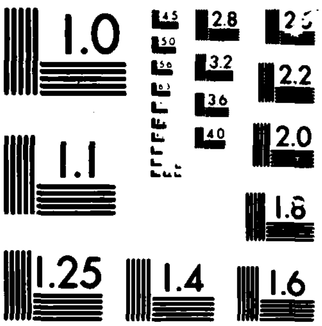
AD-A167 120 GEOMETRIC PROPERTIES OF THE MONOTONIC LOGICAL GRID ALGORITHM FOR NEAR NEIGHBOR CALCULATIONS(U) NAVAL RESEARCH LAB WASHINGTON DC S G LAMBRAKOS ET AL. 1/1

UNCLASSIFIED 24 APR 86 NRL-MR-5761 F/G 12/1 NL



MICROCOP' CHART

2

NRL Memorandum Report 5761

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

S. G. LAMBRAKOS AND J. P. BORIS

Laboratory for Computational Physics

April 24, 1986

This work was supported by the Office of Naval Research.

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5761	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Research Laboratory	6b. OFFICE SYMBOL (If applicable) Code 4040	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000		7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Research	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217		10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
	RR011-09-43		DN280-071

11. TITLE (Include Security Classification)

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

12. PERSONAL AUTHOR(S)

Lambrakos, S. G. and Boris, J. P.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM ______ TO ______	1986 April 24	35

16. SUPPLEMENTARY NOTATION

This work was supported by the Office of Naval Research.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Monotonic Logical Grid Skew-periodic
			Swapping Near-neighbors template

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

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 Logical Grid" (MLG) where adjacent objects in space have close grid indices. Using an MLG to index positions and attributes of objects in computer memory permits a near neighbor algorithm to be based on a "Maximum index offset," N_e, rather than a short range "cutoff" distance R_c. An MLG algorithm removes the necessity of having to test distances. Further, "close" objects will be indexed via contiguous memory. Thus permitting efficient vectorization of computations.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Jay P. Boris	22b. TELEPHONE (Include Area Code) (202) 767-3055	22c. OFFICE SYMBOL Code 4040

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

GEOMETRIC PROPERTIES OF THE MONOTONIC LOGICAL GRID ALGORITHM FOR NEAR NEIGHBOR CALCULATIONS

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. 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 algorithm is available to identify the near neighbors. Consequently, the operation count for this distance checking procedure scales as N^2. Even when an interaction is neglected because $|R_a - R_b| > R_c$, checking the separation distance requires about 10 floating point operations 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 (N_s) of the total number of nodes in the system. Scalar sorting procedures have been developed for this purpose with operation counts scaling linearly with 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 oriented or synchronous, parallel processing super computers. The communications and data structures for these scalar algorithms are also not optimum for the fastest computers available.

Neighbor list techniques [6] which are vectorizable have been developed, however, the structure of these algorithms have large storage requirements and are thus not applicable to large systems.

Boris [2] and Boris and Lambrakos [3] have developed an algorithm for keeping track of near neighbor interactions and geometric relationships which scales as N and is structured to permit optimized vector and parallel processor implementations. This development

Manuscript approved January 19, 1986.

followed from efforts on the "nearest neighbors" problem begun with Dr. K.V. Roberts [7] at Culham Laboratory in the context of gravitationally attracting stars. There we choose a field-solver approach to finding the forces not only because of the long range nature of the gravitational force but also because of the long range nature of the gravitational force but also because a good nearest neighbors algorithm was lacking.

The new algorithm uses a Monotonic Logical 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. When two adjacent nodes pass each other in real space, relative to the chosen coordinate direction, 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 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 depends on 1) its easy vectorization; 2) the rapid convergence of the "swapping" procedure, to restore the MLG; 3) the average distance the nodes travel between MLG reorderings being large; 4) the number of interactions for each node; and 5) the computational cost of computing them. In our tests and applications to date the cost in computer time 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 resetting 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 domain which is periodic in X and Y and bounded in Z by two reflecting walls. The nodes are non-interacting 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 particular near neighbor indexing grid we have used, the "skew-periodic" MLG designed to facilitate long vector operations. This paper examines an MLG comprised of NZ identical logical 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 §4.

"Swapping" to maintain the monotone mapping is an iterative process. Its convergence rate depends on the extent to which relative node positions are perturbed between restoration of the MLG, i.e. how far the nodes move, and on the size of the grid. Analysis of "swapping" iteration level frequencies show a better integrated convergence rate for "large" changes in the positions of the nodes, i.e. long timesteps resulting in significant MLG distortion. This result is extremely encouraging as it implies that the MLG swapping procedure is also a good way to order completely 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 and fluid dynamics which require calculation of near neighbor interactions for a large number of nodes whose relative positions can change in a variety of ways.

2. Near Neighbors Template for Monotonic Logical Grid Consisting of Indexed Planes

The "order N" scaling of the MLG algorithm is effected by restricting the computations of node-node interactions to 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 "target node" (TN). If a particular node is taken as the target, the remaining nodes of the "template" define a local "pattern" in the MLG corresponding to the relative index offsets of what can be assumed to be near neighbors of the target 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 since all interactions with nodes having a negative address offset will have been calculated previously when those nodes were target nodes. Three shells of interaction are defined in Fig. 1 corresponding (approximately) 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 × 5 × 5 cubical template shown in Fig. 1. nominally includes 125 nodes. Since the target node does not interact with itself and each interaction does not need to be counted twice, there are $\frac{(125-1)}{2} = 62$ interactions considered for each node. To complete the shell of circle points requires considering nodes still further from the target cell 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 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 nearmiss, that is a node which is outside the template being found within R_c of the target node, is acceptably small.

The characteristics of a Near Neighbors Template will depend on the particular monotone mapping implemented between 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 property of MLGs provides latitude for further optimization with respect to particular problems. For example, an optimum MLG for one problem may minimize distances to near neigbors. In another problem, the MLG may be optimized when that the shortest distance to non-near neighbors is maximized. This paper examines the properties of a simple NNT of the type shown in Fig. 1., applied to a skew-periodic MLG.

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 logically adjacent? How does this average depend on the offsets of the MLG indices between the two nodes?

(2) How does the necessary configuration and size of a 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 to compute interactions based on a particular MLG.

The appropriate NNT of course depends on the specific MLG scheme selected. The MLG considered in this analysis is a skew-periodic grid as described in §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 "target node". An example of the computational cost coefficient in front of the order N scaling of the MLG algorithm is given in the appendix.

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, etc.

(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. The velocities of these points was random and uniformly distributed in each coordinate from -1×10^7 cm/sec to 1×10^7 cm/sec . The spatial domain. 80 Å × 80 Å × 80 Å, corresponds to an average separation of adjacent nodes of approximately 10 Å in each coordinate, roughly the density of gas near STP.

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

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

The distance classification interval DR is adjusted according to the number of timesteps used.

Shown in Fig. 2. are probability distributions for neighbor separations corresponding to two NNT offsets. In computing these distributions all 512 nodes were sampled. The distance classification interval DR for each of these distributions was .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} second. This time interval is sufficiently long that the high velocity nodes easily traverse the spatial domain, i.e. 80 A, several times. Fig. 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 target node and the interaction node.

Shown in Fig. 3. are frequencies for neighbor separations at short range corresponding to an NNT cell having a relatively large index offset from the target node, i.e. the probability distribution in Fig. 2. corresponding to NNT offset (-3,1,0). As can be seen, even for relatively large index offsets from the target 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 required NNT size.

Additional information concerning neighbor positions is provided by cumulative integrals over f(R;i,j,k) from 0 to a given separation from the target node. These integrals give the probability the field nodes with a given offset from their target node come within a particular distance of the target 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 is effected via analysis of the near miss frequency for each NNT *offset*, i.e. the cumulative probability of finding a node indexed outside the NNT but within a distance R_c of the target node. Shown in Fig. 4. are the near miss probabilities as a function of NNT offset for the particular case $R_c = 3$Å. These probabilities were computed with the same sampling used for the frequency functions shown in Figs. 2 and 3. NNT optimization based on near miss probability is discussed in §5.

For the system of nodes represented in Fig. 4, i.e. a system where the cutoff radius $R_c = 3$Å, all NNT cells more than three index offsets from the "target cell" in any direction have zero probability of recording a "near miss", i.e. coming within 3 Å of the target 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 (referring to the appendix) the total number of node-node interactions computed each timestep for a MLG of size N is

$$\text{Computed Interactions} = 171 \times N - 294 \times (N^{\frac{2}{3}}). \qquad (3.2,$$

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 Fig. 4, further deletion of NNT cells is possible using an NNT which is more nearly spherical. This aspect of NNT optimization is examined presently for the "vector compatible" skew-periodic MLG.

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

More than one type of MLG for indexing nodes within a given spatial domain exists. In particular, for nodes in a 3D spatial domain with periodic boundary conditions, a Monotonic Logical Grid can be constructed which permits more efficient vector manipulation of node attributes than the regular periodic grid. This so called "skew periodic" MLG also results in an average partitioning of the computational domain in cells of equal statistical volume.

"Position skewing" is a statistical property of a system of nodes such that the average relative location of neighbor nodes is a function of the direction of the MLG coordinate offset relative to the target node. These properties 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, as shown clearly in Figs. 7 and 8. It is shown presently that position skewing depends on the MLG configuration and diminishes with increased size of the node system.

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 Logical Grid, obtains when the node locations in space and the node indices in the computer memory satisfy a set of monotonicity conditions,

$$\begin{array}{rlcl} & X(i,j,k) < X(i+1,j,k) & ; & 1 < i < NX-1 \\ & Y(i,j,k) < Y(i,j-1,k) & ; & 1 < j < NY-1 \\ \text{and} & Z(i,j,k) < Z(i,j,k-1) & ; & 1 < k < NZ-1. \end{array} \qquad (4.1)$$

Here N, the total number of nodes, equals $NX \times NY \times NZ$. 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 (4.1), although mathematically satisfactory for organizing random locations in space, is not optimal for mapping node positions into a vector computer memory. A skew periodic MLG is described in §4 which is more efficient.

The MLG is comprised of a set of logical planes, NZ, 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 $Z(k) < Z(k-1)$ for 1 < k < $NZ - 1$. Given that $N = NX \times NY \times NZ$, each logical plane of the MLG will index $NX \times NY$ nodes randomly located in 2D space. Complete compact vectorization of each plane is then effected by indexing in monotonic order the locations of the "first" NX nodes and selected periodic images of the other $(NX) \times NY - 1$ nodes which are all at assigned distances from the actual node positions. This indexing scheme provides a mapping of 2D space onto a single continuous MLG coordinate axis. Such a mapping satisfies slightly different monotonicity conditions from (4.1).

$$\begin{array}{rlcl} & X(ij,k) < X(ij-1,k) & ; & 1 < ij < NX \times NY \\ & Y(ij,k) < Y(ij-NX,k) & ; & 1 < ij < NX \times NY \\ \text{and} & Z(ij,k) < Z(ij,k-1) & ; & 1 < k < NZ. \end{array} \qquad (4.2)$$

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

Consider $NX \times NY$ nodes located in a doubly-periodic 2D region of space having an area $Lx \times Ly$. Letting NX and NY equal 3 here for presentation purposes the $NX \times NY$ regularly spaced nodes and their periodic images are shown in Fig. 5.

The MLG indices in circular brackets, i.e. (ij), represent the location of the nodes which form the Regular Periodic Monotonic Logical Grid. The square brackets, i.e. $[ij]$, represent the locations of some of the node images, each of which is a distance $L = ((mLx)^2 + (nLy)^2)^{\frac{1}{2}}$ $(m,\ n = 0, \pm 1, \ldots)$ 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. 5) interrupt the adjacency of data in computer storage and hence interrupt numerical optimization through vectorization. The Skew-Periodic Monotonic Logical Grid is formed by the nodes and node images whose spatial locations satisfy the modified constraints (4.2). This is depicted in figure 6.

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" NX points and the monotonically ordered positions of the images of the remaining $(NX) \times NY - 1$ points were fit to a straight line using least-squares. The configuration sketched in Fig. 6 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.

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 $\frac{1}{8} = \frac{DY}{NX \times DX} = 0.125$. 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. 7. 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 "position skewing" resulting from the average area partitioning associated with a 2D skew periodic MLG is described using Fig. 8. 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 in the X direction is traversed.

The average relative spatial position of logical neighbors is a function of the MLG indexing scheme. We would like to know where, relative to the target cell, a node having a particular MLG offset is likely to be. For a regular periodic MLG the overall volume partitioning is essentially cubical. 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 parallelograms. That is, the average locations of four adjacently located random nodes (i, j), $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$ is a parallelogram.

The average distance of neighboring nodes from the target node is recorded in Fig. 9a for the system of 512 points with random positions. Note that each NNT node has an average RMS distance to the target node which is only about a factor of $\sqrt{2}$ longer (or less) than the corresponding distance in a perfectly regular Eulerian grid. Shown in Figs. 9b and 9c are the average separations of neighbors in X and Y, respectively. The average X separation is exactly what we would expect, 10 Å. It is seen in these figures that only the average Y separations are affected by skewing as we expect.

Indexing the NNT locations using coordinates (i, j) and taking the target node (TN) as the origin, "position skewing" is examined by taking divided differences between the Y components of neighbors. From Fig. 9c, for a $8 \times 8 \times 8$ system of nodes having random velocities (top number in each cell),

$$\begin{aligned} \frac{[Y(3,3) - Y(-3,3)]}{60} &= \frac{(33.75 - 26.25)}{60} \\ &= 0.125 = \frac{NY}{[NX \times NY]}, \end{aligned} \tag{4.3}$$

where the average node separation in each direction is 10 Å. Because "position skewing" scales as $\frac{1}{NX}$, its affect diminishes as the system size increases. This is a relatively small price to pay for more efficient data storage and vectorization. From Fig. 9c, for a $16 \times 16 \times 16$ system of nodes (bottom number in each cell shown) having random velocities,

$$\begin{aligned} \frac{[Y(3,3) - Y(-3,3)]}{60} &= \frac{(31.88 - 28.13)}{60} \\ &= .0625 = \frac{1}{16}, \end{aligned} \tag{4.4}$$

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 could not be orthogonal in any case because 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 system. 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 not want to use this technique in all directions. When the vectors become long enough, vectorizing in two dimensions at once is no longer so attractive. On a Cray, for example, vectors of length 64 are long enough.

On arrays of parallel processors, actual wraparound is sometimes implemented in hardware and skew periodic connectivity is often extended to a number of dimensions in the hypercube architecture. In these cases the difficulties and costs overcome by the skew periodic MLG in many conventional vector and pipelined processors would be absent so using the regular periodic grid might be simpler.

Although the physical system being described in the two representations, i.e. Figs. 5 and 6, 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 above the nodes 1, 2 and 3. In the skew periodic grid, each successively higher row of nodes (in this case $NY = 3$ rows) is displaced a full system width Lx 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 therefore disappeared in the X direction.

The number of extra storage locations needed to provide enough ghost nodes is much smaller in the skew periodic grid since ghost 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. As shown in Fig. 10a, 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 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. 10b, 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 cells 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 node system described in §3. The system of 512 noninteracting points represents a worst case for NNT optimization in significant applications of near neighbor algorithms. For example, in molecular dynamics studies the interaction is a force which becomes repulsive at small separations. It is more unlikely that a particle several NNT cells away will be within R_c of the "target particle" if it is a finite-size particle rather than zero sized. There is zero probability for more than one molecule to occupy the same position in space. A system of non-interacting points demonstrates two significant aspects of MLG indexing: 1) One is able to construct an optimal NNT in terms of the average volume about the target node containing nearest neighbors, i.e. an approximate sphere; and 2) That NNT indexing via offsets relative to the target cell, 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 (3.2), is to determine via statistical analysis an upper bound on the MLG index offsets of nearest neighbors. Such an analysis is shown in Fig. 11 for the 512 node system for an interaction cutoff radius $R_c = 4\mathring{A}$. Because of symmetry, only a portion of the NNT is considered for each of the logical planes.

Figure 11 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 to consider only those logical nodes for which the indexing of "close" nodes is possible. For the system desribed in Fig. 11, this would suggest using an NNT having roughly the shape of a hemisphere. The NNT shape can be selected by storing in separate data arrays both the logical and spatial offsets of near neighbors defined according to the skew periodic indexing scheme and the desired NNT shape. The spatial offsets are the fixed separations of the periodic image nodes. This procedure is described in Fig. 12a. 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. 12a, 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. 12a, however, is not optimum for vector oriented computers and does not utilize the vector compatibility of the skew-periodic indexing scheme. Vector processors are designed such that the inner DO-loops of a computing procedure are "vectorized". The procedure shown in Fig. 12b is equivalent to that shown in Fig. 12a but is structured to take full advantage of the vector attributes of skew periodicity.

7. Analysis of Swapping (Random Motion)

There are two important measures of "swapping" for a given node system. These are: (1) The total number of "swaps" per timestep, and (2) the convergence rate for the "swapping", i.e. 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. 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 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.

A quantitative description of the convergence rate of swapping iterations for a system of non-interacting nodes is given in Figs. 13 and 14. Shown in Fig. 13 are distributions of swapping iterations for three different timesteps which differ 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.

Shown in Fig. 14 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 Figs. 14 and 15 are unrealistically large since the question of accuracy for real orbits has not been addressed. 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. 15 for three different timesteps.

Appendix

Derivation of (3.2), an example of the linear scaling of the total number of pair interactions with the size of the system. The coefficient of this scaling is a function of NNT size.

Let the MLG indexing nodes in the system consist of N_g logical planes each consisting of $N_g \times N_g$ logical cells arranged in an $N_g \times N_g$ array. Next, let the spatial domain be doubly periodic in X and Y and bounded in Z by two reflecting end walls. And finally, let N_c be the upper bound on logical offsets included in the NNT. Given these conditions, the number of node-node interactions computed each timestep can be described using the following semitriangular array of logical plane index, k, pairs.

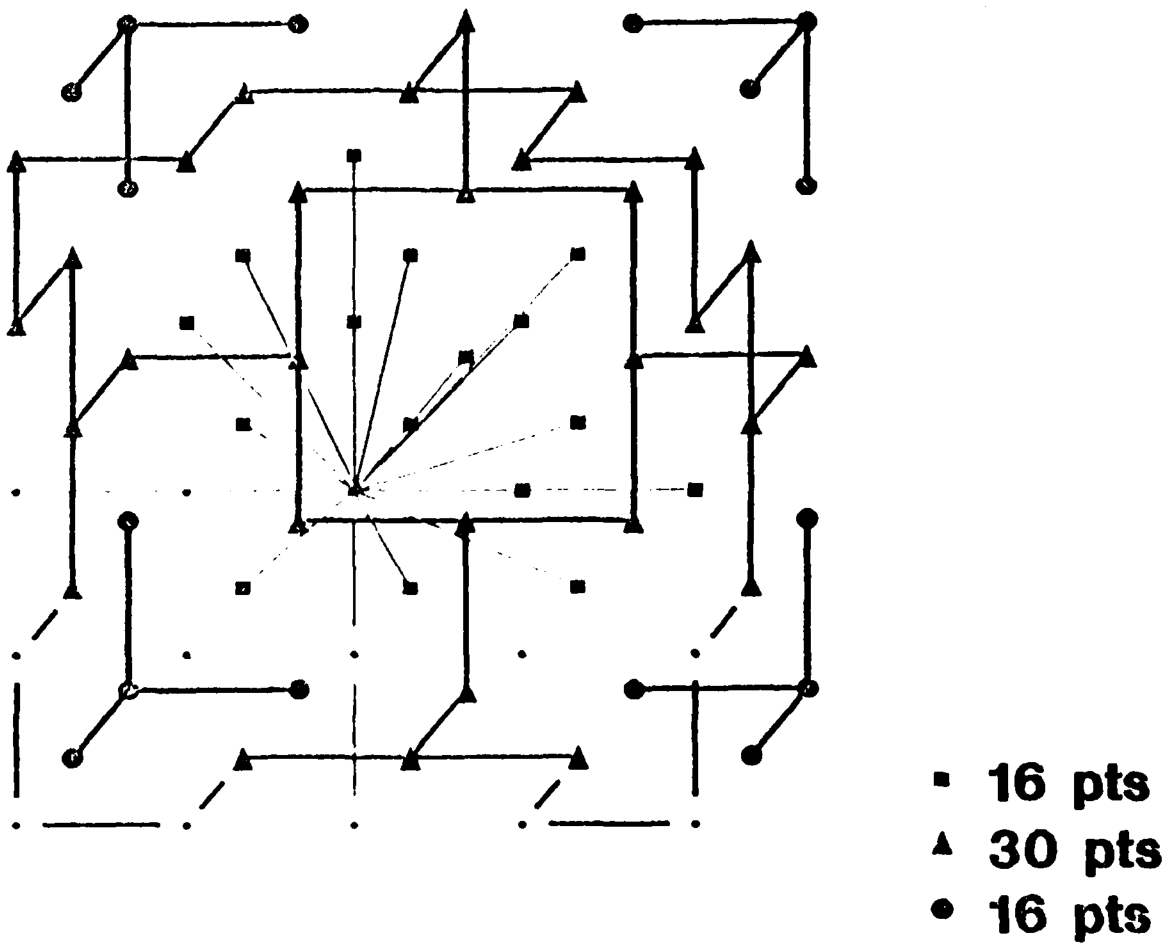
$$\begin{array}{cccccccc}
(1,1) & (2,2) & (3,3) & (4,4) & (5,5) & (6,6) & \cdots & (N_g,N_g) \\
(1,2) & (2,3) & (3,4) & (4,5) & (5,6) & \cdots & (N_g-1,N_g) & \\
(1,3) & (2,4) & (3,5) & (4,6) & \cdots & (N_g-2,N_g) & & \\
\vdots & \vdots & \vdots & \vdots & & & & \\
(1,Nc-1) & \cdots & (N_g-N_c,N_g) & & & & &
\end{array} \tag{A.1}$$

These index pairs represents all possible inter- and intraplane node-node interaction calculations. The number of different inter- and intraplane k index pairs are $(N_c \times N_g - \frac{N_c(N_c-1)}{2})$ and N_g, respectively. For each k index pair corresponding to interplane calculations, i.e. (k1,k2) where k1 $\neq$ k2, one can associate $[(2N_c-1)^2] \times [N_g^2]$ intercell calculations. Similarly, for each k index pair (k1,k2) where k1 = k2, one can associate $[2N_c^2 - 2N_c] \times [N_g^2]$ intercell calculations. It follows that the total number of node-node interactions computed each timestep is

$$\begin{aligned}
\text{Computed Interactions} = {} & [N_c \times N_g - \frac{N_c(N_c-1)}{2}][(2N_c-1)^2] \times [N_g^2] \\
& + N_g[2N_c^2 - 2N_c] \times [N_g^2].
\end{aligned} \tag{A.2}$$

Letting N represent the total number of nodes, i.e. $N = N_g^3$, (3.2) follows from (A.2) by setting $N_c = 3$.

LOGICAL DISPLACEMENTS OF NEAREST NEIGHBORS



August 1984

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

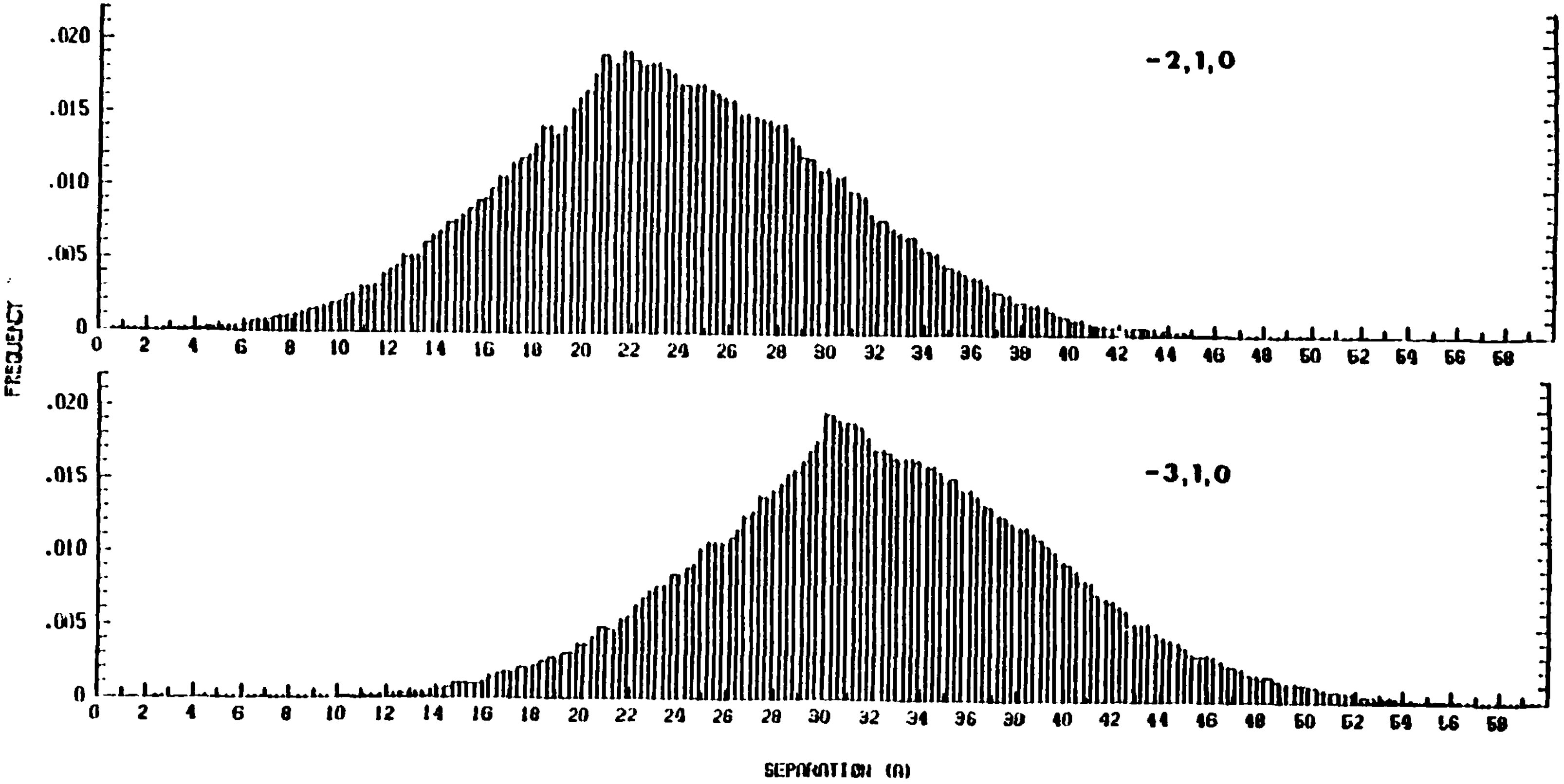


FIG. 2. Probability distribution of neighbor separation corresponding to NNT cell index offsets (-2, 1, 0) and (-3, 1, 0). The bin size for these distributions is 0.3 Å.

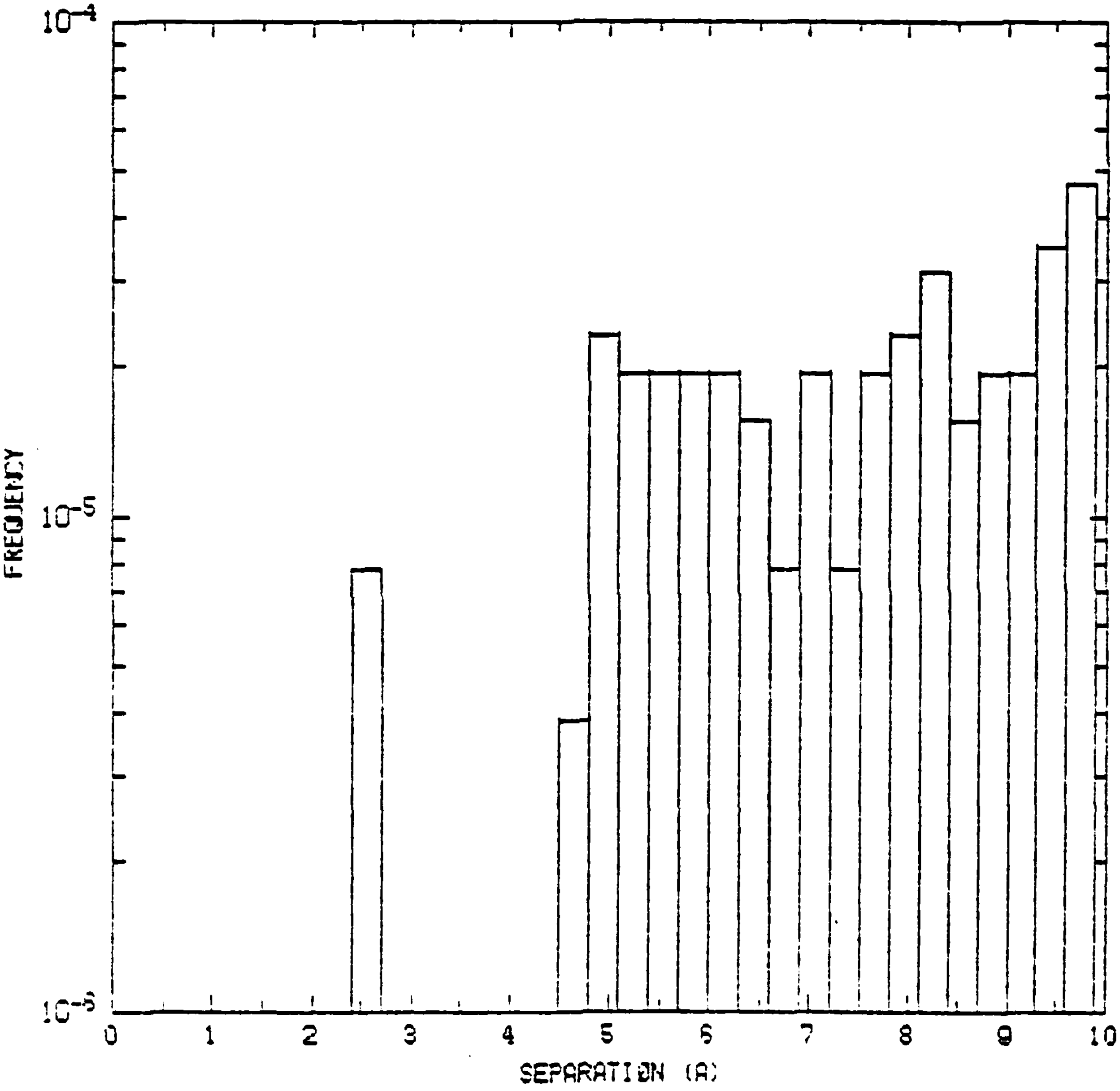


FIG. 3. Frequencies for neighbor separations at short range corresponding to relatively large index offsets, i.e. (-3, 1, 0), from the target cell. These frequencies are the same as those shown in Fig 2 for the range of distance 0 to 10 Å.

J + 3	0. 0	0. 0	. 0023	0. 0	0. 0	0. 0	0. 0
J + 2	0. 0	. 0016	. 0222	. 0175	. 0218	0. 0	0. 0
J + 1	. 0008 *	. 0195 **	. 3138	. 9886	. 2737	. 0066	0. 0
J				T. N.	1. 061	. 0125	0. 0
				i	i + 1	1 + 2	i + 3

*, ** --- Frequency distributions corresponding to these cells are shown in Fig. 2.

FIG. 4. Near miss % probability of coming within $R_c = 3\AA$ as a function of NNT offset. $(8 \times 8 \times 8)$.

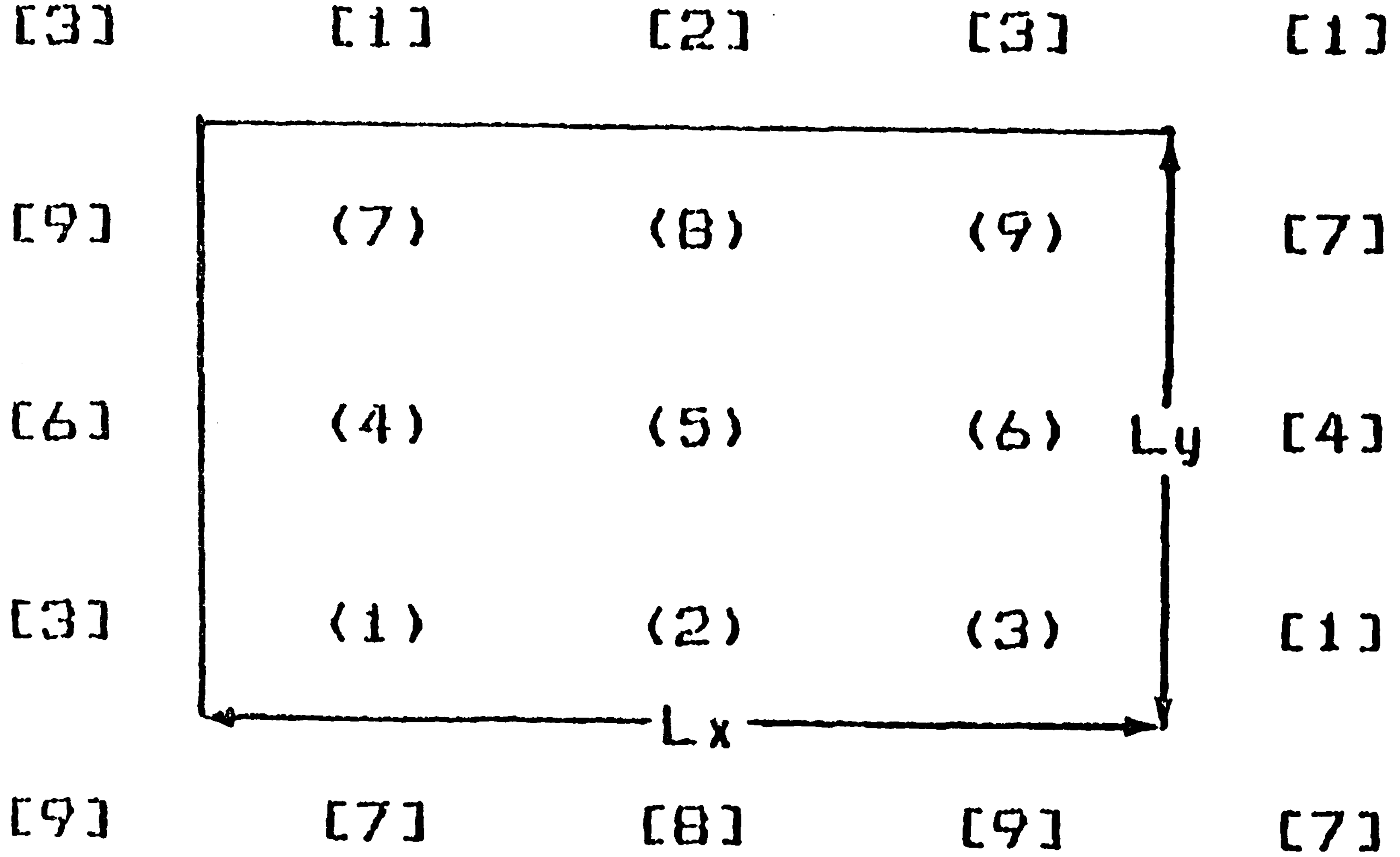


FIG. 5. Node positions in 2D space. Regular periodic MLG.

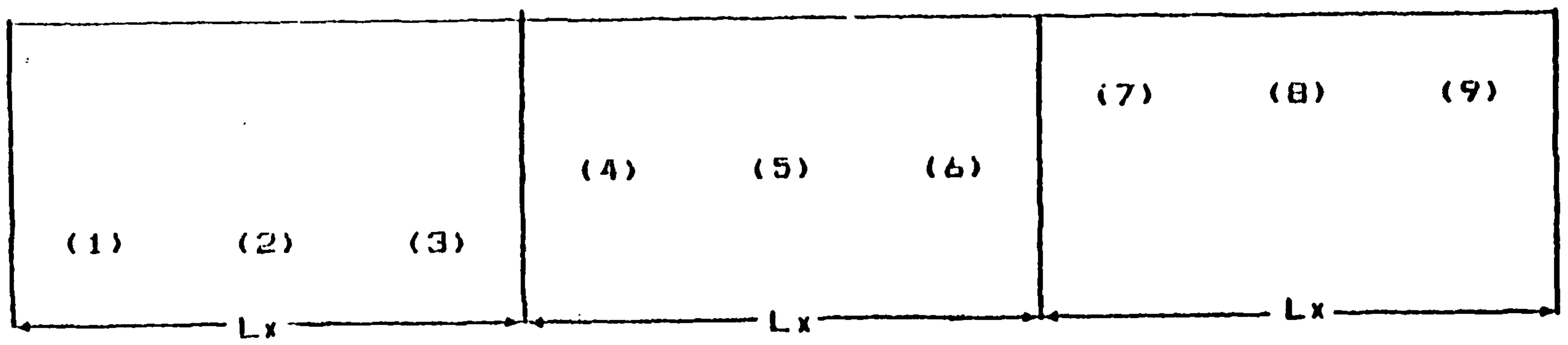


FIG. 6. Node positions in 2D space. Skew periodic MLG. The skew periodic MLG in this figure and the regular periodic MLG in Fig. 5 index the same node positions.

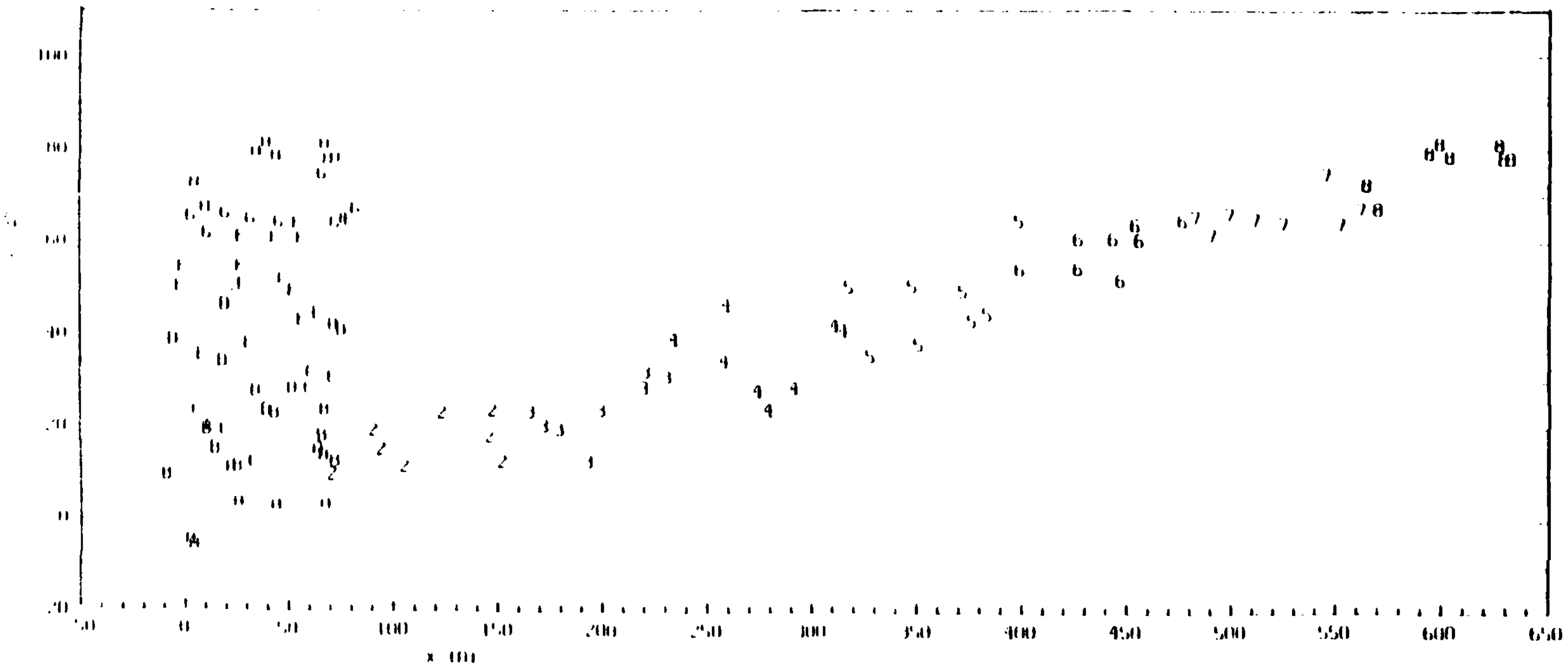


FIG. 7. Comparison of node positions indexed by regular and skew periodic MLG's.

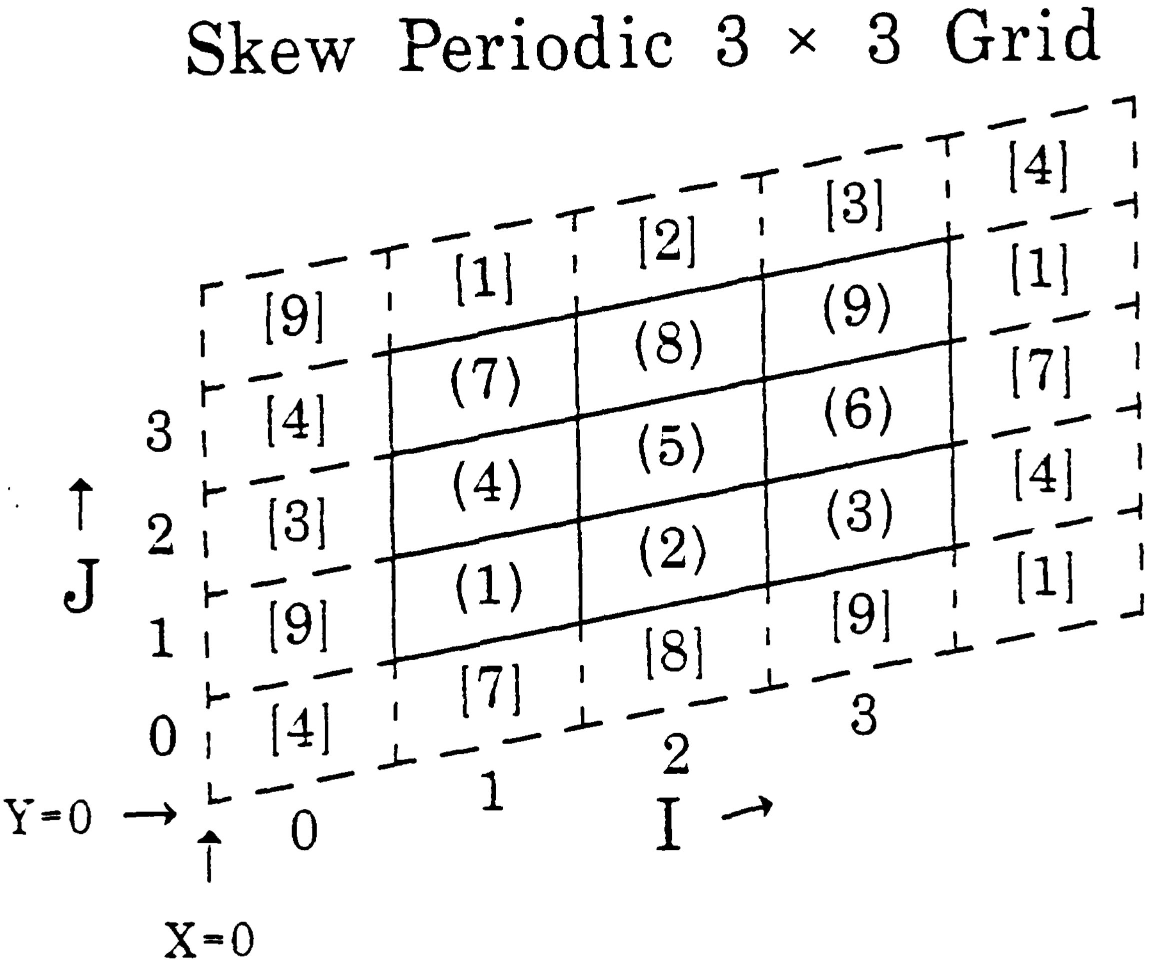


FIG. 8. Average area partitioning via parallelograms.

j + 3	41.20 42.65	35.54 36.77	32.13 32.95	31.70 31.96	34.37 34.05	39.52 38.74	46.32 45.20
j + 2	35.69 36.80	28.53 29.56	23.59 24.38	22.44 22.73	25.60 25.36	31.80 31.19	39.63 38.77
j + 1	32.37 33.02	23.73 24.42	16.78 17.43	14.02 14.37	18.12 18.09	25.68 25.40	34.57 34.12
j				T N	14.06 14.42	22.59 22.79	31.96 32.04
				i	i + 1	i + 2	i + 3

FIG. 9a. Average distance of neighbors from target node as a function of NNT offset. The top and bottom numbers in each grid cell correspond to (8 × 8 × 8) and (16 × 16 × 16) systems, respectively.

J + 3	-30. 00 -30. 00	-20. 00 -20. 00	-10. 00 -10. 00	. 0004 . 0000	10. 00 10. 00	20. 00 20. 00	30. 00 30. 00
J + 2	-30. 00 -30. 00	-20. 00 -20. 00	-10. 00 -10. 00	. 0000 . 0000	10. 00 10. 00	20. 00 20. 00	30. 00 30. 00
J + 1	-30. 00 -30. 00	-20. 00 -20. 00	-10. 00 -10. 00	. 0004 . 0000	10. 00 10. 00	20. 00 20. 00	30. 00 30. 00
J				T. N.	10 00 10. 00	20. 00 20. 00	30. 00 30. 00
				i	i + 1	i + 2	i + 3

FIG. 9b. Average separation in X of neighbors from target node as a function of NNT offset. The top and bottom numbers in each grid cell correspond to $(8 \times 8 \times 8)$ and $(16 \times 16 \times 16)$ systems, respectively.

j + 3	26.25 28.13	27.50 28.75	28.75 29.38	30.00 30.00	31.25 30.63	32.50 31.25	33.75 31.88
j + 2	16.25 18.36	17.50 18.91	18.75 19.46	20.00 20.00	21.25 20.67	22.50 21.31	23.75 21.95
j + 1	6.250 10.34	7.500 10.56	8.750 10.69	10.00 10.00	11.25 11.59	12.50 12.35	13.75 12.98
j				T. N.	1.250 6250	2.500 1,250	3.750 1.875
				i	i + 1	i + 2	i + 3

FIG. 9c. Average separation in Y of neighbors from target node as a function of NNT offset. The top and bottom numbers in each grid cell correspond to ($8 \times 8 \times 8$) and ($16 \times 16 \times 16$) systems, respectively.

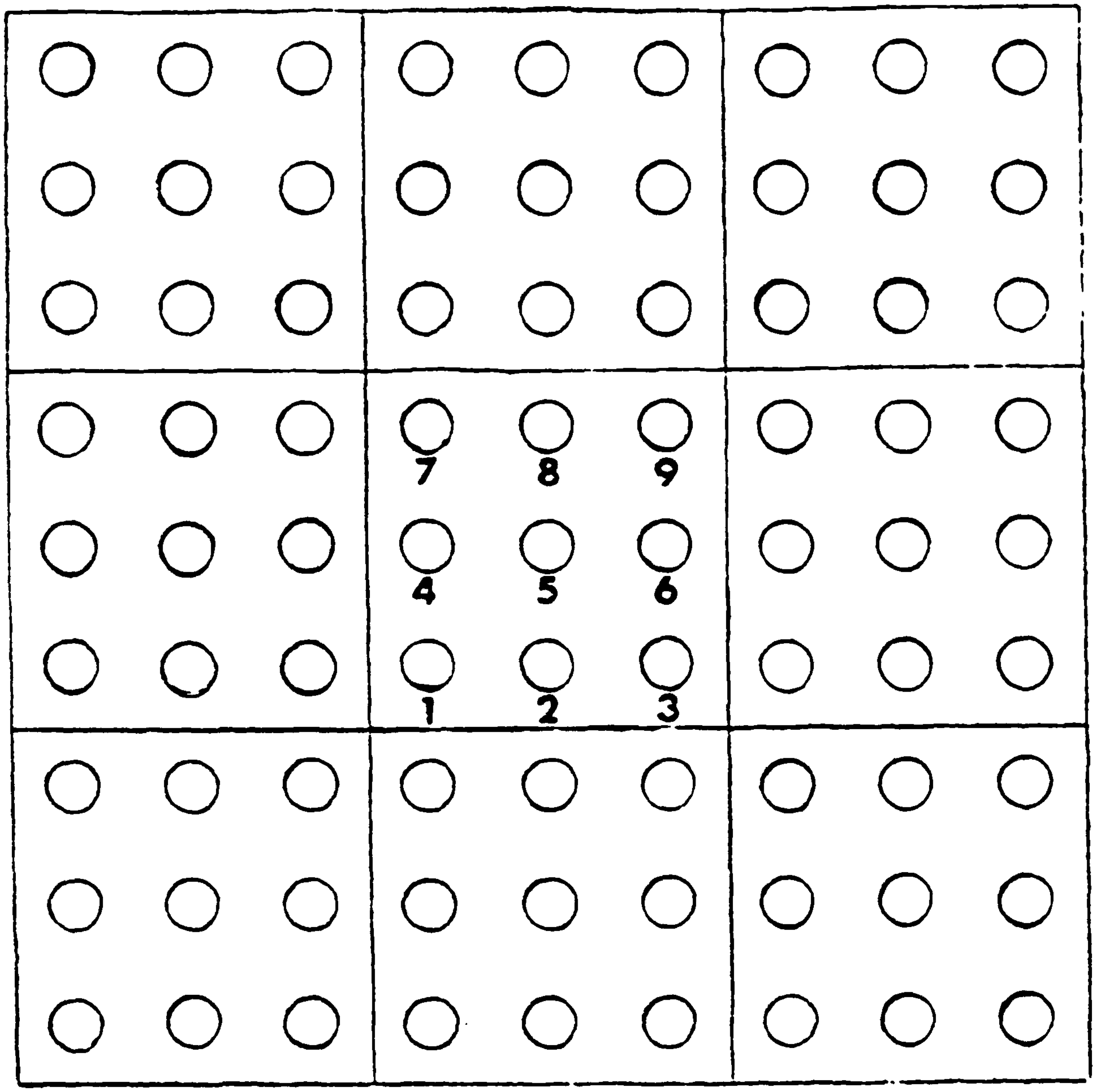


FIG. 10a. Schematic representation of ghost cells required for an NNT having maximum index offsets of three when node positions are indexed via a regular periodic MLG.

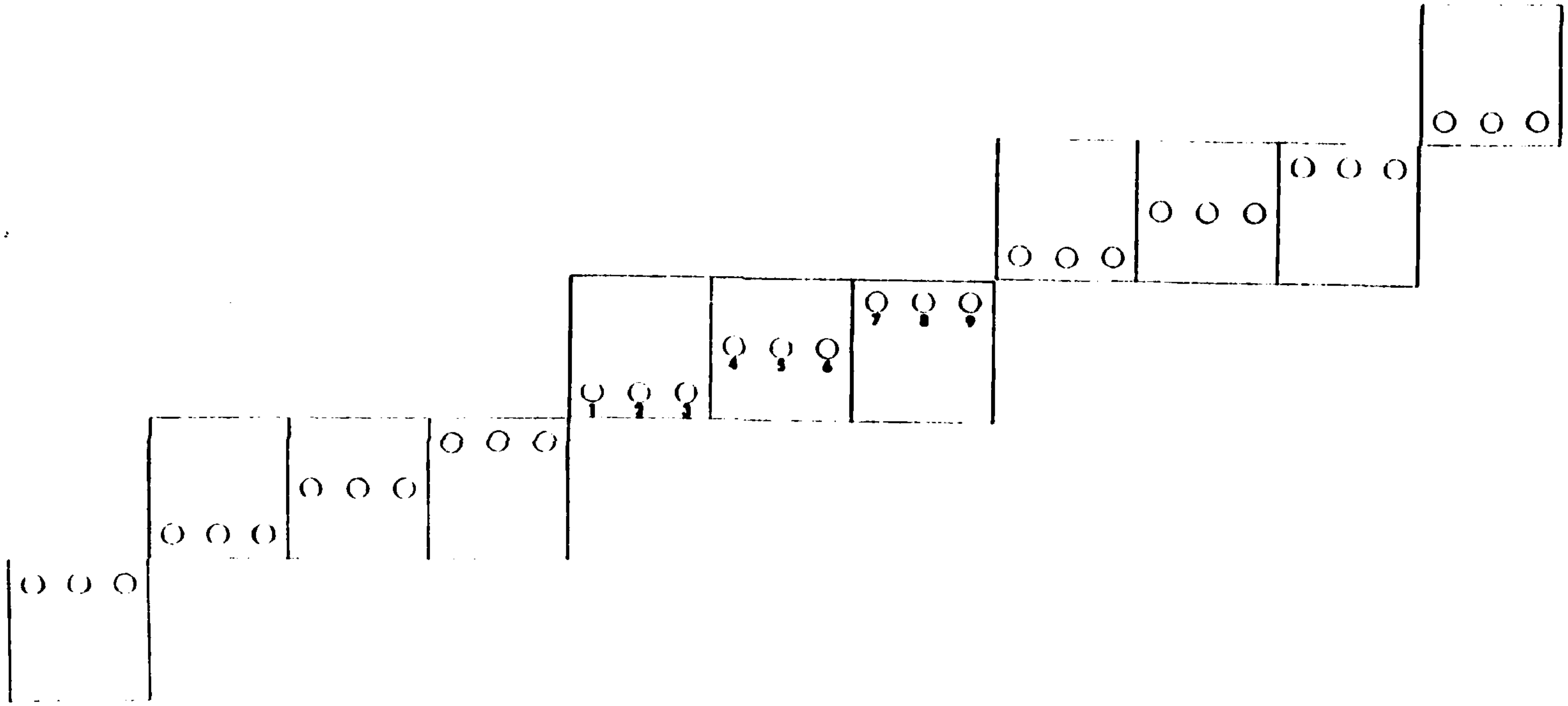


FIG. 10b. Schematic representation of ghost cells required for an NNT having maximum index offsets of three when node positions are indexed via a skew periodic MLG.

j + 3	0.0	0.0	.0039	0.0	0.0	0.0	0.0	
j + 2	0.0	.0055	.0585	.0678	.0499	.0051	0.0	
j + 1	.0008	.0511	.7610	2.115	5960	.0226	0.0	k = 0
j				T N	2.277	.0716	0.0	
j + 3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
j + 2	0.0	.0031	0365	0517	.0134	.0062	0.0	
j + 1	0.0	.0258	3319	7240	2696	0241	0.0	k = 1
j	0.0	.0183	5355	2.062	3929	0593	0.0	
j + 3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
j + 2	0.0	0.0	0062	0021	0.0	0.0	0.0	
j + 1	0.0	0031	0260	.0385	0161	0.0	0.0	k = 2
j	0.0	0036	.0114	0281	0475	0021	0.0	
j + 3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
j + 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
j + 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	k = 3
j	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	i - 3	i - 2	i - 1	i	i + 1	i + 2	i + 3	

FIG. 11. Near miss % probability of a neighbor coming within $R_c = 4\text{Å}$ as a function of NNT offset. ($8 \times 8 \times 8$).

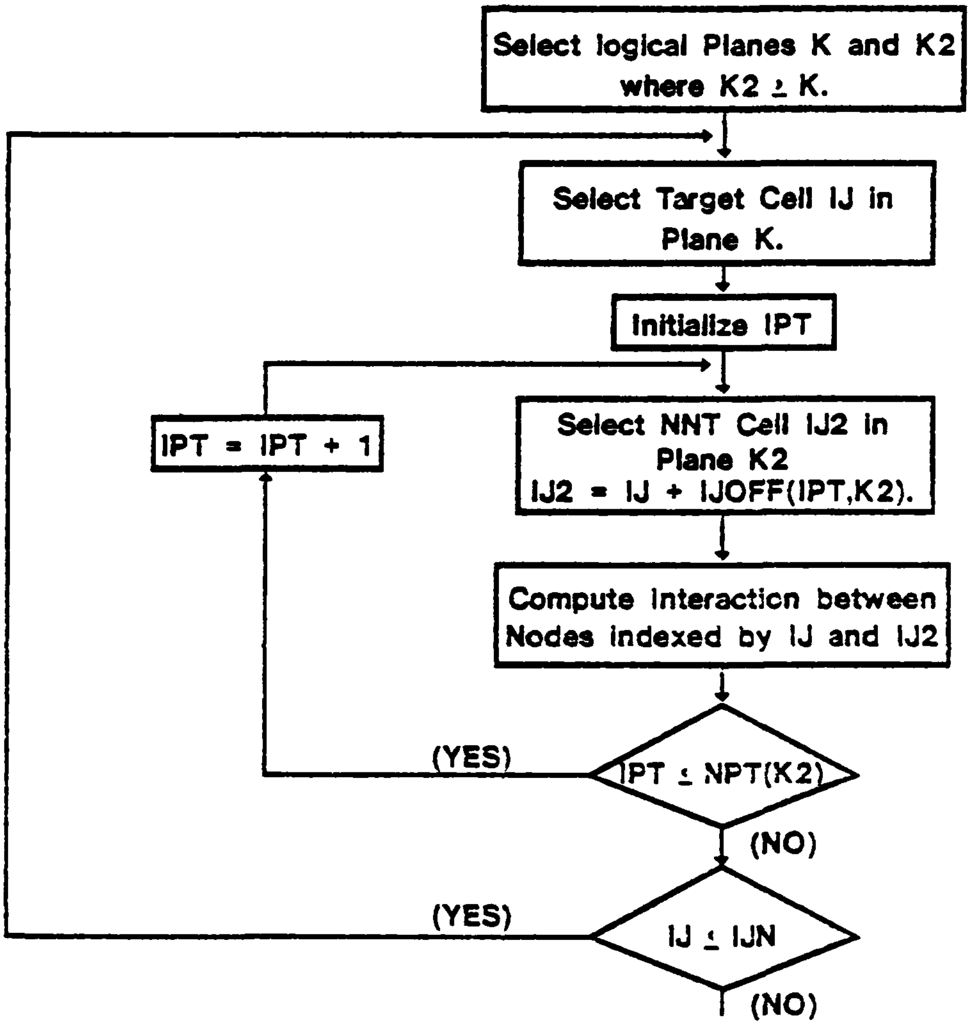


FIG. 12a. Steps of interaction calculation between two logical planes.

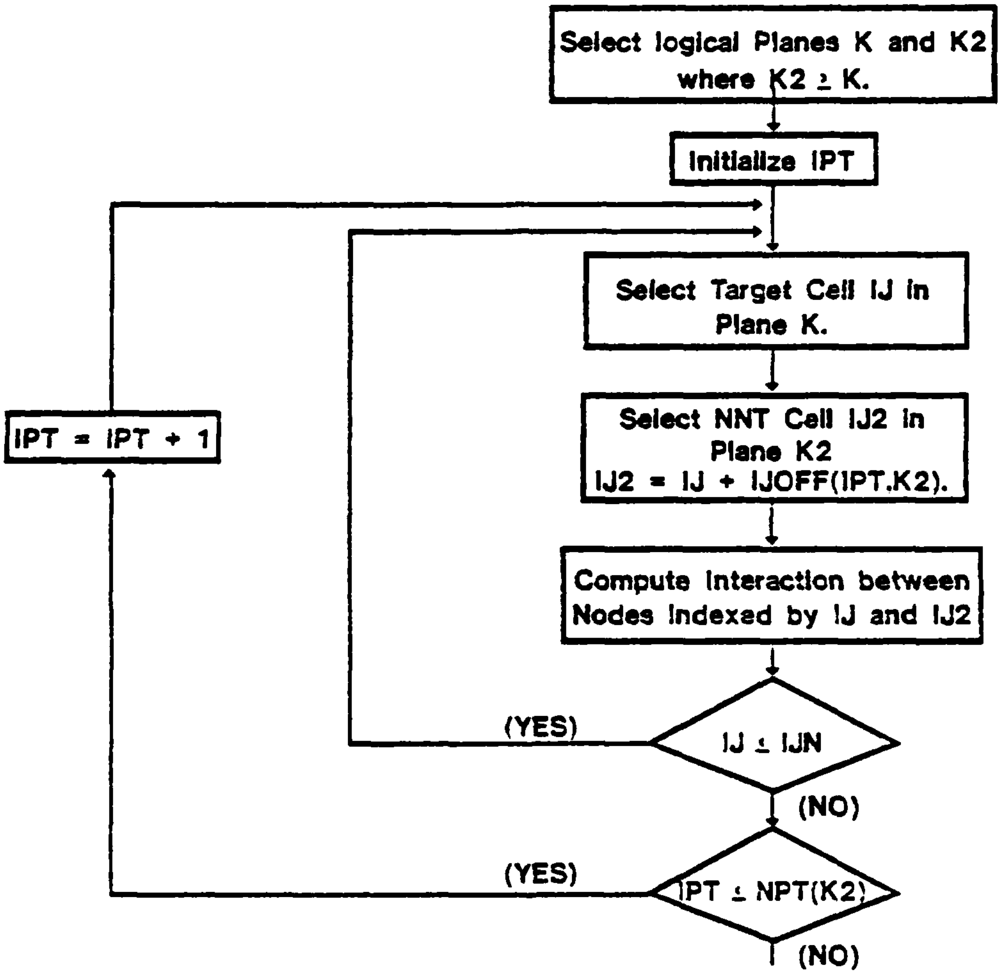

Select logical Planes K and K2 where K2 ≥ K.
Initialize IPT
Select Target Cell IJ in Plane K.
Select NNT Cell IJ2 in Plane K2 IJ2 = IJ + IJOFF(IPT,K2).
Compute Interaction between Nodes indexed by IJ and IJ2
IJ ≤ IJN
(YES)
(NO)
IPT ≤ NPT(K2)
(YES)
(NO)
IPT = IPT + 1

FIG. 12b. Steps of interaction calculation between two logical planes optimized for vector processing.

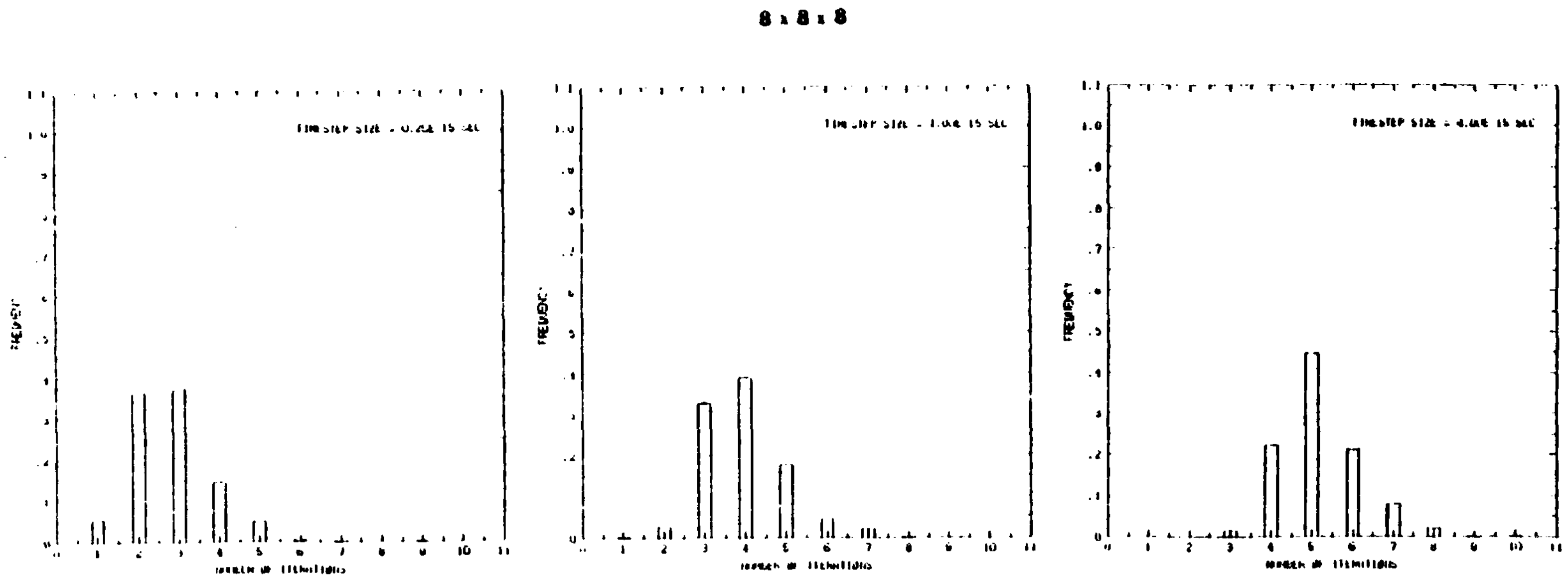


FIG. 13. Distribution of swapping iterations as a function of timestep.

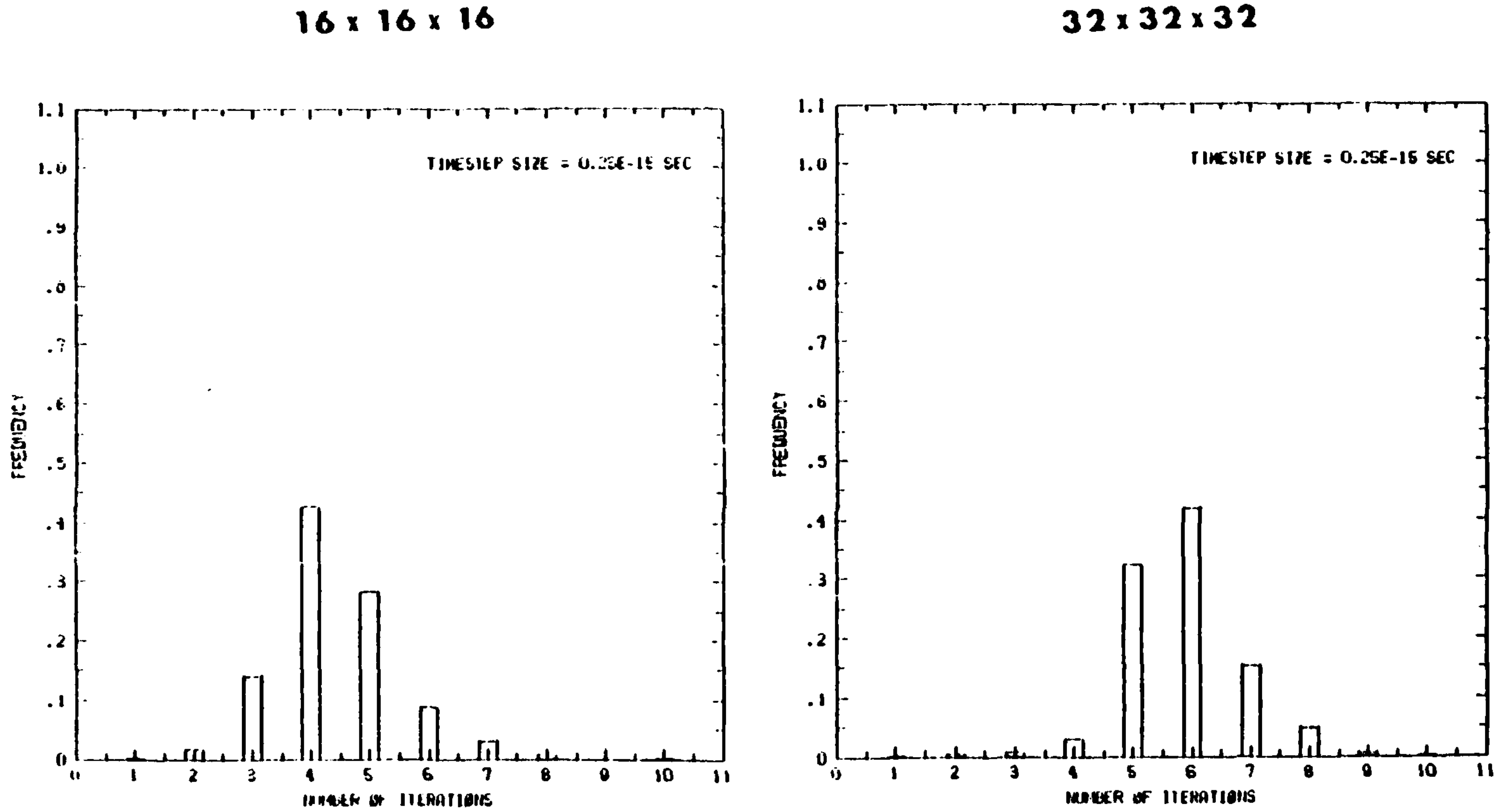


FIG. 14. Distribution of swapping iterations as a function of system size.

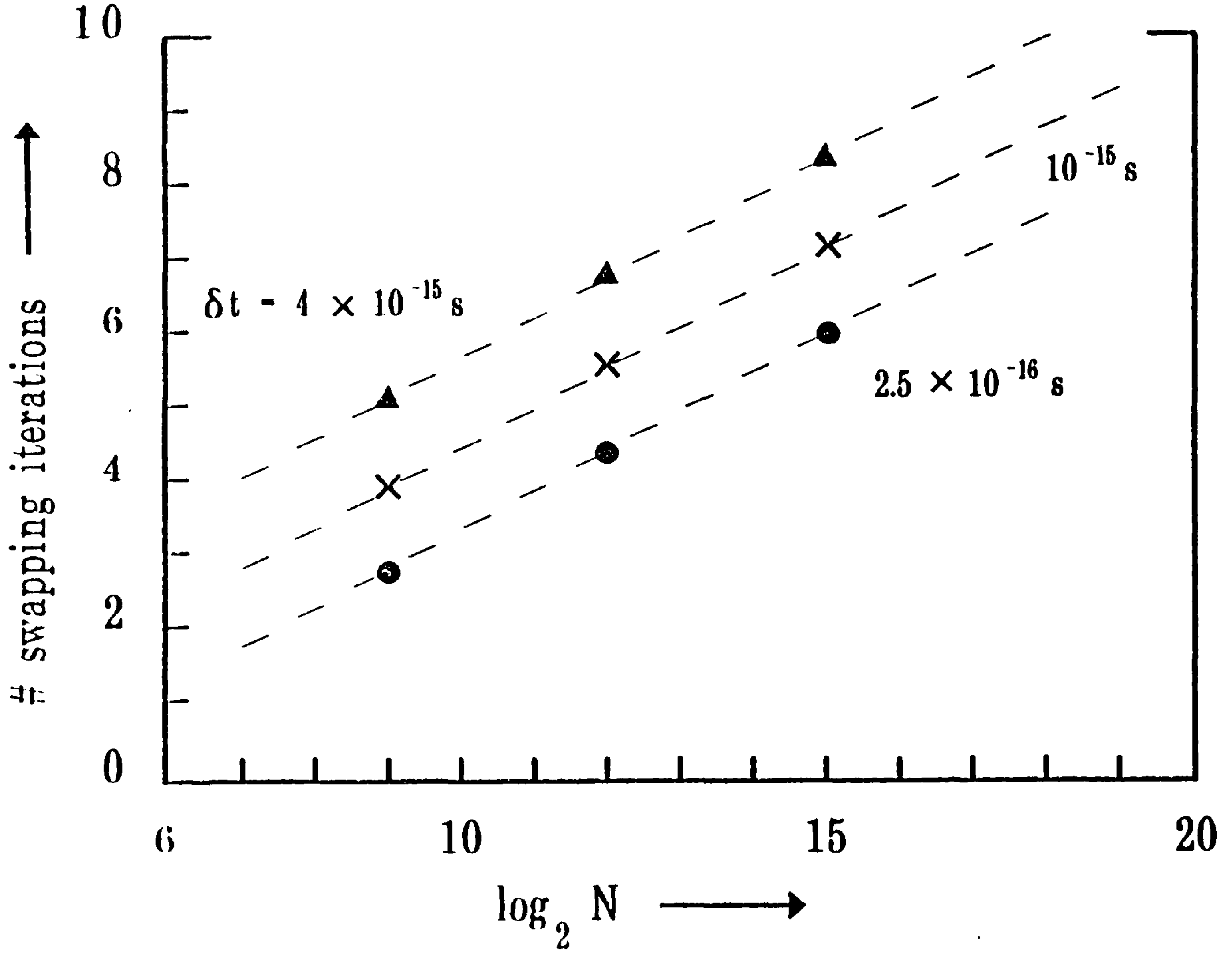


FIG. 15. Swapping iterations as a function of system size.

Acknowledgements

This work was supported by the Office of Naval Research and the Naval Research Laboratory. The authors wish to thank Dr. Michael Page who was 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 parts of this work.

REFERENCES

[1] R. W. Hockney and J. W. Eastwood, Computer Simulation Using Particles, McGraw-Hill Inc., New York, 1981, Chapter 8, pp 267-304.

[2] J. Boris, A Vectorized "Nearest-Neighbors" Algorithm of Order N Using a Monotonic Logical Grid, NRL Memorandom Report 5570, May 1985, J. Comput. Phys., ADA155169

[3] J. P. Boris and S. G. Lambrakos, A Vectorized "Near Neighbors" Algorithm of Order N Using a Monotonic Logical Grid, Proceedings of the State- of-the Art, Free-Lagrangian Methods Meeting, March 1985.

[4] J. P. Boris and S. G. Lambrakos, Dynamical Organization of Evolving Data Using a Monotonic Logical Grid, Proceedings of the 1985 Summer Computer Simulation Conference, July 1985.

[5] A. W. Appel, An Efficient Program for Many-Body Simulation, SIAM J. Sci. Stat. Comput., Vol. 6, No. 1, 85 (1985).

[6] W.F. Van Gunsteren et al., On Searching Neighbors in Computer Simulations of Macromolecular Systems, J. Comput. Chem., 5, 272 (1984).

[7] J.P. Boris and K.V. Roberts, The Optmization of Particle Calculations in 2 and 3 Dimensions, J. Comput. Phys., 4, 4 (1969).