JOURNAL OF COMPUTATIONAL PHYSICS 67, 223–229 (1986)

Note

The Algorithm for Three-Dimensional Voronoi Polyhedra

1. INTRODUCTION

The Voronoi polyhedra [1–3] are useful tools for studies on geometrical properties of systems of disordered centers. Such systems are interesting for various domains of science: physics [4–6], materials science and engineering [7], geology, biology [8]. The wide use of the Voronoi polyhedra (VP) derives from the fact that they give a natural description of the local surroundings for every center of the system. The VP for a given center is defined as the volume of space containing all points closer to this center than to any other center. In three-dimensional space a VP is a convex polyhedron formed by certain planes drawn perpendicularly to the intercenter vectors at their midpoints. The neighbouring centers making the VP faces are geometrical neighbours of the given center.

The first stage in studying systems of disordered centers is the VP construction (obtaining the Voronoi of Delaunay tessellations). The next stage is studies of characteristics for the constructions obtained. It is obvious that the final aims of investigations determine the choice of the algorithm for computing a VP. At present different algorithms are available for computing both a single VP and Voronoi tessellations.

The algorithms proposed by mathematicians [9–11] are of interest: they are recursive, i.e., a tesselated system of centers (here the Delaunay tessellation) is supplied with (deprived of) a new center and the tesselation is recomputed. This approach is especially convenient in the cases when the overall number of centers may be unknown beforehand. The algorithms proposed by Bowyer and Watson [10, 11] are universal for spaces of various dimensions ($d=2, 3, 4$). However, the most efficient algorithm for a space of a given dimension is to be that developed for this very dimension. For instance, the algorithm by Green and Sibson [9] specialized for $d=2$ proves to be five times as efficient as the universal one by Bowyer for $d=2$ [10].

The algorithms developed by physicists are for computing individual VPs since usually they are aimed at finding the distribution functions for some metric and topological characteristics of VPs. One of the first and simplest algorithms was developed by Finney [12]. First, for a given atom i (from now on the term "atom" will be used instead of "center") a subset of N_s neighbouring atoms is selected. Then the circumcenters are found for all possible assemblies of four atoms, one being the i atom, the others from the subset N_s. The circumcenters closer to i than to any

0021-9991/86 $3.00

other atoms give the VP vertices. This algorithm is especially applicable for homogeneous close packings where a small N_s (about 20) suffices to determine a VP reliably. In this case, the time of computing a single VP increases as N_s^3.

Brostow, Dussault, and Fox [13] have proposed a somewhat more elaborate algorithm with N_s-independent efficiency. For a given atom i one finds first its "direct" neighbours which are used then to construct a "direct" polyhedron, i.e., one whose faces include the midpoints of the vectors between the atom i and its neighbours. Thereafter, the atoms in the sphere with the radius equal to double separation between the i atom and the farthest vertex of the "direct" polyhedron are used to construct the true VP.

"Direct" polyhedrons are employed by Brostow solely as intermediates in computing VPs. However, we have shown [14, 15] such polyhedra (termed by us "simplified" polyhedra) to be of interest for describing local surroundings in disordered systems. Topological variety of the simplified VPs are less noticeable than of the true VPs, and the difference, e.g., between a perturbed crystal and a liquid is more perceptible in statistics of simplified VPs.

Recently Tanemura, Ogawa, and Ogita [16] have published a new algorithm which computes first a dual (polar) [17] polyhedron consisting of a set of Delaunay tetrahedrons (DT) with the common vertex at an atom i. This dual polyhedron is then employed to compute the VP. The first step is to compute the initial DT. As proved, it is formed by the atoms i, i_1, i_2, i_3, where i_1 is closest to the central i; i_2 is determined from the condition of the minimum circumradius for i, i_1, i_2; i_3 is determined from the condition of the minimum circumradius for i, i_1, i_2, i_3. At the second step new DTs are attached in succession to the faces of the DT obtained (i.e., one finds a new atom i_b so that the circumcenter for the atom i_b and three atoms of a given face are at a minimum distance from this face) until all the DTs around the atom i become determined. The time of computing a single VP increases as N_s.

Our algorithm, as well as one by Tanemura *et al* [16], produces at first the initial construction and then computes successively a complete polyhedron. However, both algorithms realize quite different geometrical ideas. We begin with finding the VP faces and simultaneously build its edges and vertices.

2. The Algorithm

Let us consider constructing the VP around a given atom i. First, it is necessary to select the regular set S of the closest N_s neighbours. The number N_s must be so that all the atoms capable of forming a face belong to the set S. The geometrical assumptions used in our algorithm follow from a simple theorem.

THEOREM. *Let $\{P\}$ be the set of planes constructed to find a certain Voronoi polyhedron* (VP). *Let a point x be known to belong to this* VP. *Then, the foot of the*

perpendicular from x to the nearest plane among the $\{P\}$ *belongs, together with its neighbourhood, to a face of the* VP.

Proof. Let the plane $A \in \{P\}$ be the plane nearest to the point x. Let the point x_1 be the foot of the perpendicular from x to A. The point x_1, with its neighbourhood, does not belong to a face of the VP only if there exists a plane $B \in \{P\}$ cutting off the point x_1 from x (this follows from the VP convexity). In this case the points x and x_1 lie on opposite sides of the plane B. The xx_1 distance is then longer than the separation from x to the plane B (xy distance in Fig. 1). However, such a plane does not exist, as the plane A is the nearest one to the point x by the conditions of the theorem and the xx_1 distance is minimum for all the planes of the $\{P\}$. As a result, the point x_1 together with its neighbourhood must belong to the face VP. The theorem is thus proved. The above reasoning is obviously valid in a space of any dimension ($d \geqslant 1$).

The commonly used statement that a nearest plane forms a VP face is a trivial corollary of this theorem. Below we employ an atom i as point $\bar{x}$ at step 1, point X_1 at step 2 and point X_2 at step 3.

2.1. *Finding the Initial Information*

Step 1. Find a point X_1 which is the midpoint of the vector between the atom i and the nearest atom i_1. By our theorem, the point X_1 belongs to the face i_1 of the VP being computed.

Step 2. Try all the atoms $j \in S$ ($j \neq i_1$) and find an atom $j = i_2$ so that the plane i_2 intersecting the plane i_1 gives the line (i_1, i_2) nearest to the point X_1. Find a point X_2 which is the foot of the perpendicular from the point X_1 to this line (see Fig. 2).

Step 3. Try all the atoms $j \in S$ ($j \neq i_1, i_2$) and find an atom $j = i_3$ so that the plane i_3 intersects the line (i_1, i_2) at a point nearest to X_2. This is the first vertex of the VP (i_1, i_2, i_3).

So we have obtained the initial information: the plane giving a face with two intersecting lines which give edges, and a vertex on this face. Starting from this information, one can readily construct the whole VP.

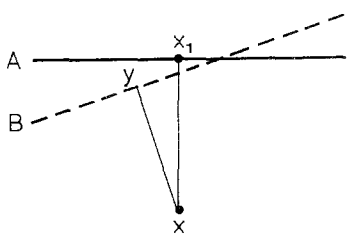

FIG. 1. A two-dimensional illustration of the proof of the theorem. The plane (line) $A \in \{P\}$ (see text) is nearest to the point X belonging to the VP. XX_1 is the perpendicular from the X to the A. The plane B might cut off the point X_1 from the VP. XY is the perpendicular from X to B. $XY < XX_1$ always.

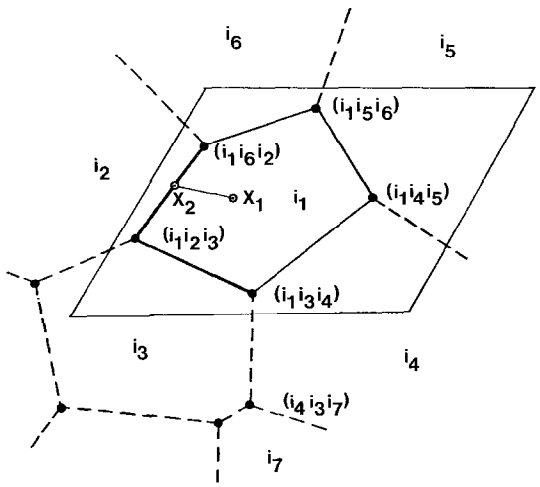

FIG. 2. Illustration of constructing the Voronoi polyhedron: i_1 represents a plane nearest to the atom i; the point X_1 is the foot of the perpendicular from the atom i to the plane i_1. The atom i is not shown. i_2 is a plane which intersects the plane i_1 along a line nearest to X_1; X_2 is the foot of the perpendicular from X_1 to this line. i_3 is a plane intersection the obtained line at a point nearest of X_2. (i_1, i_2, i_3) is the first vertex of the polyhedron. Starting with this information one can compute the whole Voronoi polyhedron (see the text).

2.2. *Closing the Faces: The Basic Idea*

To determine the whole face i_1 one is to find all the contiguous faces (planes giving the edges and vertices at this plane). To this end try all $j \in S$ $(j \neq i_1, i_2, i_3)$ and find an atom $j = i_4$ so that the plane i_4 intersects the line (i_1, i_3) at a point nearest to the vertex (i_1, i_2, i_3) on that side from the plane i_2 which includes the atom i. If the VP is primitive [2] (no degenerate vertices) this plane is the only one. It gives a face of the VP while the intersection point is a new vertex (i_1, i_3, i_4). Then try all atoms $j \in S$ $(j \neq i_1, i_4)$ and find an atom $j = i_5$, so that the i_5 intersects the line (i_1, i_4) at a point nearest to the vertex (i_1, i_3, i_4). It gives the face i_5 and the new vertex (i_1, i_4, i_5). In an analogous way find the next plane and next vertex, etc. The procedure is over when we meet the plane i_2 determining the last vertex on the face i_1, e.g., the vertex (i_1, i_6, i_2) in Fig. 2.

Thus, the face i_1 is fully determined. The rest of the faces are constructed in a similar way, since we have the initial information necessary for the above procedure for every plane which is contiguous with the face i_1. For instance, for the plane i_2 we know two crossing lines formed by the planes i_1 and i_3 affording the vertex $(i_2 i_1 i_3) = (i_1 i_2 i_3)$ and the edges on the face i_2. In fact we have even excess information: on this very plane i_2 we know also the vertex (i_2, i_1, i_6) with the contiguous plane i_6.

When going around the known planes, we determine faces and find new planes forming the VP faces. Having exhausted all the planes determined, we obtain the required VP. As a result of the above procedure, we know the numbers of all faces (atoms affording the VP faces), the order of numbers of continuous faces for every face, the incidence of vertices and faces, and the coordinates of the VP vertices. This information suffices to compute readily any topological and metric characteristic as well as DT and VP tesselations.

2.3. *Details of the Algorithm*

In the above procedure some of a VP vertices are computed thrice (once for every face). However, there is no need for unnecesary computations. For instance, let us go around the face i_4 starting from the vertex (i_1, i_3, i_4) (see Fig. 1). Prior to trying all the planes in order to find $j = i_7$, where i_7 gives the vertex (i_4, i_3, i_7), it is necessary to make sure whether the face i_3 has already been constructed. If so, the vertex $(i_3, i_4, i_7) = (i_4, i_3, i_7)$ and the plane i_7 have already been obtained too. Therefore, it is possible to use this vertex and the face i_7 without trying atoms from the set S.

3. Degeneracy

The present algorithm is to calculate primitive VPs, i.e., those with three faces converging at every vertex [2]. The author has a variant of the program for a degenerate case. It is necessary to assume that the vertex obtained may be formed by several planes, and we must go around all of them. However, as shown by experience in computing, the probability of finding a degenerate VP for disordered systems is practically zero. Hence the use of the algorithms, which are specially developed for the nondegenerate case and are more efficient and simpler than universal ones, is justified for most of problems.

4. Efficiency of the Algorithm

The efficiency of the main step in our algorithm is $O(vN_s)$, where v is the number of a VP vertices, i.e., for every VP vertex one has to try atoms from the set S. Similarly, the efficiency of Tanemura's algorithm is $O(tN_s)$, where t is the number of tetrahedrons forming a dual polyhedron. In this sense, both algorithms are of the same efficiency, since $v = t$. The weak dependence of the efficiency on N_s results from that both algorithms employ geometrical assumptions which are not obvious. Our procedure is based on our theorem (Sect. 2), Tanemura *et al.* [16] make use of four other theorems.

The above estimates of the algorithm efficiencies are rather rough. The CPU time of a VP depends essentially on proportionality factors which can hardly be estimated a priori. Moreover, these can differ by several times depending on the way of writing the program for a computer of a given type.

The CPU times per polyhedron on the same computer were compared [16] for the algorithms from Refs. [12, 13, 16]. The algorithm by Tanemura *et al.* was stated [16] to be more efficient than those by Finney [12] and Brostow *et al.* [13].

Another important characteristic of the algorithm is its simplicity and shortness. On this respect, for example, the Finney algorithm is rather good, although its efficiency is N_s^3-dependent.

Our algorithm is based on single general theorem. This makes it possible to realize the algorithm in a compact form. Our algorithm is implemented as two subroutines. The first one selects a regular set S around a given atom taking into account periodic boundary conditions. It requires some 100 FORTRAN statements. The subroutine computing a VP requires 250 FORTRAN statements, about 100 of them being a unit for computing the initial information. About 10 % of the statements are comments. Detailed information on the algorithm and the program is available on private request.

ACKNOWLEDGMENTS

The author thanks the referees for their useful comments.

REFERENCES

1. G. L. DIRICHLET, *Z. Reine Angew. Math.* **40**, 216 (1850).
2. G. F. VORONOI, *Z. Reine Angew. Math.* **134**, 198 (1908).
3. C. A. ROGERS, *Packing and Covering* (Cambridge Univ. Press, London, 1964).
4. M. TANEMURA, Y. HIWATARI, H. MATSUDA, T. OGAWA, N. OGITA, AND A. UEDA, *Prog. Theor. Phys.* **58**, 1079 (1977).
5. C. S. HSU AND A. RAHMAN, *J. Chem. Phys.* **70**, 5234 (1979); **71**, 4974 (1979).
6. J. N. CAPE, J. L. FINNEY, AND L. V. WOODCOCK, *J. Chem. Phys.* **75**, 2366 (1981).
7. J. L. MEIJERING, *Philips Res. Rep.* **8**, 270 (1953); D. WEAIRE AND J. P. KERMODE, *Philos. Mag. B* **48**, 245 (1983); D. A. ABOAB, Metall. **13**, 43 (1980); W. BROSTOW, *Science of Materials* (Wiley-Interscience, New York/London, 1979).
8. D. PHYNSBURGER, *Geogr. Anal.* **5**, 133 (1970); D. F. WATSON AND F. G. SMITH, *Comput. Geosci.* **1**, 109 (1975); F. M. RICHARDS, *J. Mol. Biol.* **82**, 1 (1974).
9. P. J. GREEN AND R. SIBSON, *Comput. J.* **21**, 168 (1978).
10. A. BOWYER, *Comput. J.* **24**, 162 (1981).
11. D. F. WATSON, *Comput. J.* **24**, 167 (1981).
12. J. L. FINNEY, *J. Comput. Phys.* **32**, 137 (1979).
13. W. BROSTOW, J.-P. DUSSAULT AND B. L. FOX, *J. Comput. Phys.* **29**, 81 (1978).
14. N. N. MEDVEDEV AND YU. I. NABERUKHIN, *Zh. Strukt. Khim.* **26**, 65 (1985).

15. N. N. MEDVEDEV AND YU. I. NABERUKHIN, "Symposium on the Structure of Liquids and Solutions," Collected Abstracts, Veszprem, Hungary, August 27–30. p. 97.
16. M. TANEMURA, T. OGAWA AND N. OGITA, *J. Comput. Phys.* **51**, 191 (1983).
17. A. D. ALEXANDROV, *Convex polyhedra* (Moscow, 1950). [Russian]

RECEIVED: September 26, 1984; REVISED: September 20, 1985

N. N. MEDVEDEV
Institute of Chemical Kinetics and Combustion, U.S.S.R. Academy of Sciences, Siberian Branch, Novosibirsk 630090, U.S.S.R