JOURNAL OF COMPUTATIONAL PHYSICS 29, 81–92 (1978)

Construction of Voronoi Polyhedra*

WITOLD BROSTOW† AND JEAN-PIERRE DUSSAULT

Department of Chemistry, University of Montreal, Montreal, Quebec, Canada H3C 3V1

AND

BENNETT L. FOX

Department of Computer Science and Operations Research, University of Montreal, Montreal, Quebec, Canada H3C 3J7

Given a configuration of points, a procedure for constructing the corresponding Voronoi diagram is given. The procedure is exact for molecules in the bulk. Polyhedra of surface molecules can be either eliminated or included using a periodic boundary condition. The construction is of interest in astronomy, biology, chemistry, materials science, as well as in physics (with points representing atoms, molecules, ions, etc.). The present method is more efficient than other procedures described in the literature.

INTRODUCTION

Consider a physical system consisting of a number of distinct entities. Typically, the entities are molecules, but they can also be ions, atoms, polymer segments, radicals, and so on. We know that equilibrium and other properties of the system depend on spatial distribution of the entities, and the question is how to represent this distribution conveniently? A method known for a long time, but which has recently been strongly increasing in popularity, consists in dividing the three-dimensional space between entities. Each entity "owns" a certain portion of the space in the shape of a polyhedron. Thus, each physical entity is principally characterized by the location of its geometrical center (to be shortly called center throughout this paper) and by the size and shape of the surrounding polyhedron.

The polyhedra in question were first defined by mathematicians [1, 2] and then "rediscovered" several times by physicists [3, 4]. Consequently, they are variously known as Dirichlet [1] regions, Voronoi [2] polyhedra, Wigner-Seitz [3] cells, or domains [4]. The name Wigner–Seitz cells seems to be popular among solid state physicists, who use this approach extensively [5]. Some time ago [6] we decided to

* This paper corresponds to an invited lecture given at the VIIth International Conference on Statistical Physics in Oaxtepec, Morelos, Mexico, 3–6 January 1978.

† New address: Center for Advanced Studies, National Polytechnic Institute, A.P. 14–740, Mexico 14, D.F., Mexico.

0021-9991/78/0291-0081$2.00/0

use the name Voronoi polyhedra; we continue with the same terminology, apparently used by most mathematicians [7].

Allotting spaces to molecules is clearly of interest to molecular physicists, biochemists, materials scientists, and physical chemists. Construction of Voronoi polyhedra is also of interst to astrophysicists, in connection with fragmentation of celestial bodies [8]. Our own interest in the polyhedra is related to a theory of liquids and amorphous solids based on the theory of information. [6, 9]

Although procedures for the construction of the polyhedra were developed by several authors [8, 10–14], they were not satisfactory for our purposes. The procedures of Kiang [8], Finney [10], Mackay [11], and Richards [12] seemed not efficient enough; there were not enough details in the paper of Rahman [13]; and Shamos and Hoey [14a] treated two dimensions only. In these circumstances, we developed a new method, which is described in the present paper.

In Section 2 we define the basic notions. In Section 3 we show how to construct the Voronoi diagram. Once proposed, our method is compared in Section 4 with earlier approaches. Section 5 contains some algebraic details pertinent for the users of our method. In the final section we discuss briefly the present and potential applications of the method; there is little point in the discussing these applications in the present paper in any detail, since the extensive work already done, by physicists in particular, proves clearly the usefulness of Voronoi polyhedra.

2. Basic Definitions

Consider a set of centers P_1, P_2,..., P_n in L-dimensional Euclidean space E. The *Voronoi polyhedron* V_i around a given center P_i, is the set of points in E closer to P_i than to any P_j : more formally,

$$V_i = \{x \in E : d(x, P_i) \leqslant d(x, P_j), j = 1, 2,..., n\}, \tag{1}$$

where d denotes distance. Thus, the polyhedra are intersections of half-spaces; they are convex but not necessarily bounded. The polyhedra partition E in a unique way. The set of Voronoi polyhedra corresponding to a given configuration of centers is called the *Voronoi diagram.*

For obvious physical reasons we consider mainly the case $L = 3$. Given a center P_i and its neighbor P_j, the line P_iP_j is cut perpendicularly at its midpoint y_{ij} by the plane h_{ij}. We call H_{ij} the half-space generated by h_{ij} that consists of the subset of E on the same side of h_{ij} as P_i ; that is,

$$V_i = \bigcap_j H_{ij}. \tag{2}$$

V_i is bounded by faces, with each face f_{ij} belonging to a distinct plane h_{ij}. Each face is characterized by listing its vertices and edges in cyclic order.

It is pertinent to distinguish various possible kinds of neighbors of P_i. We define the following classes of neighbors:

(i) *direct* neighbors: if y_{ij} belongs to V_i, then P_j is a direct neighbor (these are the full neighbors of Meijering [15]);

(ii) *indirect* neighbors: if a subset of h_{ij} is a face of V_i but y_{ij} does not belong to v_i, then P_j is an indirect neighbor; or P_j is an indirect neighbor if $f_{ij} \cap y_{ij} = \varnothing$;

(iii) *degenerate* neighbors: if the intersection of h_{ij} and V_i is just a vertex or an edge, then P_j is a degenerate neighbor;

(iv) *quasi-direct* neighbors: if P_j is a direct neighbor or if P_j would be a direct neighbor in the absence of all indirect neighbors, then P_j is a quasi-direct neighbor of P_i.

Clearly, all direct neighbors are also quasi-direct. Examples of neighbors representing classes defined above are shown in Fig. 1 in *two* dimensions; extension to three dimensions is obvious. The quasi-direct neighbors generate a *direct polyhedron* D_i in the same way that direct and indirect neighbors generate V_i. Clearly

$$V_i \subset D_i, \qquad \text{for each } i. \tag{3}$$

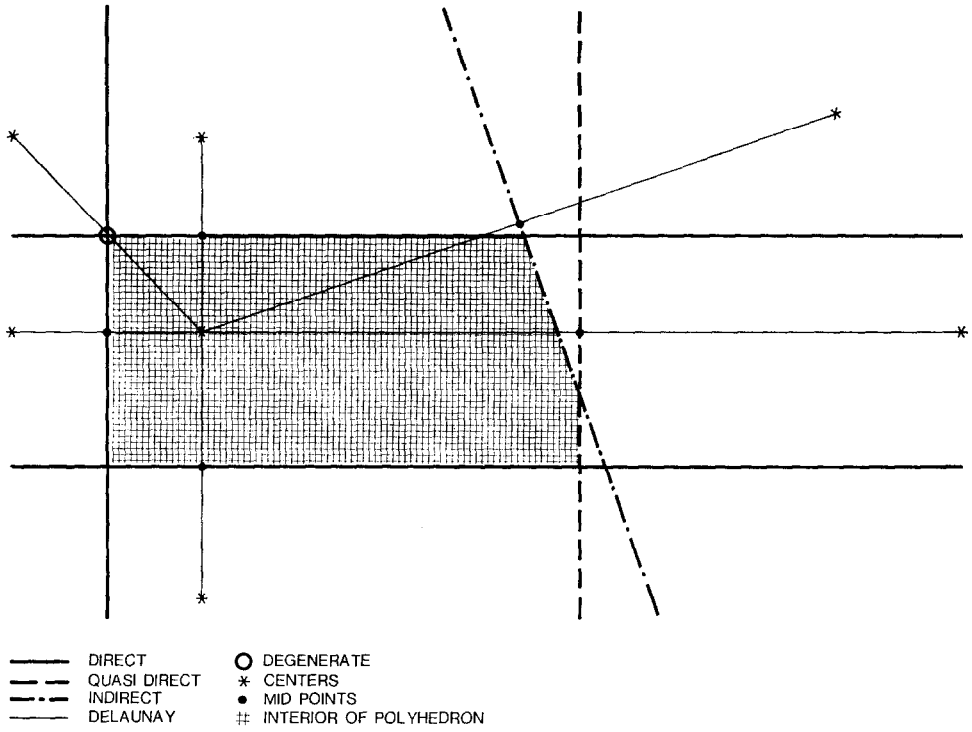

FIG. 1. Example of various classes of neighbors of a Voronoi polygon.

We define the *geometric coordination number* f_i as the number of nondegenerate direct or indirect neighbors of P_i. As stressed by one of us [16], the number f_i has to be distinguished from the *structural coordination number* z_i. The latter is defined in terms of the binary radial distribution function $g(R)$, that is, in terms of the probability of finding another molecule at a distance R from a given molecule. The average value of f for random models seems to be ≈ 15 [11], but Voronoi polyhedra with $f = 20$ have been constructed [17], and arbitrarily high values of f are possible. By contrast, if only appropriate integration limits are used in the evaluation of z, the highest value of z is 12 in crystals and 11 in liquids [16].

We embed P_i in a large "boundary" cube C_i, where the closest face of C_i to P_i is relatively far away from the farthest quasi-direct neighbor of P_i. (Except for dealing with surface phenomena, it would be convenient to use a single cube C for the whole system.) If a face of C_i cuts D_i, we call D_i *virtually unbounded.* A virtually unbounded polyhedron may be either bounded or unbounded. The appearance of virtually unbounded polyhedra is intimately related to surface effects. Therefore, detection of the presence of these polyhedra is important. Our method pinpoints them, and the corresponding V_i are not constructed. We believe that this mathematical minus is actually a physical plus in relating the Voronoi diagram to real physical systems.

It is worth noting that C_i can be very large. Extremely large D_i, while mathematically possible, simply do not occur in Voronoi diagrams representing the interior of systems of interest in astronomy, biology, chemistry, or physics. For values of D_i of realistic relative dimensions with respect to C_i, the size of C_i has little if any effect on the work of constructing the Voronoi diagram.

We also have the option of using a *periodic boundary condition.* This can be done as follows. Put the given configuration of points in a box. Now put this box in the center of a stack of congruent boxes, each containing the same configuration of points as the original. For each point in the center box, and only these, Voronoi polyhedra are constructed taking account of all points in all boxes. By definition of the periodic boundary condition, the resulting collection of Voronoi polyhedra constitutes the Voronoi diagram. When this option is taken, all D_i in the original box are automatically bounded, and a boundary cube C_i is not needed.

3. Construction of the Polyhedra

There seems to be a growing consensus among users of geometry that many geometrical problems have to be revisited and considered from the point of view of computing facilities now available. This is due to the fact that geometry developed and flourished in a period when fast algorithms were of little if any importance. This is also our experience with the Voronoi polyhedra. In spite of many applications of the polyhedra by physicists and biologists, and despite the work of mathematicians, it appears that certain properties of the polyhedra were overlooked. In the present section we discuss these properties of the polyhedra which we found useful in the quest for an efficient procedure of constructing them.

We begin with a simple observation: On the average, direct polyhedra D have simpler shapes, that is, fewer faces, than Voronoi polyhedra V. This and relation (3) motivate our key idea: begin by constructing the polyhedron D, and only then proceed towards V. Given a center P_i and its bounded direct polyhedron D_i, we can circumscribe a sphere of diameter d_i around D_i (even when not explicitly stated, we use squared distances to avoid calculating square roots). Then any point more than d_i away from P_i cannot be a neighbor; this simple criterion eliminates most of the candidates for indirect neighbors. Those which are not eliminated serve to obtain V_i from D_i; vertices, edges, and sometimes entire faces of D_i are cut off by planes generated by indirect neighbors. Note that by definition d_i is twice the distance from P_i to the farthest vertex of D_i.

In many if not most geometrical problems one begins with locating vertices, then joins appropriate pairs of vertices by edges, and finally constructs faces or planes. Studies of random models [10, 11, 13] indicate that the average number of edges per face is ≈ 5. In constructing both D_i and V_i we chose to find first the respective faces, from these to find the edges, and then finally to find the vertices. A careful analysis of the problem (cf. Section 4 and the end of Section 5) indicates that any other order would be much less efficient. A computer procedure called FACFIN for finding

TABLE I

FACFIN

1. $i \leftarrow 1$
2. Sort the squared distances $d(P_i, y_{ij})$ in increasing order. Let the indices of the sorted list be [1], [2],..., $[n-1]$, in that order.
3. $F_i \leftarrow \{[1]\}$
4. $j \leftarrow 2$
5. If $y_{[j]}$ is in H_{im} for all m in F_i, set $F_i \leftarrow F_i \cup \{[j]\}$.
6. If $j = n - 1$, go to 7. Else, $j \leftarrow j + 1$ and go to 5.
7. If $i = n$, STOP. Else, $i \leftarrow i + 1$ and go to 2.

Comment: A preliminary test for unbounded polyhedra can be made at step 5: If $j \in F_i$ and $j \leftrightarrow C_i$, set flag.

faces of the direct polyhedra is shown in Table I. We devised this procedure with the objective of satisfying:

LEMMA 1. FACFIN *terminates with F_i containing exactly the indices of the faces of D_i.*

Proof. We use induction. Clearly [1] is the index of a face. Assume that the algorithm is correct up to and including $j - 1$. If $y_{[j]}$ is in H_{im} for all m in F_i, then $y_{[j]}$ is in D_i because it cannot be cut off by a plane generated by a quasidirect neighbor, since we consider the planes in the order specified in step 2. On the other hand, if $y_{[j]}$ is not in H_{im} for some m in F_i, it is not a quasi-direct neighbor by definition.

As already noted, one checks for intersections of D_i with the boundary cube C_i. If there are nonempty such intersections, D_i is eliminated as virtually unbounded. If D_i remains, d_i is computed. A procedure which constructs and checks direct polyhedra called DIRPOL is shown in Table II. Given D_i, one finds all indirect neighbors and proceeds toward V_i. A procedure which produces V_i called VORPOL is shown in Table III. Each candidate face for V_i is considered just once. The procedure is

TABLE II

DIRPOL

1. $i \leftarrow 1$
2. With the set of edges associated with f_i. denoted by A_i. and analogously for the vertices S_{ij}, compute: $A_i = \bigcup_j A_{ij}$, and $S_i = \bigcup_j S_{ij}$.
3. Given faces c_{it} of C_i, sort the distances $d(P_i, c_{it})$ in increasing order. Sort outer half-spaces C_{it}, which contain each the respective c_{it} but not P_i, in the same order.
4. $t \leftarrow 1$
5. If $C_{it} \cap S_i \neq \phi$, set flag. (Comment: D_i is virtually unbounded).
6. If $t = 6$, go to 7; else $t \leftarrow t + 1$ and go to 5.
7. Compute d_i.
8. If $i = n$, STOP. Else, $i \leftarrow i + 1$ and go to 2.

TABLE III

VORPOL

1. $i \leftarrow 1$
2. Number the faces of D_i as 1, 2,..., f_d. For every center within d_i of P_i, construct its h_{ij}. Number the new candidate faces $f_d + 1$,..., r, say.
3. $j \leftarrow f_d + 1$, $V_i \leftarrow D_i$.

   Comment: F_i, A_i, A_{ij}, S_i, and S_{ij} are initialized to the values produced by FACFIN and DIRPOL except that F_i is the set of faces of D_i, not just their indices.
4. If h_{ij} cuts off one or more vertices of S_i:
   (a) $V_i \leftarrow V_i \cap H_{ij}$. (Comment: if f_{ik} remains but y_{ik} is cut off, then f_{ik} becomes indirect.)
   (b) $F_i \leftarrow f_{ij} \cup (F_i \cap H_{ij})$
   (c) $A_i \leftarrow A_{ij} \cup (A_i \cap H_{ij})$
   (d) $S_i \leftarrow S_{ij} \cup (S_i \cap H_{ij})$
5. If $j = f_d + r$, go to 6; else, $j \leftarrow j + 1$ and go to 4.
6. If $i = n$, STOP; else, $i \leftarrow i + 1$ and go to 2.

initiated with a list of candidate faces and this list does not grow. Hence, the procedure terminates when the initial list is exhausted. At this point, the vertices, edges, and faces of V_i are exactly those in the respective lists of candidate vertices, edges and faces. Using DIRPOL and VORPOL, it is easy to prove:

PROPOSITION 1. *If D_i is virtually unbounded, a flag is set. Otherwise, V_i is constructed exactly.*

Since efficiency as well as exactness is sought, we do not construct each polyhedron in the Voronoi diagram from scratch. Results from the polyhedra already constructed are used, provided that there is sufficient computer memory to access this information quickly. Of course, if there is not enough high-speed memory available in a given computer, each face can be computed twice, once for each polyhedron involved. Observe that if P_j is a neighbor of P_i, then P_i is a neighbor of P_j. This also holds for any qualification of a neighbor such as direct, indirect or degenerate. Perhaps less obvious, we have:

LEMMA 2. *For any face f_{ij}, each vertex of f_{ij} is a vertex of both polyhedra V_i and V_j. Thus, $f_{ij} = f_{ji}$ and its edges and vertices have to be calculated just once.*

Proof. Suppose that x is a vertex of f_{ij} but not of f_{ji}. On the plane h_{ij}, draw a circle of radius ϵ around x. For sufficiently small positive ϵ, some points in this circle are in V_i and some are not; simultaneously, either every point in the circle is in V_j or none are. However, every point in h_{ij} (particularly in the circle) is equidistant from P_i and P_j. Hence, all such points are in both V_i and V_j, or in neither. Combining our remarks, we have a contradiction.

4. RELATIVE EFFICIENCY

The overall work of step 2 of FACFIN is $O(n^2 \log n)$, using HEAPSORT (cf., e.g., Knuth [18 pp. 145–149]). The work for step 5 is at worst $O(n[|F_1| + |F_2| + \ldots + |F_n|])$. As most planes will probably be eliminated well before all m in F_i are checked, the implicit proportionality constant should be small.

Suppose that each V_i has at most k faces and that the number of centers less than d_i away from P_i is $O(k)$ or less. Once D_i is found, the work to construct V_i is $O(k^3)$ or less. Thus, except for the face-finder routine (FACFIN), the overall work is at most $O(k^3 n)$. As noted in Section 2, large values of f, even if possible in principle, rarely occur in practice. Therefore, k is generally significantly less than n. If k does not grow with n, for large n the dominant term for our procedure is $O(n^2 \log n)$.

In contrast to our method, Finney's algorithm [10] is $O(n^4)$. He uses the fact that each vertex of the diagram is equidistant from four (noncoplanar) centers. Thus, he solves $\binom{n}{4}$ systems of equations. At first sight, each system consists of four simultaneous quadratic equations. However, these can be reduced to three simultaneous linear equations. Finney indicates a heuristic modification of his method where candidate neighbors more than a certain distance away from a given center are not considered. This cuts down the work significantly, but it seems difficult to choose an appropriate distance a priori. Our boundary cube is also rather arbitrary. But, as already mentioned in Section 2, the speed of our algorithm is normally not sensitive

to the size of the cube. On the other hand, it appears that the speed of Finney's algorithm is sensitive to the size of his boundary sphere.

As $O(n^4)$ is slow for n in the hundreds and hardly worth using for n in the thousands, Finney's introduction of a boundary sphere may be a practical necessity. It appears that any *a priori* choice of sphere radius could require $O(n^4)$ time in the worst case—where all centers are in a given sphere. On the other hand, if the spheres are chosen *ex post*, so that only the k, say, closest centers are inside each sphere, then $O(n^2 \log n)$ time is required to construct the (n)sphere radii and $O(k^3 n)$ time to construct the (n)polyhedra in the Voronoi diagram. These "best" bounds for Finney's procedure are comparable to the time complexity of our program, which searches for neighbors in a general way. We could of course limit our search in a way similar to Finney's, and thus make our program run still faster than it does now. A "self-checking" procedure for choosing sphere radii can, perhaps, be justified on ad hoc physical grounds, but neither indications on how to do self-checking nor any criterion whatsoever for limiting the search for neighbors can be found in the published literature.

Mackay [11] uses essentially the same method as Finney. He chooses boundary sphere radii *a priori* [11, p. 226, step 1]. No criterion for prescribing the radii is given. If only crystal lattices are considered, such a criterion can be easily formulated. Mackay insists, however, and we agree with him entirely, that Voronoi polyhedra ought to be used to study both crystalline and noncrystalline arrangements of atoms.

Richards [12] finds the Voronoi diagram with a method similar to ours, *except* that he does not introduce the direct polyhedron. Instead, he uses a "large" boundary tetrahedron to eliminate candidate neighbors [12, p. 5, paragraph (c)]. He does not indicate a method for choosing the size of this tetrahedron. If this tetrahedron were truly large, his algorithm would be slow.

An approximate method consists in putting a grid over the space, associating a small cube to each grid point, and associating each grid point with the closest center. This was done, for instance, by Kiang [8] who had 100 randomly distributed centers in a cube containing 20^3 lattice points. Such calculations can be used to obtain qualitative information only; Kiang was not interested in faces, edges, etc., but in the distribution of the volumes of the polyhedra in the Voronoi diagram. Any attempt to increase the accuracy of the method involves an increase in the number of lattice points, and therefore a dramatic loss in efficiency.

A lower bound, to within a constant factor, on the work required by any exact algorithm is the sum of numbers of faces, edges, and vertices in the diagram. Call this sum $s(n)$. For planar problems, Shamos [14] shows that $s(n) = O(n)$ in the worst case by relating the Voronoi diagram to a planar graph. The worst case for three dimensions is not known. For two dimensions, Shamos shows that $O(n \log n)$ is a *tight* lower bound on the work to construct the diagram. He gives a clever algorithm for constructing planar Voronoi diagrams. The three dimensional analog would be to partition E into cubes, construct the Voronoi polyhedra corresponding only to the centers in the respective cubes, and then recursively merge these polyhedra to obtain a correct overall diagram. If merging two diagrams takes time $O(u(n))$, then the overall time required is $O(u(n) \log n)$. Clearly $u(n)) \geqslant O(s(n))$ in the worst case.

A three-dimensional merge procedure with $u(n) < O(n^2)$ would lead to an algorithm asymptotically faster than ours. Such a procedure has not been found. If it exists, it is surely complex and involves much overhead. It is not clear *a priori* that it would be efficient for practical values of n.

As mentioned in the Introduction, Rahman [13] also generated Voronoi polyhedra to study structures of liquid phases. His configuration of centers was generated by the method of molecular dynamics. It is impossible to judge the efficiency of his method, since no details about how he constructed the polyhedra are given.

5. Algebraic Details

For completeness, we indicate how to interpret our geometrical constructions in terms of standard solid analytic geometry. A plane in 3-space has the form

$$h(x) = b, \tag{4}$$

where

$$h(x) = a_1x_1 + a_2x_2 + a_3x_3 . \tag{5}$$

Points c and d are on the same side of this plane if and only if $h(c)$ and $h(d) \leqslant b$ or $h(c)$ and $h(d) \geqslant b$. Obviously, point e is on the plane if and only if $h(e) = b$.

Consider two (nonparallel) planes:

$$a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 = b_i , \tag{5'}$$

$$a_{j1}x_1 + a_{j2}x_2 + a_{j3}x_3 = b_j . \tag{5''}$$

Define A, B, C by the determinant

$$\begin{vmatrix} \alpha & \beta & \gamma \\ a_{i1} & a_{i2} & a_{i3} \\ a_{j1} & a_{j2} & a_{j3} \end{vmatrix} = A\alpha + B\beta + C\gamma. \tag{6}$$

Suppose that the point (k_1 , k_2 , k_3) is on the line formed by the intersection of these planes. Then

$$B(x_1 - k_1) = A(x_2 - k_2), \tag{7'}$$

$$C(x_1 - k_1) = A(x_3 - k_3) \tag{7''}$$

determine this line. Cyclic permutation of parameters A, B, and C, along with the indices 1, 2, and 3 can be made to avoid zero denominators in subsequent equations. To determine a suitable point (k_1 , k_2 , k_3) easily, note that at least one of the planes

$$x_1 = 0, \qquad x_2 = 0, \qquad x_3 = 0, \tag{8}$$

must intersect both the given planes. For example, suppose $x_3 = 0$ works. Then $k_3 = 0$ and (k_1, k_2) is the solution to the system

$$a_{i1}x_1 + a_{i2}x_2 = b_i, \tag{9'}$$

$$a_{j1}x_1 + a_{j2}x_2 = b_j, \tag{9''}$$

which is easily solved by the usual method of determinants, with the value of C calculated only once.

For a given face i, varying j gives all the candidate edges of this face. Use primes to denote quantities associated with the second candidate edge, and put $\alpha = A/B$ and $\alpha' = A'/B'$. If $\alpha = \alpha'$ (for every cyclic permutation of A, B, C along with the indices 1, 2, and 3), the two edges are parallel; otherwise, we have

$$x_1 = \alpha(x_2 - k_2) + k_1, \tag{10}$$

$$x_2 = (\alpha k_2 - \alpha' k_2' + k_1 - k_1')/(\alpha - \alpha'), \tag{10'}$$

$$x_3 = C(x_1 - k_1)/A + k_3, \tag{10''}$$

from which we get (in order) x_2, x_1, x_3. The pairwise intersections of these candidate edges give the candidate vertices. The real edges and vertices are selected from these candidates using cutoff criteria discussed in Section 3. If we had tried to find the candidate vertices on a given face directly, we would have had to solve $\binom{f-1}{2}$ systems of three simultaneous equations. Finding the edges first is also an $O(\binom{f}{2})$ scheme, but the proportionality constant is smaller.

6. Some Concluding Remarks

Throughout this paper we have talked about individual physical entities represented by the Voronoi polyhedra. We know that most molecules are polyatomic, but this by no means prevents the use of the present method. Molecules of chemical compounds may be represented by graphs, and such graphs are directly useful for predicting thermodynamic properties of liquid phases [19, 20]. Atoms (or groups of atoms, or polymeric segments) may be represented by graph points, and these points may serve as centers of the Voronoi polyhedra. Thus, it is only necessary to indicate connectedness between the centers, e.g., in a way discussed more in detail in an earlier paper [6]. Physical entities of *different* size can be treated also. A reliable even if approximate procedure, aimed at dealing with atoms in protein molecules, was proposed by Richards [12]. The dividing planes for covalent bonds are drawn perpendicular to the interatomic vector P_iP_j and positioned so as to divide P_iP_j in proportion to the appropriate covalent radii. Clearly each such plane is an analog of the plane h_{ij} as defined in Section 2, except that in general it does not include the midpoint y_{ij}. For a pair of atoms which are not bonded Richards uses a similar procedure, but this time the van der Waals radii of atoms determine the position of the dividing plane

on the P_iP_j line. Richards admits that in his method certain regions of space may not be assigned to any atom. Criticizing this, Mackay [21] suggests an alternative (Voronoi-like) diagram.

Solid state physicists use the Voronoi polyhedra mostly to describe crystalline materials [5], but the polyhedra are clearly even more useful for dealing with irregular structures. Bernal [22] developed a dense random packing (DRP) model of liquid phases. The model was subsequently translated [10] into the language of the Voronoi diagrams. It was applied also to amorphous solids [23–26]. The Voronoi polyhedra used to describe amorphous semiconductors [23–25] are fairly complex and have relatively large values of f [24], while those used to represent glassy metals are somewhat simpler [26]. Reviewing various approaches to amorphous structures, Takayama [26] stresses attractiveness of the DRP model.

As already mentioned, our main objective is to pursue the information theory approach to liquid and amorphous solid phases. It is possible to obtain expressions for equilibrium properties such as pressure or configurational energy (cf., e.g., [6]), but they all contain a geometric parameter μ. This parameter decreases with the structural coordination number z [9], but the general $\mu \leftrightarrow z$ correspondence is unknown. Thus, given the procedure for constructing the Voronoi diagram, our first task consisted in elucidating the $\mu \leftrightarrow z$ dependence [27]. The next stage is that of studying relations between z and the number of faces f. For crystalline materials, the configurations of centers are simple mixtures of regular lattices. For liquids and amorphous solids, the configurations are generated by a Monte Carlo procedure according to a probability law that seems physically reasonable in terms of intermolecular potential $u(R)$.

The computer program VORDIG which constructs the Voronoi diagram is available from us on request.

Acknowledgments

Sincere thanks are due to Dr. J. S. Sochanski, Department of Physics, The University of Quebec, Trois-Rivières, for his detailed comments on the manuscript. Discussions with Dr. M. Grmela, Center for Research in Mathematics, University of Montreal, are appreciated also. A referee of this paper has drawn our attention to studies on characterization of minerals [11] and on interpretations of structures of proteins [12] in terms of Voronoi polyhedra. This work forms a part of a program supported financially by the National Research Council of Canada, Ottawa.

References

1. G. L. Dirichlet, *Z. Reine Angew. Math.* **40** (1850), 216.
2. G. F. Voronoi, *Z. Reine Angew. Math.* **134** (1908), 198.
3. E. P. Wigner and F. Seitz, *Phys. Rev.* **43** (1933), 804.
4. F. C. Frank and J. S. Kasper, *Acta Crystallogr.* **11** (1958), 184.
5. J. M. Ziman, "Principles of the Theory of Solids," 2nd ed., Cambridge Univ. Press, London/New York, 1972.

6. W. BROSTOW AND Y. SICOTTE, *J. Statist. Phys.* **9** (1973), 339.
7. C. A. ROGERS, "Packing and Covering," Cambridge Univ. Press, London/New York, 1964.
8. T. KIANG, *Z. Astrophys.* **64** (1966), 433.
9. W. BROSTOW AND Y. SICOTTE, *Physica A* **80** (1975), 513.
10. J. L. FINNEY, *Proc. Roy. Soc. Ser. A* **319** (1970), 479.
11. A. L. MACKAY, *J. Microscopy* **95** (1972), 217.
12. F. M. RICHARDS, *J. Mol. Biol.* **82** (1974), 1.
13. A. RAHMAN, *J. Chem. Phys.* **45** (1966), 2585.
14. (a) M. I. SHAMOS AND D. HOEY, Proc. 16th Ann. IEEE Symposium on Foundations of Algorithms, Berkeley, Calif., p. 151, 1975; (b) M. I. SHAMOS, Proc. 7th ACM Symposium on Theory of Computing, 1975.
15. J. L. MEIJERING, *Philips Res. Rep.* **8** (1953), 270.
16. W. BROSTOW, *Chem. Phys. Lett.* **49** (1977), 285.
17. F. W. SMITH, *Can. J. Phys.* **43** (1965), 2052.
18. D. E. KNUTH, "The Art of Computer Programming," Vol. 3, Addison–Wesley, Reading, Mass., 1973.
19. W. BROSTOW AND A. SCHINZEL, *J. Statist. Phys.* **4** (1972), 103.
20. W. BROSTOW, *Phys. Chem. Liquids* **3** (1972), 91; W. BROSTOW AND B. C.-Y. LU, *Phys. Chem. Liquids* **4** (1974), 83.
21. A. L. MACKAY, Birbeck College, University of London, preprint.
22. J. D. BERNAL, *Proc. Roy. Soc. Ser. A* **280** (1964), 299.
23. M. V. COLEMAN AND D. J. D. THOMAS, *Phys. Status Solidi* **24** (1967), K111; R. GRIGOROVICI AND R. MANAILA, *Thin Solid Films* **1** (1968), 343; R. GRIGOROVICI, *J. Non-Cryst. Solids* **1** (1969), 303.
24. R. GRIGOROVICI AND R. MANAILA, *J. Non-Cryst. Solids* **1** (1969), 371.
25. D. TURNBULL AND D. E. POLK, *J. Non-Cryst. Solids* **8–10** (1972), 19.
26. S. TAKAYAMA, *J. Mater. Sci.* **11** (1976), 164.
27. W. BROSTOW, J.-P. DUSSAULT, B. L. FOX, AND J. S. SOCHANSKI, in preparation.