JOURNAL OF COMPUTATIONAL PHYSICS 15, 529–532 (1974)

Note

Generation Technique for Chains on Lattices in Computer Experiments

It is current practice to use inversely restricted sampling methods to obtain configurational averages concerning nonintersecting chains on lattices in computer simulation experiments. Biased chain configurations are built by taking only the steps that do not lead to occupied lattice sites. The bias is removed by an appropriate weighting of the chain properties in the calculation of averages. The method is described by Hammersley and Handscomb [1] and interesting examples are given by Mazur and McCrackin [2].

The aim of this note is to present a particularly efficient operational mode which simultaneously collects the information needed:

(1) to make the step choices during chain construction;

(2) to compute the interaction energy of the chains with the surroundings;

(3) to determine the bias weights for the configurational averages.

This represents the quasitotality of the computational problem. The method consists essentially in calculating one single coded number before making each step. The coded number enables us immediately to obtain the number of occupied and empty first neighbors of the element last placed in the course of the chain construction, and to choose the direction in which the next step will be made. The computation of the coded number involves only q occupancy tests on the lattice in which the chains are embedded, q being the coordination number. This seems a minimum.

The method is based upon the following obvious properties. From a lattice point i, one can jump to q first neighbors j which can either be occupied or empty, so that to the set of the q neighbors are associated 2^q occupied–empty configurations. These can be numbered N, $N = 1, 2, \ldots, 2^q$; to each configuration corresponds the set of informations consisting in the number of occupied and empty j points, and the directions in which the free j points can be found. (The correspondence between N and a specific configuration will be explained later). All possible sets of informations can be preset in the computer memory, and all that is needed is a unique device to detect the sequence number N.

Consider a chain in the course of construction and let i be the last chain element placed on the lattice. Before starting the calculations leading to the next

construction-step, preset a coded number CØD to 1.00. As usual a matrix is memorized, the elements of which correspond to the different lattice sites. For example, an element is set to 0 if the corresponding site is empty, and to 1 if it is occupied. All the first neighbors j of i are considered in a fixed order $j_1, j_2, \ldots, j_q$; each can be reached formally from i by a step denoted $s = 1, 2, \ldots, q$. For example, for a simple cubic lattice the order can be: $x+$ ($s = 1$); $x-$ ($s = 2$); $y+$ ($s = 3$); $y-$ ($s = 4$); $z+$ ($s = 5$) and $z-$ ($s = 6$), where $x+$ means a step in the positive x direction, etc. If j_s is found to be empty (by examination of the matrix) CØD is incremented by $2^{q-s} + 0.01$. For example, for the simple cubic lattice ($2^q = 64$), if all six points j are empty, CØD becomes 64.06 after six occupancy tests (this obviously can happen only for $i = 1$); if all are occupied CØD remains 1.00 (in that case the chain cannot be completed). Each occupancy pattern of the points j leads to a unique value of CØD. The digits of CØD after the decimal point indicate the number of empty neighbors of i, say q_i. The number of occupied neighbors distinct from the preceding chain point $i - 1$ is simply $\sigma_i = q - q_i - 1$ except for σ_1 which is $q - q_1$ because the first point has of course no predecessor. The digits before the decimal point indicate the specific occupancy pattern N found for the neighbors (in fact, the arithmetic procedure to find CØD automatically and uniquely always associates the same N to the same pattern, thus defining N). Lists of directions s, in which all free points j can be found for each pattern N, are preset in memory. To decide in which j we will place the next chain element $i + 1$, we simply call a random number R between 1 and q_i. Knowing R and N we pick up a well-defined direction.

By repetition of the procedure, a completed chain consisting of r points is characterized by $b = \prod_{i=1}^{r-1} q_i$ which is the bias removal factor for all properties related to the particular chain configuration (by construction, each chain has a probability $1/b$). The interaction energy of the chain with its surroundings is

$$E = \epsilon \sum_{i=1}^{r} \sigma_i ,$$

if one considers only first neighbor interactions ϵ. The configurational average of a chain parameter T is then given by

$$\langle T \rangle = \sum_{m=1}^{M} T_m b_m e^{-E_m/kT} \Big/ \sum_{m=1}^{M} b_m e^{-E_m/kT}$$

where m indicates a chain configuration built following the procedure described above.

The figure represents the example of the simple cubic lattice, on which most of the experiments are performed. The tree indicates all the routes which can be

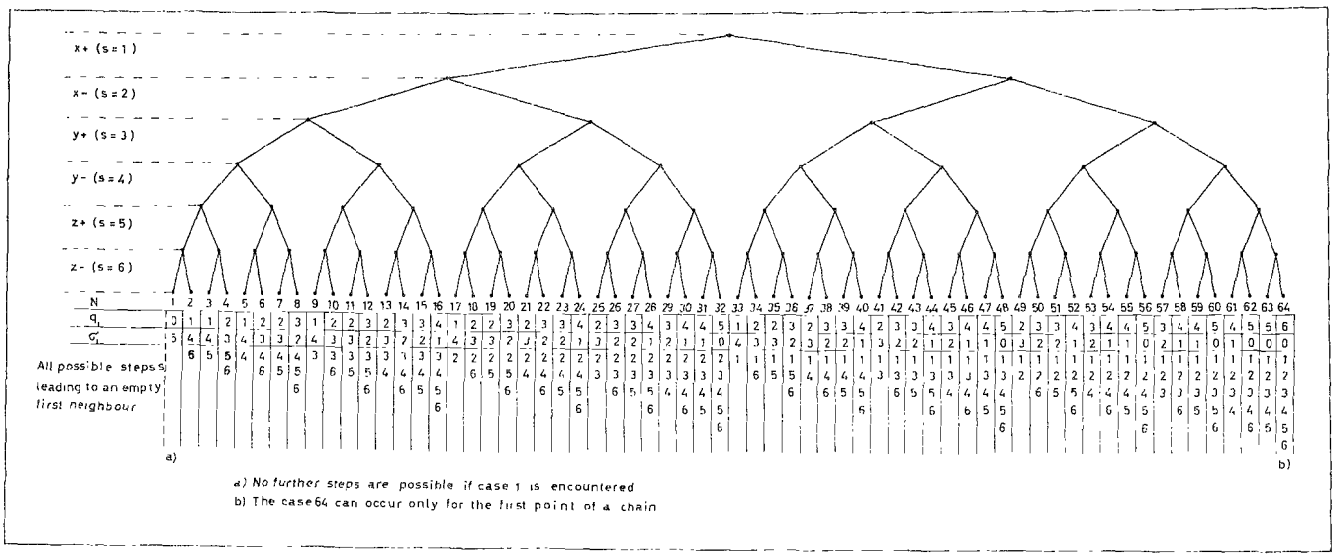

FIG. 1. Tree indicating the different routes leading to the 64 occupancy patterns of the first neighbors of a point on a simple cubic lattice.

followed in the computation of one CØD value, by making a single occupancy test on each of the 6 first neighbors j of a point i. The order in which the neighbors are considered is as cited in the text. A step to the left in the tree is to be made if a neighbor is found to be occupied. A step to the right is made if the neighbor is empty.

Each terminal point of the tree corresponds to a particular occupancy pattern N of the six neighbors of i. The associated values of q_i and σ_i are given as well as all the possible steps s leading from i to an empty j.

This computation method has been used to calculate the configurational entropy and the mean end-to-end distances in polymer solutions [3, 4].

On a CDC 6500, our procedure makes it possible to perform ~7000 steps per sec, including the calculation of b and E, on a simple cubic lattice with periodic boundaries.

REFERENCES

1. J. M. HAMMERSLEY AND D. C. HANDSCOMB, "Monte Carlo Methods," Chap. 10, Methuen and Co., London, 1964.
2. J. MAZUR AND F. L. MCCRACKIN, *J. Chem. Phys.* **49** (1968), 648.
3. A. BELLEMANS AND E. DE VOS, *J. Poly. Sci.*, Symposium No. 42 (1973), 1195.
4. E. DE VOS AND A. BELLEMANS, *J. Poly. Sci. C* (1974), to be published.

RECEIVED: April 1, 1974

M. JANSSENS* AND E. DE VOS

Department of Physics
Free University of Brussels
Brussels, Belgium

* Boursier I.R.S.I.A.