JOURNAL OF COMPUTATIONAL PHYSICS 27, 291–293 (1978)

Note

Identifying Clusters in Computer Experiments on Systems of Particles

In computer experiments involving molecular dynamics [1] and in statistical mechanical cluster theory [2], especially in the theory of liquids [3], it is sometimes necessary or desirable to identify the clusters in systems of particles.

Clusters may be defined in a number of ways [2]. The definition we use here is equivalent to one given by Stillinger [4], in which one identifies a critical interparticle distance r_c outside of which the interaction between two particles can be neglected. Loosely speaking, a cluster C is defined as the smallest set of particles satisfying the condition: If $i \in C$ and $r_{ij} \leqslant r_c$ then $j \in C$, where r_{ij} is the distance between particles i and j.

More precisely, a cluster C (relative to r_c) of particles, from the universe of the N particles of a system in a given configuration, is defined as follows:

C is a cluster if and only if it has the properties

(1) If $i \in C$ and $r_{ij} \leqslant r_c$ then $j \in C$.

(2) If A is any set satisfying (1), and if i is in both A and C, then $A \cap C = C$ (where $\cap$ indicates set intersection).

The second property guarantees that a cluster C is not made up of two or more disjoint groups which are separated by more than r_c. From this definition, it is straightforward to show that clusters are disjoint and that each particle belongs to exactly one unique cluster.

Particle clusters are identified and stored in a list L of N members which has the property that L_i = index of next particle in the cluster containing particle i. Thus, L consists of disjoint, circular sublists, each sublist containing the members of a distinct cluster. When identifying clusters, once the first member of a cluster is identified, all members are identified, before proceeding to other clusters. Thus, as clusters are disjoint, in seeking new members of a cluster it is sufficient to examine only those particles not yet identified as belonging to any cluster. The algorithm proceeds as follows: L is initialized so $L_i \rightarrow i$, for all i. Each particle is compared to other eligible particles to see if they belong in the same cluster. If they do, the list elements are swapped. For example, suppose i is within r_c of both j and k ($i < j < k$). First, L_i and L_j would be swapped to give: $L_i \rightarrow j$, $L_j \rightarrow i$. Subsequently, L_i and L_k would be swapped, giving: $L_i \rightarrow k$, $L_k \rightarrow j$, $L_j \rightarrow i$. Note that when a new member is added to a sublist, as above, it is always added immediately following the current member being processed. Sublist members are processed in the order in which they

0021-9991/78/0272-0291$02.00/0

appear on the sublist; therefore, any member added at one step will be processed in a succeeding step.

A particle k is eligible for comparison only if $L_k = k$. This is so because if $L_k \neq k$, then either k is a member of a previously identified cluster or has already been identified as a member of the current cluster. In either case k should not be reexamined.

The identification algorithm can be coded very simply in FORTRAN, as follows (some DO limits illegal in most compilers are used for compactness):

```
Initialize L_i = i for i = 1, N.                    (1)
    DO 300 I = 1, N - 1                             (2)
    IF (I .NE. L(I)) GO TO 300                      (3)
    J = I                                           (4)
100 CONTINUE                                        (5)
    DO 200 K = I + 1, N                             (6)
    IF (L(K) .NE. K) GO TO 200                      (7)
    IF (R(J, K) .GT. RC) GO TO 200                  (8)
    CALL SWAP (L(J), L(K))                          (9)
200 CONTINUE                                        (10)
    J = L(J)                                        (11)
    IF (J .NE. I) GO TO 100                         (12)
300 CONTINUE                                        (13)
```

In the above, $R(J, K) = r_{jk}$, $RC = r_c$, and SWAP is a subroutine which exchanges its arguments. The main loop checks each particle i except $i = N$, which is either in a cluster by itself or with one of the first $N - 1$ particles. If $L_i \neq i$ (step 3), i is bypassed because it must belong to an earlier cluster. At step 4, $L_i = i$ and i is the least (i.e., least index) member of its cluster, for otherwise the value i in L_i would have been swapped earlier. In steps 5–12, all of the members of the cluster containing i are found. As each new member is identified (step 8) it is pushed onto the sublist containing i (step 9) following the present member being processed, so it will be processed in a later iteration on the same cluster (step 11). All cluster members have been checked against all possible members when, and only when, step 11 pops the original index i to the top of the sublist. At that time, the sublist contains precisely all of the members of the cluster. On completion, L consists of a collection of disjoint lists specifying the clusters in the system.

To extract an individual cluster from L, one begins with any particle i and collects list entries $L_i = j$ (say), $L_j = k$, etc., until at some point s one finds $L_s = i$. If the

elements of L are flagged as clusters are extracted, successive clusters can be found by beginning with the next unflagged element.

Storage required for this algorithm is just the N locations to hold the list L. Timing varies from proportional to N if all particles are within r_c of particle one to proportional to N^2 if each cluster contains only one particle. Thus, the efficiency tends to be greater for denser systems.

ACKNOWLEDGMENT

I would like to thank the referee for several helpful suggestions and especially for pointing out the existence of Refs. [2, 4].

REFERENCES

1. S. D. STODDARD AND J. FORD, *Phys. Rev. A* **8** (1973), 1504.
2. H. P. GILLIS, D. C. MARVIN, AND H. REISS, *J. Chem. Phys.* **66** (1977), 214.
3. J. BARKER AND D. HENDERSON, *Rev. Mod. Phys.* **48** (1976), 587.
4. F. H. STILLINGER, *J. Chem. Phys.* **38** (1963), 1486.

RECEIVED: May 17, 1977; REVISED: October 12, 1977.

SPOTSWOOD D. STODDARD

Computer Technology Department,
Purdue University at Fort Wayne,
Fort Wayne, Indiana 46805

Printed in Belgium