JOURNAL OF COMPUTATIONAL PHYSICS 73, 183–202 (1987)

Geometric Properties of the Monotonic Lagrangian Grid Algorithm for Near Neighbor Calculations

S. G. LAMBRAKOS AND J. P. BORIS

Laboratory for Computational Physics, Naval Research Laboratory, Code 4040, 4555 Overlook Ave., S. W., Washington, D.C. 20375-5000

Received January 15, 1986; revised August 6, 1986

Because spatial coordinates define a natural ordering of positions, it is always possible to associate with a set of randomly located points in a 3D space, grid indices which are ordered according to their relative positions. Such an indexing scheme can be used to construct a "monotonic Lagrangian grid" (MLG), a data structure where adjacent objects in space have close grid indices. Using an MLG to index positions and attributions of objects in computer memory permits a near neighbor algorithm to be based on a "maximum index offset," N_c, rather than a short range "cutoff" distance R_c. An MLG algorithm removes the necessity of having to test distances or directions to determine adjacency. Further, "close" objects can be indexed via contiguous memory, thus permitting efficient vectorization of computations.

1. BACKGROUND

This paper analyzes an efficient algorithm for keeping track of "near neighbor" relationships among a large number of nodes, i.e., locations, objects, or particles, in a region of 3D space. The need to treat "near neighbor" interactions applies to any system where:

(1) The node positions change due to particle velocity, local fluid velocity or changing view point. The neighborhood of each node is subject to continual change as some nodes move closer and others away.

(2) Nearby node pairs interact. The interaction could be an interparticle force or the rate of exchange of some quantity. Other relationships include geometric obscuration or graphical hidden line removal.

(3) One can define a "cutoff" separation or radius R_c according to the type of interaction considered. For internode separations greater than R_c the interactions may be neglected, computed through some other approximation, or included through interactions with nearer nodes.

For a large system of N nodes, it is advantageous to compute the interactions of each node with only a relatively few near neighbors. Algorithms are desired in which most of the nodes can be ignored without having to compute distances.

0021-9991/87 $3.00

Pairwise interactions are only computed when the relative separation of the two nodes is less than the "cutoff radius."

It follows, for each node in a system of N nodes, that one must make $N-1$ "cutoff" tests when no special algorithm is available to identify the near neighbors. Consequently, the operation count for distance checking scales as N^2. Even when an interaction is neglected because $|R_a - R_b| > C_c$, checking the separation distance requires about 10 floating point operators per pair, a substantial fraction of the work needed to calculate the entire interaction.

The operation count to identify near neighbors can be reduced significantly when node coordinates are ordered such that cutoff separation tests need only be performed over a small subset of the number of nodes in the system. Scalar sorting procedures have been developed for this purpose with operation counts scaling linearly with N or $N \times \log_2(N)$ [1, 5]. Because of the relatively slow scalar operations required in these algorithms to keep track of near neighbors, however, the computational cost is still prohibitive for large 3D systems using vector or parallel-processing supercomputers. The communications and data structures for these scalar algorithms are also not optimum for the fastest computers available.

Algorithms using neighbor list techniques [7] which are vectorizable have been developed. These algorithms, however, have larger storage requirements and are thus not ideal for large systems.

Boris [2] and Boris and Lambrakos [3] have developed an algorithm for keeping track of near neighbor interactions and geometric relationsips which scales as N and is structured to permit optimized vector and parallel processor implementations. This development followed from efforts on the "nearest neighbors" problem begun with K. V. Roberts [8] at Culham Laboratory in the context of gravitationally attracting stars. There we chose a field-solver approach to finding the forcces not only because of the long range nature of the gravitational force but also because a good near neighbors algorithm was lacking.

Our new algorithm uses a monotonic Lagrangian grid (MLG) for indexing the geometric positions and other dynamical attributes of the moving nodes in computer memory. The indexing ensures that nodes which are adjacent in real space are given MLG indices which are also very close. Given N nodes randomly located in a region of 3D space, one can associate with each node not only its spatial coordinates (X, Y, Z) but also a set of MLG indices (i, j, k). A useful mapping, which we have named a monotonic Lagrangian grid, is obtained when the node locations in space and the node indices in the computer memory satisfy a set of monotonicity conditions, for example,

$$\begin{aligned} &X(i,j,k) < X(i+1,j,k) \qquad \text{for } 1 < i < N_x - 1 \text{ and all } j, k \\ &Y(i,j,k) < Y(i,j+1,k) \qquad \text{for } 1 < j < N_y - 1 \text{ and all } i, k \end{aligned}$$

and (1.1)

$$Z(i,j,k) < Z(i,j,k+1) \qquad \text{for } 1 < k < N_z - 1 \text{ and all } i, j.$$

Here N, the total number of nodes, equals $N_x \times N_y \times N_z$. Note that the average separation of neighbors is not independent of the direction of their MLG index displacement since the metrics of the X, Y, and Z coordinates in the spatial domain need not be equal (nor even orthogonal). The MLG conditions defined in (1.1), although mathematically satisfactory for organizing random locations in space, are not optimal for mapping node positions into a vector computer memory. A skew periodic MLG with somewhat different indexing is described in Section 4.

When two adjacent nodes pass each other in real space, relative to one of the chosen coordinate directions, their indices are exchanged or "swapped" in the MLG by moving the data for each node from its original indexed location to the indexed location of the other node which it just passed in space. The data for the nodes are swapped in the computer memory cells. This local swapping maintains a monotone mapping between the instantaneous positions of the nodes in real space and their MLG indices. The ordinal node locations within the compact, regular MLG arrays are the same as the ordinal node locations in space. Node positions or any node attributes indexed in computer memory according to this scheme are said to be in "MLG order."

The speed of the MLG algorithm is controlled by (1) its straightforward vectorization; (2) the rapid convergence of the swapping procedure to restore the MLG; (3) the average distance the nodes travel between MLG reorderings; (4) the number of pairwise interactions for each node; and (5) the computational cost of computing these interactions. In our test and applications [9, 10] to date, the cost of swapping iterations scales as $C_1 \times N \times \log_2(N)$ while the cost of calculating pair interactions scales as $C_2 \times N$. So much work is done per node to calculate the pair interactions, however, that $C_1 \times \log_2(N)$ is of order $0.04 \times C_2$ for $N = 512$. When $N = 262{,}144$, $\log_2(N)$ is a factor of two larger and the cost of restructing the MLG increases to 8% of the interaction calculations.

This paper presents an analysis and statistical results of the MLG algorithm applied to the random motion of point nodes in a cubical region. This computational domain is periodic in X and Y and is bounded in Z by two reflecting walls. The nodes are noninteracting and have a random distribution of initial velocities. The two major aspects of the MLG algorithm considered here are the convergence of the swapping algorithm to maintain MLG ordering and the spatial properties of the new near neighbor indexing we have used, the "skew-periodic" MLG designed to facilitate long vector operations. This paper examines an MLG comprised of N_z identical k-planes. Node locations within each k-plane are indexed via a "skew-periodic" two-dimensional grid. The skew periodic indexing scheme is described and analyzed in Section 4.

Swapping to maintain the monotone mapping is an iterative process. Its convergence rate depends on the size of the grid and how far the nodes move between restoration of the MLG ordering by swapping. Analysis of the required number of swapping iterations shows a better overall convergence rate for "large" changes in the positions of the nodes, i.e., long timesteps which result in significant MLG distortion. This is extremely encouraging as it implies that the MLG

swapping procedure is also a reasonable way to order fully disordered nodes. This paper presents frequency distributions of swapping iterations and swapping convergence characteristics for different timestep sizes and system sizes.

The MLG algorithm is adaptable to a wide range of applications including important problems in astrophysics, molecular dynamics, particle, vortex, and object tracking data bases, and fluid dynamics which require calculation of near neighbor interactions for a large number of nodes whose relative positions change continually.

The authors point out that the monotonic Lagrangian grid described in this paper *is* the monotonic logical grid described in related work given in the references. This name change has evolved from two distinct aspects of MLG algorithms which contribute to their efficiency. These are: the method of dynamically updating, in a Lagrangian manner, neighbor information and that the MLG inherently establishes an ordering relational or "logic" among information concerning nodes as it is arranged in memory thus providing a compact data structure for efficiently computing interactions among nodes.

2. Near Neighbors Template for the Monotonic Lagrangian Grid

The "order N" scaling of the MLG algorithm's interaction calculation is effected by calculating node–node interactions for only a finite set of small index offsets in the MLG which correspond to the near neighbors in space. The size and configuration of this set of index offsets, termed the near neighbors template (NNT), influences the coefficient of the MLG cost which scales with N.

The NNT can be visualized as a cluster of nodes, the near neighbors, surrounding a "focal node" (FN). If a particular node is taken as the focal node, the remaining nodes of the "template" define a local "pattern" in the MLG corresponding to the relative index offsets of the near neighbors of the focal node. Typical NNTs with different upper bounds for the logical displacement of near neighbors (i.e., shells) are shown in Fig. 1. Only index offsets larger than zero need be considered. Interactions with nodes having a negative address offset will be calculated when those nodes are focal nodes. Three shells of interaction are defined in Fig. 1 corresponding to neighbors at different probable separations. The 16 neighboring nodes indicated with squares form the closest shell. The 30 triangle nodes are on average further away and the 16 circle nodes are yet further away. The full $5 \times 5 \times 5$ cubical template shown in Fig. 1 nominally includes 125 nodes, but since the focal node does not interact with itself and each interaction does not need to be counted twice, there are $(125-1)/2 = 62$ interactions considered for each node. To complete the shell of circle points requires considering nodes further from the focal node than two layers in each direction.

In general, there should be a correlation between the size of an NNT, in terms of the number of nodes included in the logical near neighborhood and the average distance between nodes in the system. However, the NNT size and configuration is

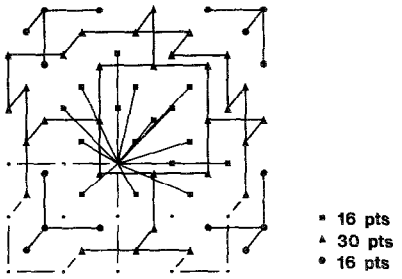

FIG. 1. Near neighbors template defining the logical displacement of nodes considered to be near neighbors.

controlled by the probability, as a function of MLG index, of a neighboring node having a separation less than R_c. The NNT should be taken large enough that the likelihood of a near miss, that is, a node which is outside the template but yet within R_c of the focal node, is acceptably small.

The characteristics of a near neighbors template will depend on the particular monotone mapping connecting the spatial locations of nodes and the corresponding locations in the computer memory. When the average node separation in Z, for example, is half of the average separation in X or Y, the NNT probably should reach more MLG index layers in the k direction (along Z). In fact, there usually exists more than one MLG defining a monotone space-to-index mapping. This non-unique property of MLGs provides latitude for further optimization in particular problems. For example, an optimum MLG for one problem may minimize distances to near neighbors. In another problem, the MLG may be optimized when the shortest distance to non-near neighbors is maximized.

The statistical analysis of near neighbor locations which follows is discussed in terms of the NNT described below. This analysis considers the following questions:

(1) What is the correspondence between relative index offsets of nodes in the MLG and the corresponding relative spatial positions? What is the average separation in space of two nodes which are adjacent in the MLG? How does this average separation depend on the index offsets between the two MLG nodes?

(2) How does the necessary configuration and size of an NNT depend on the specific type of MLG used for indexing node positions, i.e., the specific indexing scheme? In particular, what characteristics of a skew-periodic MLG might require or benefit from a modification of the NNT configuration?

(3) For a given MLG, what methods are available for optimizing the configuration of an NNT and how does node motion affect this optimization? For example, for nodes moving randomly in 3D space the probability distribution for the relative separation of near neighbors is spherically symmetric. This symmetry can be used to reduced NNT size without inhibiting the accuracy of an algorithm using the MLG.

The appropriate NNT of course depends on the specific MLG indexing scheme selected. The MLG considered in this analysis is a skew-periodic grid as described in Section 4. It consists of 8 logical planes each consisting of 64 logical cells arranged in an 8×8 array. For this MLG the number of node–node interactions computed each timestep depends on the number of inter- and intraplane interactions indexed by the NNT for each "focal node." An example of the computational cost coefficient multiplying the order N scaling of the MLG algorithm is given in Ref. [5].

3. Statistical Analysis of Near Neighbor Positions for Points in 3D Random Motion

Interpreting statistical information concerning spatial relationships and correlations between positions in the MLG requires specifying the parameters which affect these positions each timestep. For any system of nodes some of the major parameters are:

(1) The size of the spatial domain relative to the number of nodes comprising the system, i.e., the node density.

(2) The nature of the motion of the node system, e.g., random or nonrandom, rotational, compressional, anisotropic.

(3) The logical structure of the MLG used to index the node positions.

The statistical analysis described in this section considers a system of 512 noninteracting points which can pass arbitrarily close to each other. The velocities of these points are random and uniformly distributed in each coordinate from -1×10^7 cm/s to 1×10^7 cm/s. The spatial domain, $80\ \mathring{A} \times 80\ \mathring{A} \times 80\ \mathring{A}$, corresponds to an average separation of adjacent nodes of approximately $10\ \mathring{A}$ in each coordinate, roughly the density of gas near standard temperature and pressure.

Useful statistical information about near neighbor positions is obtained by analyzing the statistical distribution of neighbor–focal node separations for the different NNT offsets. These distributions are accumulated over a sufficiently large number of timesteps so that the fluctuations are small. For the system considered here, we define a frequency distribution function $f(R; i, j, k)$, where

$$f(R; i, j, k) = \text{Frequency for a near neighbor node (with NNT offsets } i, j, k) \text{ having a serapation from the focal node contained in the shell extending from radius } R \text{ to } R + DR. \tag{3.1}$$

The distance classification interval DR is adjusted according to the number of separations sampled.

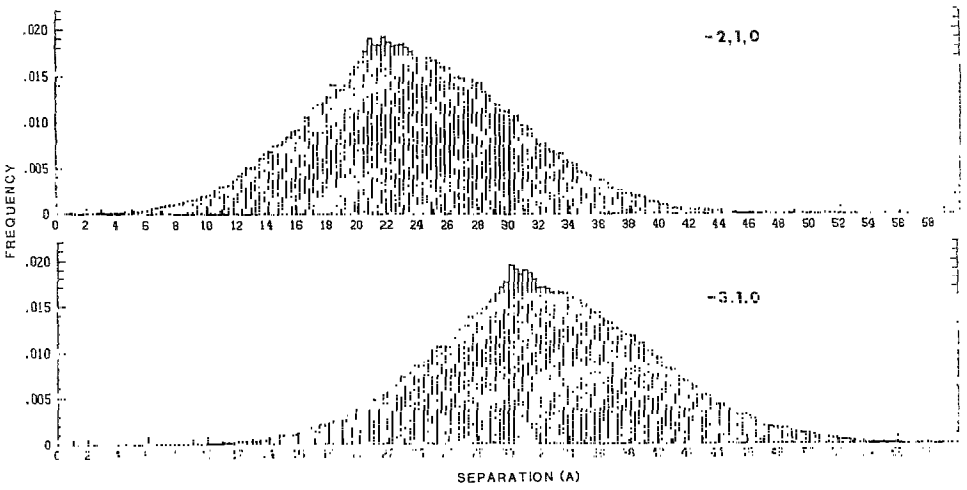

FIG. 2. Probability distribution of neighbor separation corresponding to NNT cell index offsets $(-2, 1, 0)$ and $(-3, 1, 0)$. The bin size for these distributions is 0.3 Å.

Shown in Fig. 2 are probability distributions for near neighbor separations corresponding to two NNT offsets. In computing these distributions all 512 nodes were treated as focal nodes. The distance classification interval DR for each of these distributions was 0.3 Å. The time sampling interval used in computing $f(R; i, j, k)$ for each of the NNT offsets consisted of five hundred timesteps of length 2.5×10^{-16} s. This time interval is sufficiently long that the high velocity nodes easily traverse the spatial domain, i.e., 80 Å, several times. Figure 2 shows a correlation between the mean value of $f(R; i, j, k)$ and the NNT offsets. The peak of the distribution moves approximately 10 Å toward larger separations (from 21 Å to 31 Å) when one more node is added between the focal node and the interaction node.

Frequencies for neighbor separations at short range corresponding to an NNT cell having a relatively large index offset from the focal node, i.e., the probability distribution in Fig. 2 corresponding to NNT offset $(-3, 1, 0)$, are shown in Fig. 3. As can be seen, even for relatively large index offsets from the focal node there is a small but finite probability for a not-so-near neighbor coming quite "close." It is these rare "near miss" events which determine the NNT size required to reduce the near miss probability to a statistically insignificant level.

Additional information concerning relative positions of neighbors is provided by cumulative integrals over $f(R; i, j, k)$ from 0 to a given separation from the focal node. These integrals give the probability that target nodes with a given offset from their focal node come within a particular distance of the focal node in space, a statistical "near miss" probability. Such information provides a criterion for optimizing (or minimizing) the NNT based on the "cutoff radius" R_c for the particular system. NNT optimization requires analysis of the near miss frequency for each NNT offset, i.e., the cumulative probability of finding a node indexed outside

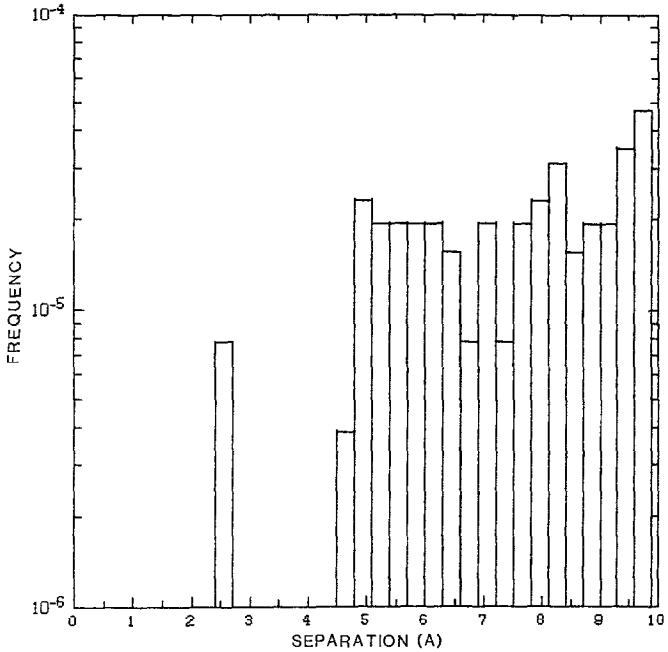

FIG. 3. Frequencies for neighbor separations at short range corresponding to relatively large index offsets, i.e. $(-3, 1, 0)$, from the target cell. These frequencies are the same as those shown in Fig. 2 for the range of distance 0 to 10 $\AA$.

the NNT but within a distance R_c of the focal node. Shown in Table I are the near miss probabilities as a function of NNT offset for the particular case $R_c = 3\ \AA$. These probabilities were computed with the same sampling used for the frequency functions shown in Fig. 2 and 3. NNT optimization based on near miss probability is discussed in Section 5. Note that no repulsion exists at short distances in these tests so this problem overestimates the near miss probability.

For the system of nodes represented in Table I, i.e., a system where the cutoff radius $R_c = 3\ \AA$, all NNT cells more than three index offsets from the focal node in

TABLE I

Near Miss % Probability of a Target Node Coming within $R_c = 3\ \AA$ of the Focal Node as a Function of NNT Offset ($(8 \times 8 \times 8)$ System)

$j+3$	0.000	0.000	0.0023	0.000	0.000	0.000	0.000
$j+2$	0.000	0.0016	0.0222	0.0175	0.0218	0.000	0.000
$j+1$	0.0008*	0.0195*	0.3138	0.9886	0.2737	0.0066	0.000
j				F.N.	1.061	0.0125	0.000
				i	$i+1$	$i+2$	$i+3$

* Frequency distributions corresponding to these cells are shown in Fig. 2. The focal node is at grid location (i, j, k).

any direction have zero probability of recording a "near miss," i.e., coming within $3\,\mathring{A}$ of the focal cell. For this system a suitable upper bound on the logical separation included in the near neighbors template is therefore $N_c = 3$. For this case [5] the total number of node–node interactions computed each timestep for an MLG of size N is

$$\text{computed interactions} = 171 \times N. \tag{3.2}$$

The NNT corresponding to (3.2) was selected on the basis of an upper bound, N_c, on the MLG location of "nearest neighbors." However, as seen in Table I, further deletion of NNT cells is possible using an NNT which is more nearly spherical. This aspect of NNT optimization is examined in the next section for the skew-periodic MLG.

4. A "Skew Periodic" MLG for Indexing the Geometric Positions of Nodes in Computer Memory

More than one type of MLG can be constructed to index nodes in a given spatial domain. In particular, for nodes in a 3D spatial domain with periodic boundary conditions, a new monotonic Lagrangian grid can be constructed which permits more efficient vector manipulation of node attributes than the regular periodic data structure. This "skew periodic" MLG also partitions the computational domain into cells of equal statistical volume. The data structure removes the need for guard cells at the ends of the rows of nodes to implement periodic boundary conditions.

"Skewing" is a statistical property of a system of nodes where the average relative location of neighbors is a function of the direction of the MLG coordinate offset relative to the focal node. These skewed positions follow from the asymmetric constraints of indexing node positions in a skew periodic lattice. In a skew periodic MLG the mapping between the spatial coordinates of the neighbors and their MLG indices is not exactly aligned. The nature of this misalignment is a result of the monotonicity conditions imposed by Eq. (4.1) given below. Position skewing depends on the MLG configuration and diminishes with increased size of the node system.

The skew periodic MLG is comprised of a set of N_z logical planes which are each skew-periodic. Thus, for all nodes in the system there will be one space coordinate, say Z, and one MLG plane index, say k, which must satisfy the monotonicity condition $Z(i, j, k) < Z(i, j, k+1)$ for $1 < k < N_z - 1$. Since $N = N_x \times N_y \times N_z$, each logical plane of the MLG will index $N_x \times N_y$ nodes randomly located in 2D space. Compact vectorization of each plane is achieved by indexing in monotonic order

positions. This indexing scheme provides a mapping of a 2D data plane onto a

single continuous MLG coordinate axis. Such a mapping satisfies slightly different monotonicity conditions from (1.1).

$$X(ij,k) < X(ij+1,k) \quad \text{for} \quad 1 < ij < N_x \times N_y - 1 \text{ and all } k$$
$$Y(ij,k) < Y(ij+N_x,k) \quad \text{for} \quad 1 < ij < N_x \times N_y - N_x \text{ and all } k$$
and (4.1)
$$Z(ij,k) < Z(ij,k+1) \quad \text{for} \quad 1 < k < N_z - 1 \text{ and all } ij.$$

To emphasize vector indexing the combined index ij is used instead of i and j. The point is that the entire plane of $N_x \times N_y$ nodes are now meaningfully contiguous. A statistical analysis of position skewing resulting from this MLG follows.

Consider $N_x \times N_y$ nodes located in a doubly periodic 2D region of space having an area $L_x \times L_y$. Letting N_x and N_y equal 3 here for presentation purposes the $N_x \times N_y$ regularly spaced nodes and their periodic images are shown in Fig. 4a.

In Fig. 4a the MLG locations, i.e., circles, which are numbered represent the location of the nodes which form the regular periodic MLG. The circles which are

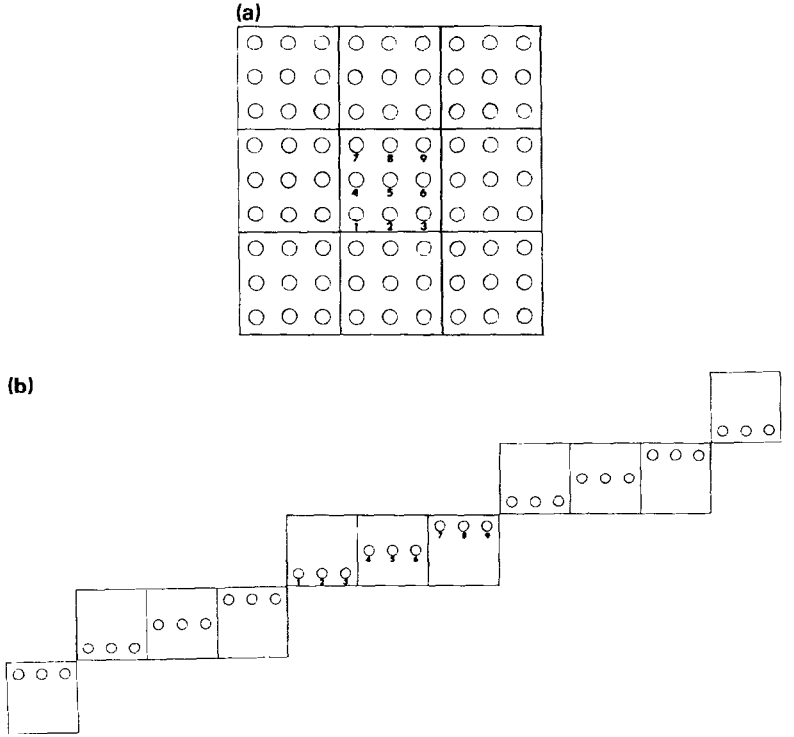

FIG. 4. Schematic representation of ghost cells required for an NNT having maximum index offsets of three: (a) indexing via regular periodic MLG; (b) indexing via a skew periodic MLG. The first and last three ghost nodes shown for the skew periodic MLG are required for an NNT having an upper bound of 3 on the logical separation.

not numbered represent the locations of some of the node images, each of which is a distance $L=((mL_x)^2+(nL_y)^2)^{1/2}$ $(m, n=0, \pm 1, \ldots)$ from their corresponding system nodes. Note that extra images, also called "ghost" nodes, must be added to all four boundaries to represent the images in locations where vector operations expect to find them. The ghost nodes in the X direction (horizontally in Fig. 4a) interrupt the adjacency of data in computer storage and hence interrupt numerical optimization through vectorization. The skew-periodic MLG is formed by the nodes and node images whose spatial locations satisfy the modified constaints (4.1). This is depicted in Fig. 4b. Note that the first and last three ghost nodes shown in Fig. 4b are required for an NNT having an upper bound of 3 on the logical separation, i.e., $N_c=3$.

The system of 512 random points was used to investigate the relative spatial positions of nodes and node images indexed by a skew periodic MLG. The positions in X and Y of the first N_x points and the monotonically ordered positions of the images of the remaining $(N_x)\times(N_y-1)$ points were fit to a straight line using least squares. The configuration sketched in Fig. 4b suggests that the nodes will lie, on average, along a skewed line which moves up one cell in Y each time the system is traversed in the X direction. This is shown for one of the 8×8 skew periodic planes from our 512-particle test problem in Fig. 5a.

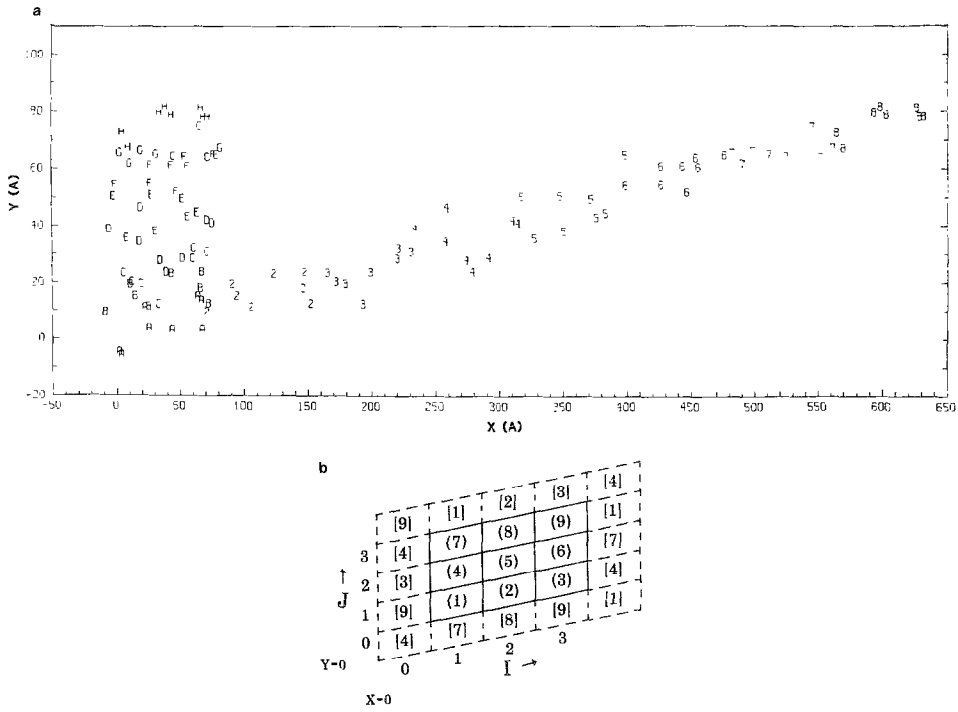

FIG. 5. (a) Comparison of node positions indexed by regular and skew periodic MLGs. (b) Average area partitioning via parallelograms. Square bracket numbers are images.

The calculated slopes of this least squares line fluctuated between 0.12 and 0.13. This is consistent with an 8×8 logical plane since $1/8 = DY/(N_x \times DX) = 0.125$. The correlation coefficient for each least-squares fit was approximately 0.95. A typical plot of positions indexed by the skew periodic MLG for a logical plane is shown in Fig. 5a. Also shown in this figure are the corresponding node positions as would be indexed by a regular periodic MLG. That is, the nodes labeled "B", whose positions are indexed by the regular periodic MLG, correspond to the nodes labeled "2", whose positions are indexed by the skew periodic MLG. This same correspondence holds between the nodes labeled "C" and "3," "D," and "4," etc.

The skewing associated with a 2D skew periodic MLG is illustrated by Fig. 5b. The average Y coordinates of neighbors in the direction of decreasing X is smaller than the average Y coordinate of neighbors in the opposite direction. This results directly from the indexing which moves up one row of nodes for each time the system length is traversed in the X direction.

The average relative spatial position of logical neighbors is a function of the MLG indexing scheme. We would like to know on average where, relative to a focal node, neighbors with a particular MLG offset are likely to be. For a regular periodic MLG the overall volume is partitioned essentially cubically. For a skew-periodic MLG the statistical volume elements have the shape of parallelepipeds. For a 2D skew-periodic MLG, i.e., a logical plane, this is equivalent to an area partitioning in parallellograms. That is, the average relative locations of four adjacently located random nodes (i, j), $(i+1, j)$, $(i, j+1)$, and $(i+1, j+1)$ form a parallelogram.

The average distance of neighboring nodes from the focal node is recorded in Fig. 4a for the system of 512 points with random positions. Note that each NNT node has an average RMS distance to the focal node which is only about a factor of $\sqrt{2}$ longer than the corresponding distance in a perfectly regular Eulerian grid. This is the local price paid for the desired global ordering of the MLG. Shown in Tables IIb and c are the average separations of neighbors in X and Y, respectively. The average X separation is exactly what we would expect, 10 $\mathring{A}$. It is seen that only the average Y separations are affected by skewing.

Indexing the NNT locations using coordinates (i, j) and taking the focal node (FN) as the origin, skewing is examined by taking divided differences between the Y components of neighbors. From Table IIc, for a $8 \times 8 \times 8$ system of nodes having random velocities (top number in each cell),

$$\frac{Y(3, 3) - Y(-3, 3)}{60} = \frac{33.75 - 26.25}{60}$$

$$= 0.125 = \frac{N_y}{N_x \times N_y}, \tag{4.2}$$

Here the average node separation in each direction is 10 $\mathring{A}$. Because the average skewing scales as $1/N_x$, it diminishes as the system size increases. This is a relatively small price to pay for more efficient data storage and vectorization. From Table IIc,

TABLE II

Average Separation of Neighbors from Focal Node as a Function of NNT Offset[a]

$j+3$	41.20	35.54	32.13	31.70	34.37	39.52	46.32
	42.65	36.77	32.95	31.96	34.06	28.74	45.20
$j+2$	35.69	28.53	23.59	22.44	25.60	31.80	39.63
	36.80	29.56	24.38	22.73	25.36	31.19	38.77
$j+1$	32.37	23.73	16.78	14.02	18.12	25.68	34.57
	33.02	24.42	17.43	14.37	18.09	25.40	34.12
j				F.N.	14.06	22.59	31.96
					14.42	22.79	32.04
				i	$i+1$	$i+2$	$i+3$
Average separation in X							
$j+3$	−30.00	−20.00	−10.00	0.0004	10.00	20.00	30.00
	−30.00	−20.00	−10.00	0.0000	10.00	20.00	30.00
$j+2$	−30.00	−20.00	−10.00	0.0000	10.00	20.00	30.00
	−30.00	−20.00	−10.00	0.0000	10.00	20.00	30.00
$j+1$	−30.00	−20.00	−10.00	0.0004	10.00	20.00	30.00
	−30.00	−20.00	−10.00	0.0000	10.00	20.00	30.00
j				F.N.	10.00	20.00	30.00
					10.00	20.00	30.00
				i	$i+1$	$i+2$	$i+3$
Average separation in Y							
$j+3$	26.25	27.50	28.75	30.00	31.25	32.50	33.75
	28.13	28.75	29.38	30.00	30.63	31.25	31.88
$j+2$	16.25	17.50	18.75	20.00	21.25	22.50	23.75
	18.36	18.91	19.46	20.00	20.67	21.31	21.95
$j+1$	6.250	7.500	8.750	10.00	11.25	12.50	13.75
	10.34	10.56	10.69	10.00	11.59	12.35	12.98
j				F.N.	1.250	2.500	3.750
					0.6250	1.250	1.875
				i	$i+1$	$i+2$	$i+3$

[a] The top and bottom numbers in each grid cell correspond to $(8\times8\times8)$ and $(16\times16\times16)$ systems, respectively.

for a $16\times16\times16$ system of nodes (bottom number in each cell shown) having random velocities,

$$\frac{Y(3,3)-Y(-3,3)}{60}=\frac{31.88-28.13}{60}$$

$$=0.0625=\frac{1}{16}, \tag{4.3}$$

as expected.

5. Discussion

The non-orthogonal, skew-periodic indexing scheme works well in the MLG case because the local grid in the various index directions cannot be orthogonal in any case since the nodes are Lagrangian. Thus, there is no advantage to maintaining orthogonality in node indexing when actual spatial orthogonality is not possible.

In large systems there is the possibility that the X coordinates may lose some accuracy because many system lengths have to be added to the X coordinate when using the technique with one big 3D skew periodic grid. If skew periodicity were being used in a $100 \times 100 \times 100$ system, the image of point (0, 0, 0) would be 10,000 system lengths away, or 10^6 typical grid spacings. Clearly 32-bit precision would not be adequate. In such a large system, however, one would probably not want to use this technique. When the vectors become long enough, vectorizing in two dimensions at once is no longer attractive. On a Cray, for example, vectors of length 64 are long enough.

On arrays of parallel processors, periodic indexing is sometimes implemented in the hardware and skew periodic connectivity is often extended to a number of dimensions in the hypercube architecture. In these cases the difficulties and costs [redacted] processors would be absent so using the regular periodic grid might be simpler. On the other hand, the skew periodic representation may also be simpler and more efficient.

Although the physical system being described in the two representations, i.e., Figs. 4a and b, are identical, the data being stored to describe each system is quite different because different images of the nodes are active in each case. The X and Y coordinates of nodes 1, 2, and 3 are the same in both representations. In the regular periodic grid the nodes 4, 5, and 6, and for that matter 7, 8, and 9, all lie in the same periodic domain as the nodes 1, 2, and 3. In the skew periodic grid, each successively higher row of nodes (in this case $N_y = 3$ rows) is displaced a full system width L_x in the X direction. Instead of following node 3 with the image of node 1, as in the regular periodic grid, node 3 is followed directly by node 4. Similarly, node 6 is followed in the skew periodic MLG by node 7, not the image of node 4 as in the regular grid. By adding the system length to each succeeding row of X positions, all nodes are automatically positioned properly for separations to be calculated directly without concern for whether the node in question is on the boundary of the domain. The boundaries of the computational domain have effectively disappeared in the X direction.

The number of storage locations needed to provide guard nodes is much smaller in the skew periodic grid since guard nodes are only needed at the "ends" of the system, not at the "sides." The "hiccups" at the beginning of each row in the regular periodic representation have been eliminated in the skew periodic representation. For example, in Fig. 4a, when the index offsets for the NNT extend to the third layer in all directions, as we must do in some of our molecular dynamics calculations, the 81 cells in the principle domain and the 8 extra 3×3 domains are

necessary to set up enough ghost cells to allow vector operations which span the entire 2D cross section without boundary interruptions or special boundary corrections. In this case, as shown in Fig. 4b, the skew periodic grid again uses many fewer ghost nodes. Three extra rows plus three extra points on each end of the system gives 12 ghost nodes at each end. The total number of points is thus 33 rather than 81, a substantial savings in computer memory.

If skew periodicity is used in all three dimensions, the number of ghost cells at each end is three planes (27 points) plus three rows (9 points) plus 3 points. Thus the skew periodic 3D grid has 105 points altogether of which 27 are active. The same system represented in a regular periodic grid needs a total of 729 points of which only 27 are active. Of course the relative difference is smaller when the active MLG is larger than $3 \times 3 \times 3$ but the total amount of wasted storage increases rapidly with the size of the system.

6. NNT Optimization and Vectorization Based on the Near Miss Probability

The discussion of "near misses" presented here concerns the system described in Section 3. These 512 noninteracting particles represents a worst case for NNT optimization in significant applications of near neighbor algorithms. For example, in molecular dynamics the interaction force becomes repulsive at small separations giving each particle a "size." It is more likely that a particle several NNT nodes away will be within R_c of the focal node if it is a finite-size particle rather than zero sized. For finite-size particles, there is zero probability for more than one particle to occupy the same position in space. A system of non-interacting nodes demonstrates two significant aspects of MLG indexing: (1) One is able to construct an optimal NNT in terms of the average volume about the focal node containing nearest neighbors, i.e., an approximate sphere; and (2) that NNT indexing via offsets relative to the focal node permits the computation of node–node interactions using vector and parallel processing methods.

The first stage of NNT optimization, i.e., minimal scaling in terms of Eq. (3.2), is determining an upper bound on the MLG index offsets of nearest neighbors via statistical analysis. Such an analysis is shown in Table III for the 512-node system for an interaction cutoff radius $R_c = 4$ Å. Because of symmetry, only a portion of the NNT is considered for each of the logical planes.

Table III shows that for nodes in the logical plane $k = 3$ there is essentially zero probability for a "near miss." Further, for the logical planes $k = 0$, 1, and 2, the contours of constant probability are seen to be roughly circular. The next stage of NNT optimization is considering only those logical nodes for which the indexing of "close" nodes is possible. For the system described in Table III, this suggests using an NNT having a roughly hemispherical shape. The NNT shape can be selected by storing in separate data arrays the logical offsets of near neighbors defined according to the skew periodic indexing scheme and the desired NNT shape. The

TABLE III

Near Miss % Probability of a Neighbor Coming within $R_c = 4$ Å as a Function of NNT Offset ($8 \times 8 \times 8$) System

$k=0$							
$j+3$	0.000	0.000	0.0039	0.000	0.000	0.000	0.000
$j+2$	0.000	0.0055	0.0585	0.0678	0.0499	0.0051	0.000
$j+1$	0.0008	0.0511	0.7610	2.115	0.5980	0.0226	0.000
j				F.N.	2.277	0.0316	0.000
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$
$k=1$							
$j+3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$j+2$	0.000	0.0031	0.0365	0.0517	0.0134	0.0062	0.000
$j+1$	0.000	0.0258	0.3319	0.7240	0.2696	0.0241	0.000
j	0.000	0.0183	0.5355	2.062	0.8929	0.0593	0.000
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$
$k=2$							
$j+3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$j+2$	0.000	0.000	0.0062	0.0021	0.000	0.000	0.000
$j+1$	0.000	0.0031	0.0260	0.0385	0.0161	0.000	0.000
j	0.000	0.0036	0.0114	0.0281	0.0478	0.0021	0.000
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$
$k=3$							
$j+3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$j+2$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$j+1$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
j	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$

spatial offsets are the fixed separations of the periodic image nodes. This procedure is described in Fig. 6a and b. In this figure the array containing the index offsets, IJOFF(IPT, K2), is computed outside the timing loop. A similar array is defined for the spatial offsets which are used in the interaction calculations.

In Fig. 6a and 6b, IJN is the maximum node index in the skew-periodic grid in logical plane k and NPT is the maximum number of nodes indexed by the NNT in logical plane K2. The procedure in Fig. 6a, however, is not optimum for vector oriented computers and does not utilize the vector compatibility of the skew-periodic indexing scheme. Vector and pipeline processors allow the inner DO-loops of a computing procedure to be "vectorized." The procedure shown in Fig. 6b is equivalent to that shown in Fig. 6a but is structured to take full advantage of the vector attributes of skew periodicity.

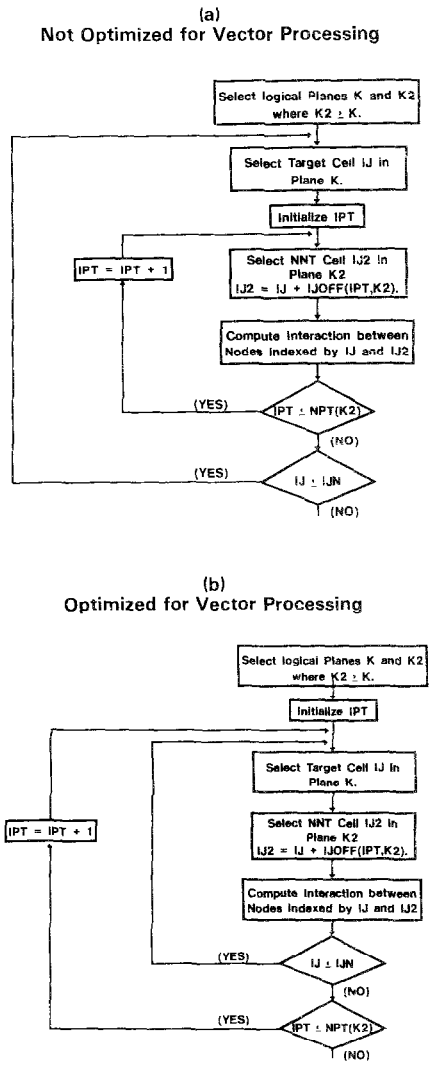

FIG. 6. Steps of interaction calculation between two logical planes.

7. ANALYSIS OF SWAPPING (RANDOM MOTION)

There are two important measures of swapping to restructure a given node system. These are: (1) the total number of swaps per timestep in all iterations and (2) the average number of swapping iterations required to reorganize the MLG each timestep. These features depend on the extent to which the MLG indices are perturbed from monotonicity each timestep by node motions. Each swapping

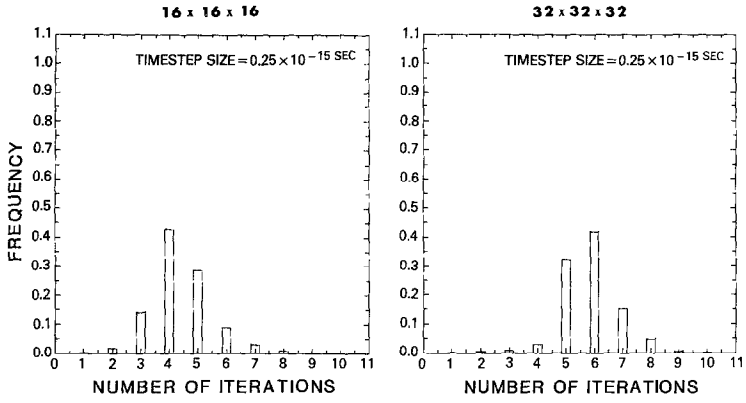

FIG. 7. Distribution of swapping iterations as a function of timestep.

iteration consists of checking ajacent pairs of MLG indices and swapping those indices which are not monotonically ordered. More than one sweep through the MLG is required in general since only adjacent indices are compared. The amount of work to restore monotonicity by swapping is therefore a function of (1) the timestep and (2) the number of nodes in the system. An increase in timestep results in a larger perturbation of the monotone indexing. An increase in system size increases the upper bound on the amount of total reordering which may be required to restore monotonicity.

The statistical results presented here for swapping are for node systems consisting of non-interacting points. A qualitative analysis of the dependence of swapping on "interaction strength" was undertaken by introducing central forces between the system nodes. The swapping required was found to be less than that for non-interacting nodes because now nodes often rebound without passing. Thus, for applications such as molecular dynamics, a system of non-interacting nodes represents a worst case.

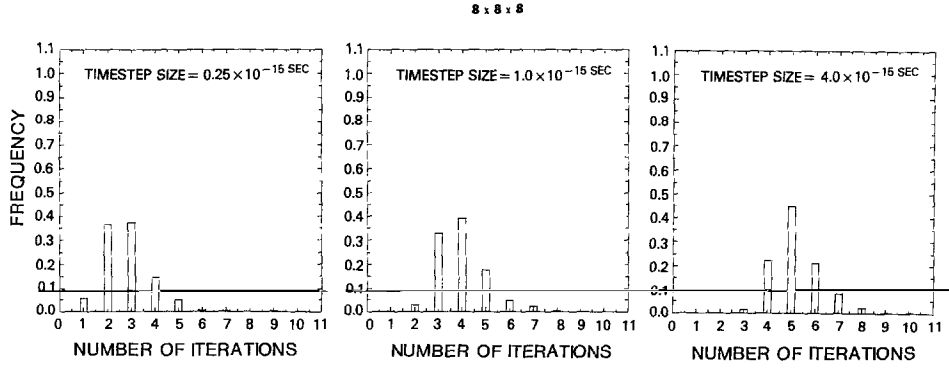

FIG. 8. Distribution of swapping iterations as a function of system size.

A quantitative picture of the convergence of swapping iterations for this system of non-interacting nodes is given in Fig. 7 and 8. Shown in Fig. 7 are distributions of swapping iterations for three different timesteps differing by a factor of 4. As expected, the relatively larger timesteps require more swaps to restore the MLG indices to monotonicity. A sixteenfold increase in timestep, however, occasions only a factor of two increase in the number of iterations. Thus, to integrate for a given time, the longer timesteps are actually much more efficient. When the nodes travel several average spacings per timestep, however, the number of iterations will increase linearly with the timestep.

Shown in Fig. 8 are distributions of swapping iterations for systems having $(16 \times 16 \times 16)$ and $(32 \times 32 \times 32)$ nodes, respectively. The larger system, as expected, requires more swaps to maintain MLG order for a given timestep. However, for the larger system, the convergence rate for swapping is still comparable to that for the smaller system.

For molecular dynamics and other manybody problems the timesteps used in the calculation of Fig. 7–9 are unrealistically large since the accuracy for real orbits has not been considered. An analysis of the swapping iteration was undertaken for timesteps having sizes suitable for such applications. For these cases no swapping was observed for a significant fraction of the time increments. For those timesteps where swaps were required, the maximum swapping iteration level was about 2.

The $N \log_2(N)$ scaling of the number of swapping iterations is demonstrated in Fig. 9 for three different timesteps.

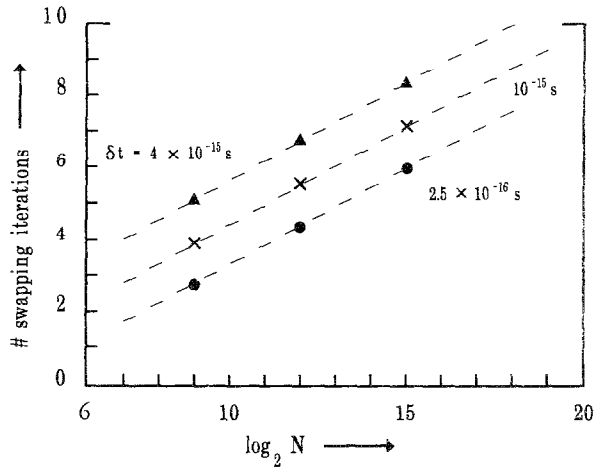

FIG. 9. Swapping iterations as a function of system size.

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research and the Naval Research Laboratory. The authors wish to thank Drs. Michael Page and Michael Picone who were always available to discuss various questions. We are pleased to acknowledge with thanks the contributions of Keith Roberts who started one of us (JPB) on the path leading to this work. The authors would also like to thank Lauree Shampine, Patricia Reed, Francine Rosenberg, and Elizabeth Gold for their assistance with various aspects of this report.

REFERENCES

1. R. W. HOCKNEY AND J. W. EASTWOOD, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981), Chap. 8, pp. 267.
2. J. BORIS, *J. Comput. Phys.* **66**, 1 (1986).
3. J. P. BORIS AND S. G. LAMBRAKOS, in *Proceedings, State-of-the Art, Free-Lagrangian Methods Meeting*, 1985, edited by M. J. Fritts *et al.* (Springer-Verlag, New York, 1985), p. 185.
4. J. P. BORIS AND S. G. LAMBRAKOS, in *Proceedings of the* 1985 *Summer Computer Simulation Conference, Chicago, Illinois*, 1985 (North-Holland, Amsterdam, 1985), p. 206.
5. S. G. LAMBRAKOS AND J. P. BORIS, NRL Memorandum Report 5761, 1986 (unpublished).
6. A. W. APPEL, *SIAM J. Sci. Statist. Comput.* **6**. No. 1, 85 (1985).
7. W. F. VAN GUNSTEREN *et al.*, *J. Comput. Phys.* **5**, 272 (1984).
8. J. P. BORIS AND K. V. ROBERTS, *J. Comput. Phys.* **4**, 4 (1969).
9. J. M. PICONE, S. G. LAMBRAKOS, AND J. P. BORIS, *SIAM J. Sci. Statist. Comput.*, in press.
10. S. LAMBRAKOS, J. BORIS, R. GUIRGUIS, M. PAGE, AND E. ORAN, *J. Chem. Phys.*, in press.