FORCE BASE WASHINGTON DC 20332

NO. OF
COPIES ORGANIZATION

1 OSD SDIO IST
L CAVENY
PENTAGON
WASHINGTON DC 20301-7100

1 COMMANDANT
USAFAS
ATSF TSM CN
FORT SILL OK 73503-5600

1 UNIV OF DAYTON RSRCH INST
D CAMPBELL
AL PAP
EDWARDS AFB CA 93523

1 NASA
LANGLEY RESEARCH CENTER
LANGLEY STATION
G B NORTHAM MS 168
HAMPTON VA 23365

4 NTNL BUREAU OF STNDRDS
J HASTIE
M JACOX
T KASHIWAGI
H SEMERJIAN
US DEPT OF COMMERCE
WASHINGTON DC 20234

2 DIRECTOR
LLNL
C WESTBROOK
W TAO MS L 282
P O BOX 808
LIVERMORE CA 94550

1 DIRECTOR
LOS ALAMOS NATIONAL LAB
B NICHOLS T7 MS-B284
P O BOX 1663
LOS ALAMOS NM 87545

2 PRINCETON COMBUSTION
RSRCH LABORATORIES INC
N A MESSINA
M SUMMERFIELD
PRINCETON CORPORATE PLAZA
BLDG IV SUITE 119
11 DEERPARK DRIVE
MONMOUTH JUNCTION NJ 08852

NO. OF
COPIES ORGANIZATION

3 DIRECTOR
SANDIA NATIONAL LABS
DIVISION 8354
S JOHNSTON
P MATTERN
D STEPHENSON
LIVERMORE CA 94550

1 BRIGHAM YOUNG UNIVERSITY
DEPT OF CHMCL ENGNRNG
M W BECKSTEAD
PROVO UT 84058

1 CALIFORNIA INST OF TECH
JET PROPULSION LAB
L STRAND MS 125 224
4800 OAK GROVE DRIVE
PASADENA CA 91109

1 CALIFORNIA INSTITUTE OF
TECHNOLOGY
F E C CULICK MC 301-46
204 KARMAN LAB
PASADENA CA 91125

1 UNIV OF CALIFORNIA
LOS ALAMOS SCNTFC LAB
P O BOX 1663
MAIL STOP B216
LOS ALAMOS NM 87545

1 UNIV OF CA BERKELEY
CHEMISTRY DEPARMENT
C BRADLEY MOORE
211 LEWIS HALL
BERKELEY CA 94720

1 UNIV OF CA SAN DIEGO
F A WILLIAMS
AMES B010
LA JOLLA CA 92093

2 UNIV OF CA SANTA BARBARA
QUANTUM INSTITUTE
K SCHOFIELD
M STEINBERG
SANTA BARBARA CA 93106

NO. OF
COPIES ORGANIZATION

1 UNIV OF CO AT BOULDER
ENGINEERING CENTER
J DAILY
CAMPUS BOX 427
BOULDER CO 80309-0427

3 UNIV OF SOUTHERN CA
DEPT OF CHEMISTRY
R BEAUDET
S BENSON
C WITTIG
LOS ANGELES CA 90007

1 CORNELL UNIVERSITY
DEPT OF CHEMISTRY
T A COOL
BAKER LABORATORY
ITHACA NY 14853

1 UNIV OF DELAWARE
T BRILL
CHEMISTRY DEPARTMENT
NEWARK DE 19711

1 UNIVERSITY OF FLORIDA
DEPT OF CHEMISTRY
J WINEFORDNER
GAINESVILLE FL 32611

3 GA INST OF TECHNOLOGY
SCHL OF AERSPCE ENGNRNG
E PRICE
W C STRAHLE
B T ZINN
ATLANTA GA 30332

1 UNIVERSITY OF ILLINOIS
DEPT OF MECH ENG
H KRIER
144MEB 1206 W GREEN ST
URBANA IL 61801

1 THE JOHNS HOPKINS UNIV
CPIA
T W CHRISTIAN
10630 LTLE PTXNT PKWY STE 202
COLUMBIA MD 21044-3200

NO. OF
COPIES ORGANIZATION

1 UNIVERSITY OF MICHIGAN
GAS DYNAMICS LAB
AEROSPACE ENGNRNG BLDG
G M FAETH
ANN ARBOR MI 48109-2140

1 UNIVERSITY OF MINNESOTA
DEPT OF MCHNCL ENGNRNG
E FLETCHER
MINNEAPOLIS MN 55455

4 PA STATE UNIVERSITY
DEPT OF MCHNCL ENGNRNG
K KUO
M MICCI
S THYNELL
V YANG
UNIVERSITY PARK PA 16802

2 PRINCETON UNIVERSITY
FORRESTAL CAMPUS LIB
K BREZINSKY
I GLASSMAN
P O BOX 710
PRINCETON NJ 08540

1 PURDUE UNIVERSITY
SCHOOL OF AERO & ASTRO
J R OSBORN
GRISSOM HALL
WEST LAFAYETTE IN 47906

1 PURDUE UNIVERSITY
DEPT OF CHEMISTRY
E GRANT
WEST LAFAYETTE IN 47906

2 PURDUE UNIVERSITY
SCHL OF MCHNCL ENGNRNG
N M LAURENDEAU
S N B MURTHY
TSPC CHAFFEE HALL
WEST LAFAYETTE IN 47906

1 RENSSELAER PLYTCHNC INST
DEPT OF CHMCL ENGNRNG
A FONTIJN
TROY NY 12181

NO. OF
COPIES ORGANIZATION

1 STANFORD UNIVERSITY
DEPT OF MCHNCL ENGNRNG
R HANSON
STANFORD CA 94305

1 UNIVERSITY OF TEXAS
DEPT OF CHEMISTRY
W GARDINER
AUSTIN TX 78712

1 VIRGINIA PLYTCHNC INST
AND STATE UNIVERSITY
A SCHETZ
BLACKSBURG VA 24061

1 APPLIED COMBUSTION
TECHNOLOGY INC
A M VARNEY
P O BOX 607885
ORLANDO FL 32860

2 APPLIED MCHNCS REVIEWS
THE AMERICAN SOCIETY OF
MECHANICAL ENGINEERS
R E WHITE
A B WENZEL
345 E 47TH STREET
NEW YORK NY 10017

1 BATTELLE
TWSTIAC
505 KING AVENUE
COLUMBUS OH 43201-2693

1 COHEN PRFSSNL SERVICES
N S COHEN
141 CHANNING STREET
REDLANDS CA 92373

1 EXXON RSRCH & ENGRNG CO
A DEAN
ROUTE 22E
ANNANDALE NJ 08801

1 GENERAL APPLIED SCIENCE
LABORATORIES INC
77 RAYNOR AVENUE
RONKONKAMA NY 11779-6649

NO. OF
COPIES ORGANIZATION

1 GENERAL MOTORS RSCH LABS
PHYSCL CHMSTRY DEPT
T SLOANE
WARREN MI 48090-9055

2 HERCULES INC
ALLEGHENY BALLISTICS LAB
W B WALKUP
E A YOUNT
P O BOX 210
ROCKET CENTER WV 26726

1 HERCULES INC
R V CARTWRIGHT
100 HOWARD BLVD
KENVIL NJ 07847

1 ALLIANT TECHSYSTEMS INC
MARINE SYSTEMS GROUP
D E BRODEN MS MN50-2000
600 2ND STREET NE
HOPKINS MN 55343

1 ALLIANT TECHSYSTEMS INC
R E TOMPKINS
MN 11 2720
600 SECOND ST NORTH
HOPKINS MN 55343

1 IBM CORPORATION
A C TAM
RESEARCH DIVISION
5600 COTTLE ROAD
SAN JOSE CA 95193

1 IIT RESEARCH INSTITUTE
R F REMALY
10 WEST 35TH STREET
CHICAGO IL 60616

1 LOCKHEED MSLS & SPACE CO
GEORGE LO
3251 HANOVER STREET
DEPT 52-35 B204 2
PALO ALTO CA 94304

1 OLIN ORDNANCE
V MCDONALD LIBRARY
P O BOX 222
ST MARKS FL 32355-0222

NO. OF
COPIES ORGANIZATION

1 PAUL GOUGH ASSOCIATES INC
P S GOUGH
1048 SOUTH STREET
PORTSMOUTH NH 03801-5423

1 SCIENCE APPLICATIONS INC
R B EDELMAN
23146 CUMORAH CREST
WOODLAND HILLS CA 91364

3 SRI INTERNATIONAL
G SMITH
D CROSLEY
D GOLDEN
333 RAVENSWOOD AVENUE
MENLO PARK CA 94025

1 STEVENS INST OF TECH
DAVIDSON LABORATORY
R MCALEVY III
HOBOKEN NJ 07030

1 NYMA INC
LERC GROUP
R J LOCKE MS SVR 2
2001 AEROSPACE PKWY
BROOK PARK OH 44142

3 THIOKOL CORPORATION
ELKTON DIVISION
R BIDDLE
R WILLER
TECH LIB
P O BOX 241
ELKTON MD 21921

3 THIOKOL CORPORATION
WASATCH DIVISION
S J BENNETT
P O BOX 524
BRIGHAM CITY UT 84302

1 UNITED TCHNLGS RSRCH CTR
A C ECKBRETH
EAST HARTFORD CT 06108

1 UNIVERSAL PRPLSN CO
H J MCSPADDEN
25401 NORTH CENTRAL AVE
PHOENIX AZ 85027-7837

NO. OF
COPIES ORGANIZATION

1 UNITED TECHNOLOGIES CORP
CHEMICAL SYSTEMS DIVISION
R R MILLER
P O BOX 49028
SAN JOSE CA 95161-9028

1 VERITAY TECHNOLOGY INC
E B FISHER
4845 MILLERSPORT HWY
P O BOX 305
EAST AMHERST NY 14051-0305

1 FREEDMAN ASSOCIATES
E FREEDMAN
2411 DIANA ROAD
BALTIMORE MD 21209-1525

1 ALLIANT TECHSYSTEMS INC
J BODE
600 SECOND ST NE
HOPKINS MN 55343

1 ALLIANT TECHSYSTEMS INC
C CANDLAND
600 SECOND ST NE
HOPKINS MN 55343

1 ALLIANT TECHSYSTEMS INC
L OSGOOD
600 SECOND ST NE
HOPKINS MN 55343

1 ALLIANT TECHSYSTEMS INC
R BECKER
600 SECOND ST NE
HOPKINS MN 55343

1 ALLIANT TECHSYSTEMS INC
M SWENSON
600 SECOND ST NE
HOPKINS MN 55343

1 BENET LABORATORIES
SAM SOPOK
AMSTA AR CCB B
WATERVLIET NY 12189

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
40	DIR, USARL AMSRL-WM-B, A W HORST AMSRL-WM-BD, B E FORCH G F ADAMS W R ANDERSON R A BEYER S W BUNTE C F CHABALOWSKI S COLEMAN A COHEN R CUMPTON R DANIEL D DEVYNCK R A FIFER B E HOMAN A JUHASZ A J KOTLAR R KRANZE E LANCASTER W F MCBRATNEY K L MCNESBY M MCQUAID N E MEAGHER M S MILLER A W MIZIOLEK J B MORRIS J E NEWBERRY S V PAI R A PESCE-RODRIGUEZ J RASIMAS P REEVES B M RICE P SAEGAR R C SAUSA M A SCHROEDER R SCHWEITZER L D SEGER J A VANDERHOFF D VENIZELOS A WHREN H L WILLIAMS

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
June 1999

3. REPORT TYPE AND DATES COVERED
Final, Jan 97 - May 98

4. TITLE AND SUBTITLE
Near-Neighbor Calculations Using a Modified Cell-Linked List Method

5. FUNDING NUMBERS
1L161102AH43

6. AUTHOR(S)
William Mattson and Betsy M. Rice

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WM-BD
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1956

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

We have modified the conventional cell-linked list method to reduce the number of unnecessary internuclear distance calculations in molecular simulations of systems containing many particles. In the conventional method, the simulation space is partitioned into cells with edge lengths no less than the cutoff distance of the interaction potential (r_{cut}). The atoms are assigned to cells according to their spatial positions, and all internuclear distances for atoms within a cell and atoms in the same and nearest neighbor cells are evaluated. While this method ensures that the internuclear separation between all atom pairs within r_{cut} is calculated, it allows for unnecessary internuclear distance calculations between pairs that are within the volume encompassing the neighbor cells but that are separated by more than r_{cut}. The modified method presented here allows for reductions in the cell sizes and the number of atoms within the volume encompassing the neighbor cells. These reductions decrease the number of atoms that are outside of the interaction range and the number of unnecessary internuclear distance calculations while ensuring that all internuclear distances within the cutoff range are evaluated. We present algorithms to determine the volume with the minimum number of neighbor cells as a function of cell size and the identities of the neighboring cells. We also evaluate the serial performance using the modified form as functions of cell size and particle density for comparison with the performance using the conventional cell-linked list method.

14. SUBJECT TERMS
internuclear distances, linked lists, molecular dynamics, Monte Carlo

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1956 (Mattson [Rice]) Date of Report June 1999

2. Date Report Received ______

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ______

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ______

CURRENT ADDRESS

Organization

Name E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)