Analytic Shape Computation of Macromolecules

I. Molecular Area and Volume Through Alpha Shape[1]

October 21, 1995

Jie Liang[2], Herbert Edelsbrunner[3], Ping Fu[4], Pamidighantam V. Sudharkar[5] and Shankar Subramaniam[5]

Abstract. Shape and geometry of macromolecules such as proteins and nucleic acids play an important role in their function. Alpha shape theory provides an analytically exact method for computing metric properties of macromolecules. The method uses the duality between the alpha complex and the weighted Voronoi decomposition of a molecule. We describe in this paper the intuitive ideas and concepts behind the alpha shape theory and explain how the method can be applied for computing areas and volumes of proteins. We demonstrate the power of the method through numerous applications and comparisons with previous methods. We also discuss several difficulties commonly encountered in molecular metric computations and outline methods to overcome these problems.

[1]Work by the first author is supported by an NSF post-doctoral fellowship, grant ASC 94-04900. The second and third authors acknowledge support by NSF, grant ASC 92-00301, and by ONR, grant N00014-95-1-0692. ADD FUNDING INFORMATION FOR SUDHARKAR AND SHANKAR.

[2]Department of Computer Science and National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

[3]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

[4]National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Champaign, Illinois 61820, USA.

[5]Beckman Institute for Advanced Science and Technology, Department of Physiology and Biophysics and National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

1 Introduction

Macromolecules such as protein and DNA have complex structure. The specific spatial configurations of molecules are important for proteins and nucleic acids to carry out their functions. Surface area and molecular volume are two geometric quantities which determine various properties of these complex molecules. They play a role in protein folding (Chotia, 1975), conformational stability (Kauzmann, 1959), solubility (Langmuir, 1925; Sharp *et al.*, 1990; Sharp *et al.*, 1991), crystal packing (Richards, 1977; Rashin *et al.*, 1986), molecular recognition and docking (Connolly, 1986), and in enzyme catalysis (— Do we have a reference? —).Recently, energy refinement ←??? methods have been developed which include the surface area (Freyberg *et al.*, 1993) or excluded volume (Kundrot *et al.*, 1991) for calculating solvation energies. With the advances in X-ray crystallography and NMR techniques, molecular structures have been determined for many proteins and nucleic acids in atomic detail. Current and future structure results provide rich material for atomic level molecular modeling and analysis.

Lee and Richards introduced the models of solvent accessible surface (SA) and molecular surface (MS) for proteins. In the important special case of a point size solvent the two surface are the same and referred to as the van der Waals surface (VS) model of the molecule, see figure 2.1. These models provide a means to unambiguously define geometric properties of molecules and they motivate the development of algorithms and software for computing such properties. Computation of the surface area and volume of molecules have been the focus of research for some time, and algorithms of both analytical and numerical nature are available. In general, however, it is still difficult to rigorously and precisely describe and manipulate various aspects of the shape of a macromolecule.

In parallel, the field of computational geometry has experienced rapid progress since its establishment in the late 70's (Preparata & Shamos, 1985; Edelsbrunner, 1987). Some developments have direct implications for molecular biology. Among these, the alpha shape theory (Edelsbrunner *et al.*, 1983) provides a quantitative method to accurately describe and compute shapes at multilevels of details in 3-dimensional space, $\mathbb{R}^3$. It uses Delaunay complexes and their filtrations to describe the topological structure of a molecule. The mathematical relation between alpha shapes and the above mentioned sphere models of a molecule has been firmly established (Edelsbrunner, 1993). With the availability of 3-dimensional alpha shape software (Edelsbrunner & Mücke, 1994; Edelsbrunner *et al.*, 1995; Varshney *et al.*, 1994), it is now possible to accurately and efficiently compute a variety of geometric aspects of a molecules.

Computational methods to determine the area and volume based on VS, SA and MS models can be broadly divided into two categories: approximation methods (Shrake & Rupley, 1973; Richmond & Richards, 1978; Alden & Kim, 1979; Wodak & Janin, 1980; Muller, 1983; Pavlov & Fedorov, 1983; Wang & Levinthal, 1991; Grand & Jr., 1993) and analytic methods (Connolly, 1983; Connolly, 1985a; Connolly, 1985b; Richmond, 1984; Gibson & Scheraga, 1987; Gibson & Scheraga, 1988; Kundrot *et al.*, 1991; Perrot *et al.*, 1992). (— The following three references have not been categorized ... Jie can you add them to the relevant lists?: (Richards, 1985; Pascual-Ahuir & Silla, 1990; Connolly, 1993). —)Most approximation methods involve certain ←??? discretizations, such as polyhedral or triangular decompositions or the representation of surface with a large number of dots. Among these methods, Richards' volume program (Richards, 1974) is widely used and is distributed in the vadar package (Wishart *et al.*, 1994). (— Why is (Richards, 1974) not included in the reference list of the approximation methods? —)It uses either bisector planes or planes based on van der Waals radii ←??? to divide the space of the internal atoms into polyhedra. Atoms on the surface are divided similarly with the help of fictitiously placed solvent atoms. The area and volume of the molecule are then calculated through these polyhedra. The gepol program, distributed as part of the arvolmol package (Pacios, 1993), is one of the two programs known to the authors of this paper that compute the MS area and volume (Pascual-Ahuir & Silla, 1990; Silla *et al.*, 1991; Pascual-Ahhuir *et al.*, 1994). It fills the solvent inaccessible space between atoms with spheres, and then triangulates these spheres. The triangles facing outside are selected and used in turn to compute area and volume.

Representing atoms by spherical balls provides the opportunity for analytic treatment of surface area and volume of molecules. Among the analytic methods, the anarea program is distributed in the vadar package and computes the area of the SA model (Richmond, 1984). The msdot program by Connolly, which is distributed as

part of ARVOMOL, is the second program computing the MS area and volume (Connolly, 1983; Connolly, 1985a; Pacios, 1993). Closed-form analytical expressions for area and volume computation were also derived by Gibson and Sheraga (Gibson & Scheraga, 1987); they eliminate overlap of 5 or more atoms using an observation by Kratky (Kratky, 1981). In this paper, we compare our results with those obtained using the programs ANAREA, GEPOL, MSDOT, and VOLUME.

The paper is organized as follow. We first introduce the ideas and concepts behind the alpha shape theory as a fundamental approach to address geometric and topological questions about molecules. We then describe the alpha shape based algorithm for computing surface area and volume. Finally, we compare our computational results with result obtained with other software.

2 Theory and Algorithms

A full account of the alpha shape theory and the resulting algorithms for computing geometric properties of molecules can be found in the computer science oriented literature (Edelsbrunner *et al.*, 1983; Edelsbrunner & Mücke, 1994; Edelsbrunner, 1993; Edelsbrunner *et al.*, 1995). This paper makes an attempt to intuitively describe the concepts and ideas behind the alpha shape theory. It also explains the algorithms that compute area and volume of a molecule based on its alpha complex.

2.1 Models of molecules and Voronoi diagram

Three definitions of surface are widely used for molecular modeling. In each case, atoms are treated as possibly intersecting spherical balls (Lee & Richards, 1971; Richards, 1977; Connolly, 1983). The *van der Waals surface* (VS) is the surface of what is covered by the atoms each represented by a spherical ball with van der Waals radius, see figure 2.1, A. The *solvent accessible surface* (SA) is generated by the center of the solvent (modeled as a rigid sphere) when rolling about the van der Waals surface of the molecule, see figure 2.1, B. The *molecular surface* (MS) is the surface generated by the front of the same solvent sphere, see figure 2.1, C.

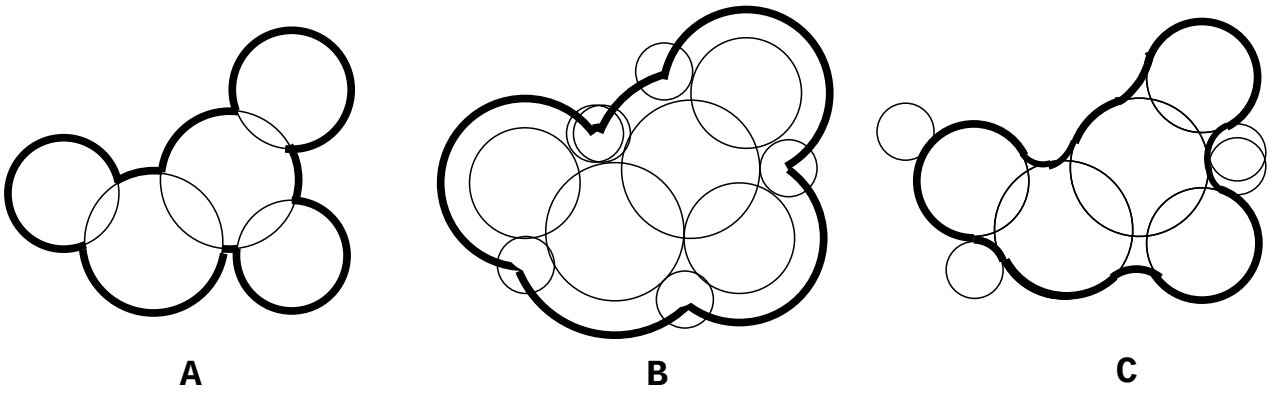

Figure 2.1: Molecular surface models: A. van der Waals surface (VS), B. solvent accessible surface (SA), and C. molecular surface (MS).

In early work on algorithms for surface area computation, Richards and others applied Voronoi diagrams to decompose a molecule (Richards, 1974; Finney, 1975; Gellatly & Finney, 1982). The Voronoi diagram divides the space into cells, one per atom. See figure 2.2 for a 2-dimensional version, where atoms are modeled as disks. The boundaries of the cells neatly divide up the entire molecule. A Voronoi cell is *generated* by the atom inside, and it consists of the part of space closest to its generating atom. Thus, for every point inside a cell, its distance to

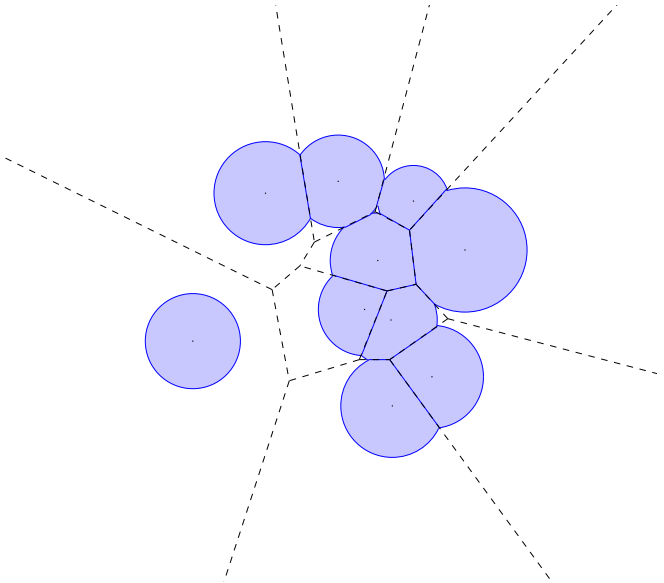

Figure 2.2: The weighted Voronoi diagram (dashed lines) decomposes the molecule into convex pieces.

the generating atom is less than (or equal to) its distance to any other atom in the molecule (Voronoi, 1907). A variety of application of Voronoi diagrams to biology and chemistry can be found in (David & David, 1982).

Exactly how the space is divided up depends on what kind of distance is used. If every atom has the same van der Waals radius, we use the Euclidean distance between the point of interest and the center of the atom. As a result, the dividing plane for two equally large atoms is the bisector plane, on which every point is equidistant to the centers of both atoms. When we have different van der Waals radii for different atoms, we use the square of the length of the tangent line segment to the surface of an atom as the *weighted distance*. The dividing plane of two atoms is called the *radical plane*, which is in general different from although parallel to the bisector plane of the centers. Every point on the radical plane has equally long tangent line segments to the both atoms. Figure 2.3 shows a radical line defined for two disks and illustrates the weighted distance to the disks. The decomposition obtained using weighted distance is called the *weighted Voronoi diagram*. Its cells have the same properties for atoms of mixed radii as the (unweighted) Voronoi cells have for atoms of identical radii. Figure 2.2 illustrates the concept by superimposing atoms modeled as disks and the weighted Voronoi diagram they define. Note that the cells neatly decompose the molecule into convex pieces. Furthermore, the cells themselves fill space without redundant overlap.

In the general 3-dimensional case, two weighted Voronoi cells either have no common intersection or they intersect in a planar facet, three cells either have no common intersection or they intersect in a common straight edge, and four cells either do not intersect or they intersect in a common vertex. Five cells do not share any common points at all. The part of an atom contained in a weighted Voronoi cell is convex since the cell is convex and so is the atom ball. Intuitively, we can sense that the weighted Voronoi diagram contains information about questions of nearness among the atoms. This information will be made more explicit shortly.

A difficulty encountered by early attempts to apply Voronoi diagrams directly to molecules originates with cells that extend to infinity (Richards, 1985). Whereas atoms occupy only a finite part of space, the Voronoi cells of some atoms at the surface of the molecule are infinitely large. This is a potential problem if Voronoi cells are used for computations and several heuristics dealing with this problem have been proposed, see (Richards, 1985; Richards, 1974; Finney, 1975; Gellatly & Finney, 1982). For example, hypothetical solvent molecules were set up

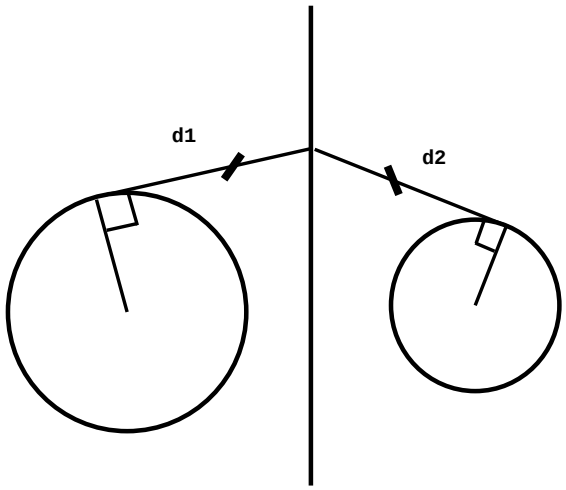

Figure 2.3: Points on the radical line have equally long tangent line segments to the two atoms. CAN WE DROP THE LABELS 'D1' AND 'D2' IN THE FIGURE?

around the molecule, with the sole purpose of defining positions of additional radical planes that close off Voronoi cells of atoms. As described in the next section, we use the dual of the Voronoi diagram as our combinatorial map to carry out area and volume computation. Together with the principle of inclusion-exclusion this leads to an analytic method for area and volume computation, without the help of any heuristic techniques.

2.2 Delaunay complex

The problem of triangulating the atom centers might at first seem unrelated to the construction of the Voronoi diagram. The gray lines in figure 2.4 show the edges of a complex that covers the convex hull of the atom centers. To describe the connection between the Voronoi diagram and a triangulating complex, we need to first define the convex hull of a set of points. Suppose we have a finite set of points in $\mathbb{R}^3$, for example the atom centers of a molecule. If we stretch a plastic wrap tightly around the points, the shape taken up by the wrap then gives the boundary of a convex body referred to as the *convex hull* of the point set. We decompose the convex hull into a collection of tetrahedra with points in the set as vertices. To cleanly fill up the convex hull, the edges of tetrahedra are not allowed to cross or intersect the triangles, except they may share common vertices. More formally, any two tetrahedra in the decomposition are either disjoint or they intersect in a common triangle or a common edge or a common vertex. Such a decomposition is called a *simplicial complex*. The complex *triangulates* the set of points. In $\mathbb{R}^2$, the tetrahedra are reduced to triangles, see figure 2.4.

A set of points can be triangulated in many ways. The complex shown in figure 2.4 is a well-known one, called the Delaunay complex or triangulation. It has many nice geometric properties, see e.g. (Rajan, 1994), and it reflects the boundary overlap among the Voronoi cells.

The Delaunay complex is derived from the Voronoi diagram by the following direct translation. If a ball has a non-empty Voronoi cell then it is a vertex in the complex. If two Voronoi cells share a common facet (contained in the radical plane of the two corresponding balls) then the edge connecting the two ball centers is in the complex. If three Voronoi cells share a common edge, then the triangle spanned by the three ball centers is in the complex. Finally, if four Voronoi cells share a common point, then the tetrahedron spanned by the four ball centers is in the complex. Observe that the rule guarantees if an edge belongs to the complex then its endpoints are vertices in the complex. Similarly, if a triangle belongs to the complex then so do its edges, and if a tetrahedron belongs to a complex then so do its triangles. This enumeration accounts for all intersection patterns among Voronoi cells. The vertices, edges, triangles, and solid tetrahedra together form a special complex, called the *Delaunay complex*

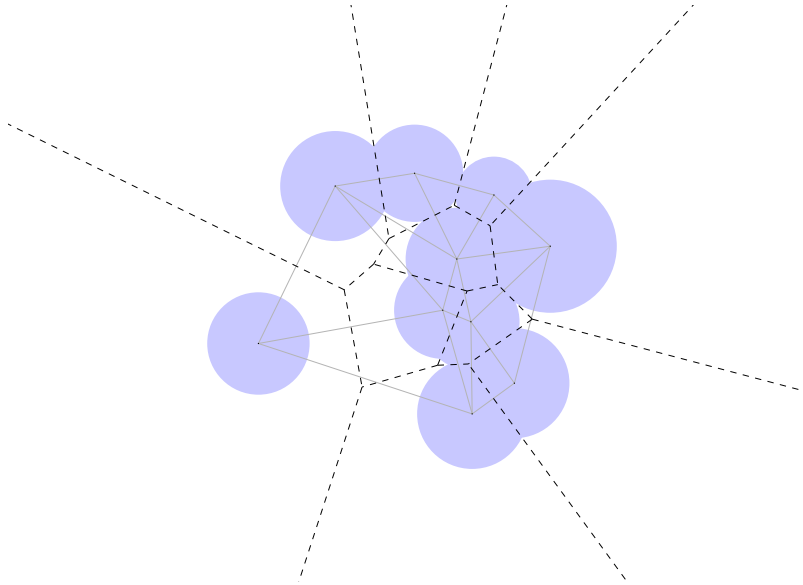

Figure 2.4: Gray lines show the Delaunay complex of the centers of the atoms. Note the duality with Voronoi diagram shown in dashed lines.

of the point set. It is named after the Russian mathematician Boris Delaunay (Delaunay, 1934) building on work of his teacher, Georges Voronoi (Voronoi, 1907).

The duality between the Delaunay complex and the Voronoi diagram is reflected in a number of aspects. In $\mathbb{R}^3$, each vertex, edge, triangle, tetrahedron in the Delaunay complex corresponds to a cell, facet, edge, vertex in the Voronoi diagram. Similarly in $\mathbb{R}^2$, each vertex, edge, triangle in the Delaunay complex corresponds to a cell, edge, vertex in the Voronoi diagram. See figure 2.4 for an illustration of the 2-dimensional case, where the Delaunay complex and the the Voronoi diagram are superimposed. As a result of the duality, the Delaunay complex and the Voronoi diagram contain exactly the same combinatorial information, although this information is represented differently. An important consequence of the duality is that algorithms for Delaunay complexes have meaning for Voronoi diagrams and vice versa. In particular, it seems easier to design a robust algorithm for constructing Delaunay complexes than for constructing Voronoi diagrams. The main reason is that the Delaunay complex comprises no new geometric information and all edges, triangles, and tetrahedra can be stored combinatorially as pairs, triplets, and quadruplets of vertex indices. In contrast, the Voronoi diagram contains vertices that are not part of the input data.

The method we use to compute the Delaunay complex of a set of spherical balls is based on the notion of *flipping* (Joe, 1991; Edelsbrunner & Shah, 1992; Edelsbrunner & Mücke, 1994). This is implemented in the DELCX program. The most common type of flip in $\mathbb{R}^2$ changes the diagonal of a convex quadrilateral. Before the flip the quadrilateral is decomposed into two triangles sharing a diagonal, after the flip it is decomposed into the two triangles sharing the other diagonal. There are two other types of flips. One adds a point in the middle of a triangle and decomposes the triangle into three smaller ones, the other removes a vertex common to three triangles and replaces them by their union, which is again a triangle. In $\mathbb{R}^3$, there are four types of flips. An edge shared by three tetrahedra can be replaced by the triangle shared by two tetrahedra occupying the same part of space. Inversely, a triangle shared by two tetrahedra can be replaced by the edge shared by three tetrahedra. A single tetrahedron can be decomposed into four by adding a new vertex inside. Inversely, a vertex shared by four tetrahedra can be removed replacing the four by their union, which is again a tetrahedron.

The program DELCX adds one point at a time via a flip that decomposes the containing tetrahedron into four. Successive flips are used to transform the neighborhood of the new point into a decomposition with Delaunay tetrahedra, (Joe, 1991; Edelsbrunner & Shah, 1992). This algorithm has time-complexity $O(n \log n)$, implying that

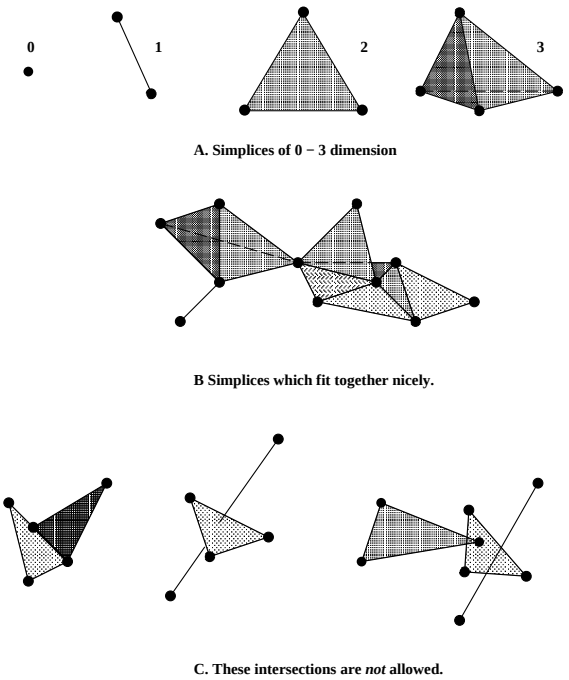

Figure 2.5: A. A 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron. B. A collection of simplices that fit together nicely in 3-dimensional space. C. Intersection patterns among simplices that are not allowed in a complex. JIE, CAN WE DELETE THE INDIVIDUAL CAPTIONS, EXCEPT THE A, B, C?

the computing time required scales roughly like to $n \log n$, where n is the number of atoms. This is a consequence of the spatial distribution of atoms in molecules, which typically form a dense arrangement of balls. All atoms have bond lengths of roughly the same length and the distribution is more or less of uniform density. For such a spatial arrangement, the number of tetrahedra, triangles, edges, and vertices in the Delaunay complex is $O(n)$ and the required time is $O(n \log n)$.

2.3 Simplicial complexes

We are interested in the Voronoi decompositions of the molecule rather than the decomposition of the entire space. Similarly, we are not just interested in the Delaunay complex of the molecular convex hull but rather the part that corresponds to the molecule. To understand how to obtain information from the Delaunay complex about the actual molecule, we first need to describe a few topological concepts, which can be found in introductory texts on topology, see e.g. (Munkres, 1984).

Topology creates a unified language and notation by calling a vertex a 0-*simplex*, an edge a 1-*simplex*, a triangle a 2-*simplex*, and a solid tetrahedra a 3-*simplex*. The integer number indicates the intrinsic dimension. Examples are shown in figure 2.5 A. Observe that the boundary of a simplex consists of other simplices, albeit their dimensions are lower. These lower-dimensional simplices are the *faces* of the original simplex.

Complicated geometric objects can be build from a collection of simplices. The goal is to construct the object in an organized fashion. This is achieved by adhering to the following two rules:

(i) for every simplex used, its faces are also be part of the construction, and

(ii) the common intersection of any two simplices is either empty or a face of both simplices.

Figure 2.5, `B` and `C`, illustrates these ideas. If the above two rules are followed, the resulting object is called a *simplicial complex*. We have already seen an example of such a complex, namely the Delaunay complex of a molecule. It contains a wealth of combinatorial information about the molecule, as will be apparent shortly.

2.4 Alpha complexes

A molecule has many simplices in its Delaunay complex. We need to organize them rather than randomly pile them up. There is a natural way to arrange the simplices in a sequence (Edelsbrunner & Mücke, 1994). The idea is roughly analogous to the principle employed in gel filtration or gel electrophoresis, where molecules are separated according to their size. To explain the idea, we first ignore the differences in size among atoms and assume all have the same radius. For all the atoms in this peculiar molecule, we start to grow balls simultaneously from each atom center by gradually increasing the uniform radius, α. A ball grows only inside its own Voronoi cell and is clipped when it reaches the boundary of this cell. A simplex is collected at the moment the clipped balls growing from its vertices have a common boundary intersection.

We have the following scenario. At the beginning when radius is 0, we only have vertices in our collection, and we take the atom centers as the first elements to appear in our sequence of simplices. All vertices appear simultaneously. Because the balls gradually grow they will eventually overlap. At the moment when the boundaries of 2 clipped balls overlap, we time-stamp the corresponding edge in the Delaunay complex, and add it at the end of the evolving sequence. When the boundaries of 3 clipped balls grow to overlap, we time-stamp the corresponding triangle in the Delaunay complex and add it to the sequence. We do the same for the solid tetrahedra whenever the boundaries of 4 clipped balls overlap. When the balls are grown large enough, all the simplices in the Delaunay complex will have been put in the sequence. Thus we have organized all the simplices into a sequence. In topology, this sequence is called a *filter* of the simplicial complex. The filter has an important property. If we sequentially choose any number of simplices from the beginning of the sequence we obtain a collection of simplices that itself forms a complex. It is a *subcomplex* of the Delaunay complex. The sequence of such subcomplexes is called a *filtration* of the Delaunay complex.

Note that when 2 balls grow, their bounding spheres intersect in a circle that sweeps out the bisector plane. A piece of this plane is found as a 2-dimensional facet in the Voronoi diagram. Now we consider atoms of different sizes. We assign the van der Waals radius of the associated atom as the initial radius r_0 to each ball. We grow or shrink the ball by changing the α parameter: the actual radius is

$$r_\alpha \quad = \quad \sqrt{r_0^2 + \alpha^2}.$$

For increasing α the ball grows and for decreasing α the ball shrinks until it vanishes when $\alpha^2 = -r_0^2$. This would never be the case if we chose α from the set of real numbers. To avoid this technical difficulty, we choose α^2 from the set of real numbers, positive and negative, which really means we choose α from the set of non-negative reals or positive multiples of the imaginary unit, $\sqrt{-1}$. When $\alpha = 0$, we have the actual size of the molecule. By growing the balls in this fashion we again obtain a sequence of the Delaunay simplices, the filter.

If we increase α from its least possible value, we can imagine an index moving along the filter from its start. When we stop at a certain value, all simplices to the left of the index have shown up and form a simplicial complex. The corresponding ball diagram at the moment is characterized by the particular α value, which controls the ball size. The simplicial complex associated with an α value is a subcomplex of the Delaunay complex and is referred to as the *alpha complex* (Edelsbrunner *et al.*, 1983; Edelsbrunner & Mücke, 1994). The *alpha shape* is the part of space covered by simplices in the alpha complex. Figure 2.6 shows the alpha complex of the 2-dimensional molecule in figures 2.2 and 2.4 for a small, medium, and large value of α. Because of the filter property, simplices in an alpha complex for a smaller value of α are present in an alpha complex of a larger value of α. As a result, the alpha complex for a smaller value of α is always a subcomplex of one for a larger value of α, and both are subcomplexes of the Delaunay complex. Furthermore, the number of possible alpha complexes for a molecule cannot exceed the number of simplices in the Delaunay complexes.

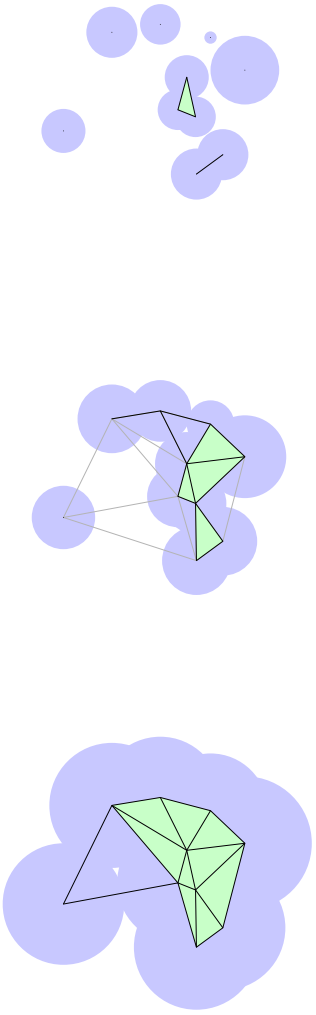

Figure 2.6: A. The alpha complex for the small value of alpha consists mostly of vertices, together with 4 edges and one triangle. B. The alpha complex for the medium value of alpha consists of 2 components, one just a vertex and the other consisting of quite a few triangles, edges and vertices. It contains the complex in A as a subcomplex. C. The alpha complex for the large value of alpha is connected and contains the other alpha complexes as subcomplexes. CAN THE FIGURES BE ARRANGED FROM LEFT TO RIGHT, TO SAVE SPACE?

The molecule with van der Waals atom radii is faithfully represented by the alpha complex for $\alpha = 0$. Together with the Delaunay complex it contains a wealth of information about the spatial arrangement of the molecule. The key to making a connection to the actual molecule is to link the alpha complex with the weighted Voronoi decomposition of the molecule. As demonstrated in (Edelsbrunner, 1993), the alpha complex reflects or encodes combinatorial, topological, and metric information about the molecule. The combinatorial equivalence between the alpha complex and the corresponding Voronoi decomposition of the union of balls is most obvious from the definition: each simplex indicates a collection of clipped balls with non-empty common intersection.

The topological correspondence between the molecule and the alpha complex (for $\alpha = 0$) can be seen from figure 2.6. Each component of the union of disks contains a component of the complex, and each hole of the union of disks is contained in a hole of the complex. More precisely and generally, the molecule is homotopy equivalent to the alpha complex (for $\alpha = 0$), see (Edelsbrunner, 1993). Consider for example the tiny hole in the molecule represented by the disk union in figure 2.6 `B`. It corresponds to a much larger triangular hole in the alpha complex. If the atom sizes were a little larger, such as in `C` where that specific hole has disappeared, the planar triangle would have been added so as to fill the hole in the alpha complex. One characteristic of combinatorics is that changes occur in discrete steps, such that the actual size of the hole has no direct influence on the nature of the alpha complex. Taking advantage of this topological correspondence, we can locate all voids inside a protein, regardless their sizes. We can also identify atoms facing outside with precision. Void computations will be explained in a companion paper.

The third correspondence between the molecule and its alpha complex is metric. As an application, we will see in the next section and the companion paper how the simplices in the alpha complex can be translated into a combinatorial expressions for the volume and surface area of a molecule and its voids.

3 Computing Area and Volume

As mentioned in the introduction, the computation of surface area and volume of a molecule has received much attention in the past. The problem is difficult because the spherical balls modeling the van der Waals atoms overlap due to chemical bonds, van der Waals contacts and solvent contacts (when a solvent probe can touch two or more atoms simultaneously). Were the atoms completely isolated, we would only need to sum up the area and volume of each individual ball. An obvious approach is to use the principle of inclusion-exclusion: when two atoms overlap, we subtract the overlap, when three atoms overlap, we add the overlap, etc. This continues when there are four, five, or more atoms intersecting. At a combinatorial level, the principle of inclusion-exclusion is related to the Gauss-Bonnet theorem used by Connolly (Connolly, 1983). It is still difficult to accurately keep track of where the overlaps occur, especially when there are many different combinatorial situations (Petitjean, 1994). Current numerical algorithms for estimating area and volume (Shrake & Rupley, 1973) ignore the combinatorial problem. SHOULD WE SAY SOMETHING ABOUT HOW THEY IGNORE THE PROBLEM, E.G. BY ASSUMING THERE IS AT MOST 3-FOLD OVERLAP?

3.1 Direct inclusion-exclusion

Once the alpha complex is constructed, it provides a route to untangling the combinatorial complexity of atom intersections. This is done by trimming the full inclusion-exclusion formula until it contains no redundant terms. An example of area computation by alpha shape trimmed inclusion-exclusion is shown in figure 3.7 `A`. Let b_1, b_2, b_3, b_4 be the four disks. To simplify the notation we write A_i for the area of b_i, A_{ij} for the area of $b_i \cap b_j$, and A_{ijk} for the area of $b_i \cap b_j \cap b_k$. The total area of the union, $b_1 \cup b_2 \cup b_3 \cup b_4$, is

$$\begin{aligned} A_{\text{total}} &= (A_1 + A_2 + A_3 + A_4) \\ &- (A_{12} + A_{23} + A_{24} + A_{34}) \\ &+ A_{234}. \end{aligned}$$

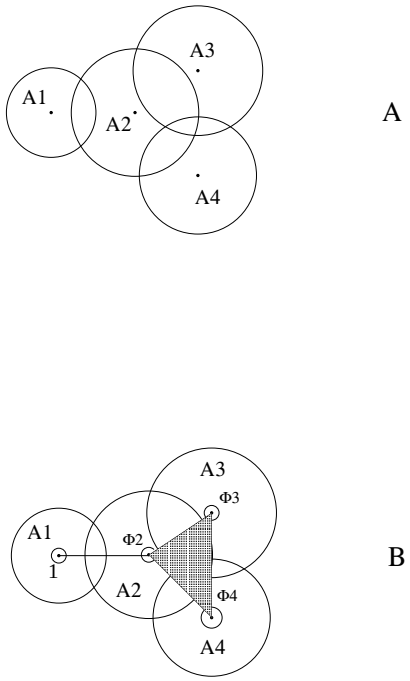

Figure 3.7: The alpha complex of a molecule can be used to trim the list of terms implied by the straightforward application of the inclusion-exclusion principle. The area formula contains a term per simplex in the alpha complex. Can we put the figures side by side, to save space? Also, please relabel balls to b. Also, can you enlarge the circles without changing the alpha complex so that there is at least one type of intersection that does not correspond to a simlex? Actually, it would maybe be easiest to use Pat's program to redo the figure without labels, which would be fine.

We add the area of b_i if the corresponding vertex belongs to the alpha complex, we subtract the area of $b_i \cap b_j$ if the corresponding edge belongs to the alpha complex, and we add the area of $b_i \cap b_j \cap b_k$ if the corresponding triangle belongs to the alpha complex. This is an example of what we call the *direct* inclusion-exclusion method. In three dimensions the most complicated terms are intersections of 4 spherical balls, and these terms correspond to tetrahedra in the alpha complex.

It turns out that for 3-dimensional molecules in $\mathbb{R}^3$ intersections of 5 or more atom balls at a time can always be reduced to a signed combination of intersections of 4 or fewer balls, (Edelsbrunner, 1993) That overlaps of more than 4 balls actually occur for real molecular data and reductions are applicable is argued in (Petitjean, 1994). In particular, the union of all balls can be expressed as a signed combination of intersections, with a term per vertex, edge, triangle, tetrahedron in the alpha complex. Each vertex corresponds to a single ball taken positive, each edge corresponds to the intersection of two balls taken negative, each triangle corresponds to the intersection of three balls taken positive, and each tetrahedron corresponds to the intersection of four balls taken positive. In other words, following the combinatorial information of the alpha complex avoids redundant high order intersections altogether. Although the resulting formula is much smaller than entire inclusion-exclusion formula, it expresses the exact volume and surface area of the molecule, see (Edelsbrunner, 1993).

A similar but weaker result has been discovered for the 2-dimensional case earlier (Kratky, 1981). Kratky observed that the common intersection of 4 or more circular disks can be reduced to a signed sum of lower order intersections. Successive application of this idea eventually leads to a formula where each term is the intersection of at most 3 disks. This formula may still contain redundant terms and will generally be longer than the formula obtained from the alpha complex. Kratky's idea has been applied to 3 dimensions without proof (Gibson & Scheraga, 1987).

In this context it is worth mentioning that the intersections of up to 4 atom balls derived from the alpha complex all have a uniform structure. This simplifies the necessary analytic computation of volume and area for such small groups of balls. To explain the uniformity, consider a tetrahedron in the alpha complex whose vertices are the centers of balls b_1, b_2, b_3 and b_4. First, the intersection of the balls is non-empty. Second, the common intersection of any 3 of the 4 balls is non-empty and not contained in the 4th ball. Third, the common intersection of any 2 of the 4 balls is non-empty and not contained in the union of the other two balls. Finally, no ball is contained in the union of the other three balls. Such a configuration is referred to as an *independent* collection of balls. The above reductions are possible because collections of 5 or more spherical balls in $\mathbb{R}^3$ cannot be independent and because every non-independent collection can be reduced to a signed combination of smaller collections. This also implies that a formula obtained by following the ideas of Kratky can still be further reduced until all terms correspond to independent collections.

3.2 Short inclusion-exclusion

Although the inclusion-exclusion formula obtained from the alpha complex contains only independent terms and is therefore minimal, it is possible to find even shorter expressions of area and volume if non-integer coefficients are used. Indeed, we can use the angles at the atom centers relative to its neighbors in the alpha complex as coefficients for that purpose. All angles are measured as fractions of circles or spheres and are thus automatically normalized between 0 and 1. This is what we refer to as the *short* inclusion-exclusion method. It expresses the surface area as a sum of areas of intersections of at most 3 balls, each with an angle coefficient. Similarly for volume, except that the total volume of all tetrahedra in the alpha complex needs to be added to the sum of intersections. Tetrahedra volumes are significantly easier to compute than volumes of the intersections of 4 balls, which leads to improved running-time even in the case of volume. The short inclusion-exclusion method is important for practical computation.

Rather than describing the short formula in detail we refer to (Edelsbrunner, 1993) and present a small 2-dimensional example. Figure 3.7 B illustrates the method in 2 dimensions. In this case, the alpha complex consists of a single triangle, 4 edges and 4 vertices. Let ϕ_2, ϕ_3, ϕ_4 be the outside angles of the triangle. Then the area of

the disk union is

$$\begin{aligned} A_{\text{total}} &= (1 \cdot A_1 + \phi_2 \cdot A_2 + \phi_3 \cdot A_3 + \phi_4 \cdot A_4) \\ &- (1 \cdot A_{12} + \frac{1}{2} \cdot A_{23} + \frac{1}{2} A_{24} + \frac{1}{2} A_{34}) \\ &+ A, \end{aligned}$$

where A is the area of the triangle. In general, A is the total area of all triangles in the alpha complex, or in 3 dimensions, it is the total volume of all tetrahedra.

3.3 Algorithms

Following the above illustrations, we briefly describe the algorithm as implemented in the VOLBL package. The notation uses $\mathcal{K}$ for the alpha complex, σ for a simplex in $\mathcal{K}$, i for a vertex, ij for an edge, ijk for a triangle, and $ijkl$ for a tetrahedron. The algorithm expressing the direct inclusion-exclusion method can be written as follows.

$V := A := 0.0$;
for each $\sigma \in \mathcal{K}$ **do**
  if σ is a vertex i **then** $V := V + vol(b_i)$; $A := A + area(b_i)$ **endif;**
  if σ is an edge ij **then** $V := V - vol(b_i \cap b_j)$; $A := A - area(b_i \cap b_j)$ **endif;**
  if σ is a triangle ijk **then** $V := V + vol(b_i \cap b_j \cap b_k)$; $A := A + area(b_i \cap b_j \cap b_k)$ **endif;**
  if σ is a tetrahedron $ijkl$ **then** $V := V - vol(b_i \cap b_j \cap b_k \cap b_l)$; $A := A - area(b_i \cap b_j \cap b_k \cap b_l)$ **endif**
endfor.

Here, $vol()$ and $area()$ are metric functions for volume and area computations of the insection of up to 4 balls. As mentioned earlier, all intersections are of a uniform type, namely the atom balls involved are independent.

For the short inclusion-exclusion method, we divide our computations into three parts: void computation, shape computation and outside-fringe computation. *Voids* here are defined in strictly topological sense, that is, they are cavities buried inside the molecule that have no open outlets to connect them to the outside. In void computation, the volume of a void is computed by first measuring the volume of the corresponding alpha complex void, then measuring the volume and area of the atom balls that reach into this complex void. Volume of the complex void minus the volume covered by the reaching in balls gives the volume of the actual void in the molecule. More details of void computation can be found in the companion paper, see also (Edelsbrunner *et al.*, 1995). Shape computation is straightforward: it sums the volume over all the tetrahedra found in the alpha complex.

The *outside-fringe* is the part of the molecule that lies outside the alpha complex. It is measured by the short inclusion-exclusion method which was illustrated by our second example earlier. The algorithm can be summarized as follows.

$V := A := 0.0$;
for each σ on the outside boundary of $\mathcal{K}$ **do**
  if σ is a vertex i **then** $V := V + \phi_\sigma \cdot vol(b_i)$; $A := A + \phi_\sigma \cdot area(b_i)$ **endif;**
  if σ is an edge ij **then** $V := V - \phi_\sigma \cdot vol(b_i \cap b_j)$; $A := A - \phi_\sigma \cdot area(b_i \cap b_j)$ **endif;**
  if σ is a triangle ijk **then** $V := V + \phi_\sigma \cdot vol(b_i \cap b_j \cap b_k)$; $A := A + \phi_\sigma \cdot area(b_i \cap b_j \cap b_k)$ **endif**
endfor.

For a vertex σ, ϕ_σ is the solid angle at σ outside the alpha complex. For an edge σ, ϕ_σ is the dihedral angle at σ outside $\mathcal{K}$. Finally, for a triangle σ, ϕ_σ is 1 if both sides of σ lie on the outside and $\frac{1}{2}$ if only one side lies on the outside of $\mathcal{K}$.

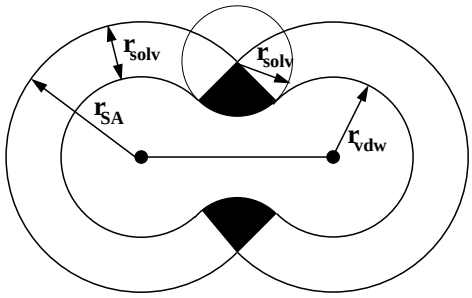

Figure 3.8: Correspondence between the molecular surface (MS) and the solvent accessible (SA) model of a molecule. JIE, CAN WE CHANGE THE LABELING TO r_{VS} AND r_s?

3.4 Area and volume of MS model

As illustrated in figure 2.1, the combinatorial structure of the SA and the MS models of a molecule are the same. Both are represented by the same alpha complex. Each sphere patch of the SA model corresponds to somewhat smaller but otherwise identical sphere patch in the MS model. Each circular arc of the SA model corresponds to a torus patch swept out by the solvent sphere as its center moves along the arc. Each corner point of the SA model corresponds to an inverse sphere patch that lies on the surface of the solvent sphere whose center is the corner point. Figure 3.8 illustrates the relationship between the MS and the SA model in 2 dimensions, and it indicates how the metric sizes of the MS patches can be computed from the SA model, see also (Edelsbrunner *et al.*, 1995).

We explain this in more detail. The ball radius of an atom in the SA model, r_{SA}, is the sum of its van der Waals radius r_{VS} and radius of the solvent sphere, r_s. We shrink the SA model by excluding all volume that can be reached by a copy of the solvent sphere with center on the SA surface. The effect of this shrinking process is different for the sphere patches, the circular arcs, and the corner points of the SA model. A sphere patch shrinks to radius r_{VS} towards its center; this gives the convex sphere patches of MS model. The circular arcs grows into a torus patch, which is a saddle shaped patch. In 2 dimensions, the torus patch becomes an arc of a circle outside the MS model, see figure 3.8. A corner point at the intersection of 3 spheres in the SA model grows into a triangular patch on the solvent sphere centered at the corner point.

Measuring the MS model requires the computation of area and volume of various basic geometric pieces. For example, we have the following formula for computing the fraction of the torus swept out by a revolving circular arc bounding the disk sector shown in black in figure 3.8.

$$\mathrm{Area} \;=\; 2\pi \cdot r_s(r_d \cdot \theta - r_s \cdot \sin\theta),$$

where r_d is the radius of the circular arc in the SA model, which is also the center circle of the torus. The following formula computes the volume of the piece of solid torus swept out by the disk sector.

$$\begin{aligned}\mathrm{Volume} \;&=\; \frac{\pi}{3} \cdot \tan\theta \cdot (r_d^3 - r_{a,\mathrm{VS}}^3 \cdot \cos\theta) \\ &+\; \pi \cdot r_s \cdot \sin\theta \cdot (r_d \cdot r_s \cdot \cos\theta - r_d^2 - r_s^2) \\ &+\; \frac{\pi}{3} \cdot r_s^3 \cdot \sin^3\theta + \pi \cdot r_s^2 \cdot r_d \cdot \theta.\end{aligned}$$

JIE, WE NEED TO EXPLAIN WHAT $r_{a,\mathrm{VS}}$ AND θ ARE.Formulas for other basic geometric pieces, such as ball wedges, ball sectors, sphere caps, etc., can be found in (Edelsbrunner & Fu, 1994). ←???

JIE, I TOOK OUT ALL THE DISCUSSIONS OF CONTRIBUTIONS TO MS SINCE IT SEEMS NOT WELL-DEFINED. THE TEXT IS STILL IN THE FILE, AND WE SHOULD DISCUSS THIS IN REGARD TO THIS PAPER AS WELL AS FUTURE VERSIONS OF VOLBL. ←???

3.5 Checking correctness

The program VOLBL has a built-in mechanism for checking the correctness of the computations. Using the 'checking' option, it computes the surface area and volume of both SA and MS models in two different ways and compares the results. The two ways correspond to the direct and the short inclusion-exclusion formulas. After computing area and volume, the two results are checked to match to the last digit of precision. Besides checking the overall area and volume, VOLBL also checks the correctness of atomic contributions, which are again computed in 2 different ways and results are finally compared.

4 Results

The structures chosen from the protein databank for calculating area and volume are listed in table 4.1. The computation based on alpha complexes involves four steps. The first step assigns the radii to the atom centers. The radius of an atom is its van der Waals radius plus the probe radius. When the radius of the probe is assigned to 0.0Å, both the solvent accessible and molecular surface results will give van der Waals area and volume. The assignment is done through a utility program, PDB2WA. Van der Waals radii are taken from Richards' parameter set in VADAR. All accessible and molecular surface calculations use the probe radius of 1.2Å. The second step computes the (weighted) Delaunay complex of the collection of atom balls, using the coordinates of the sphere centers and the radii as assigned by the first step. This is achieved by the program DELCX. Its expected running time is on the order of $n \log n$, n the number of atoms, assuming they form a more or less dense distribution in space. Implicitly, DELCX also builds the (weighted) Voronoi diagram that decomposes the molecular space into non-overlapping convex pieces. The third step constructs the filter of the Delaunay complex using the program MKALF (Edelsbrunner & Mücke, 1994). The running time is again on the order of $n \log n$ assuming dense distribution. The final step uses VOLBL to compute the metric area and volume, as well as the atomic contributions thereof. The SA and MS area and volume are computed simultaneously using one probe radius. All calculations are performed without corrections for the cusp condition. VOLBL requires time proportional to n. The constant of proportionality is relatively high so that the actual running time of VOLBL sometimes exceeds the time required for constructing the Delaunay complex and its filter, even for large proteins with tens of thousands of atoms. The actual running time for the short inclusion-exclusion version of VOLBL is significantly less than that for the direction inclusion-exclusion method. This is mostly due to the smaller constant of proportionality: only triplets rather four-tuplets of intersecting balls need to be considered.

The computed area for the proteins listed in table 4.1 are given in tables 4.2 through 4.4. For comparison, VS, SA and/or MS area are also computed using several programs distributed in the VADAR and ARVOMOL packages (Wishart *et al.*, 1994; Pacios, 1993). With the exception of VS volume, comparisons are given with the computations obtained with at least one analytical and one approximation method. From VADAR, VS volume is computed using Richards' VOLUME program (the bisection/radical plane method), and SA area is computed using Richmond's analytical ANAREA program. From ARVOMOL, VS, SA and MS area and volume is computed using Pascual-Ahuir *et al.*'s GEPOL program. VS and MS area are also computed using Connolly's analytical MS/MSDOT program (Pacios, 1993). In the calculations with GEPOL, we used 60 triangles to approximate an atom sphere. For consistency, all calculations use Richards' van der Waals radii parameter set (available from the VADAR package (Wishart *et al.*, 1994)) and a probe radius of 1.2Åwhen SA or MS area/volume are computed.

Table 4.2 reports the VS surface area computed with the alpha shape based analytical method, VOLBL, with Connolly's analytical method, MSDOT, and with Pascual-Ahuir *et al.*'s GEPOL method. Table 4.3 reports the SA areas computed with VOLBL, ANAREA and GEPOL. Table 4.4 reports the molecular surface area computed with VOLBL, MSDOT and GEPOL. Residue contributions of SA surface area for three proteins, `5mbn` (deoxy myoglobin), `2ptn` (pancreatic trypsin) and `2rns` (ribonuclease), are shown in figures 4.1, 4.2 and 4.3. The SA area contribution results computed with VOLBL are compared with the results obtained with Richmond's ANAREA.

Tables 4.5 through 4.7 compare results of volume computation using the alpha shape based VOLBL with results

Table 4.1: List of proteins used in the calculations of molecular area and volume.

PDB name	# res	protein name	authors and reference
1eca	137	Hemoglobin (erythrocruorin, aquo met)	(Steigemann & Weber, 1979)
1nxb	62	Neurotoxin B	(Tsernoglou, 1978)
1rbc	124	Ribonuclease S mutant met13ala	(Varadarajan & Richards, 1992)
1rbe	124	Ribonuclease S mutant met13phe	(Varadarajan & Richards, 1992)
1rbf	124	Ribonuclease S mutant met13gly	(Varadarajan & Richards, 1992)
1rbg	124	Ribonuclease S mutant met13ile	(Varadarajan & Richards, 1992)
1rbh	124	Ribonuclease S mutant met13leu	(Varadarajan & Richards, 1992)
1rbi	124	Ribonuclease S mutant met13val	(Varadarajan & Richards, 1992)
2act	218	Actinidin	(Baker & Dodson, 1980)
2cha	248	Alpha chymotrypsin a	(Birktoft & Blow, 1972)
2lyz	129	Lysozyme	(Diamond, 1974)
2ptn	230	Trypsin	(Walter *et al.*, 1982)
2rns	124	Ribonuclease S	(Kim *et al.*, 1992)
2sn3	51	Scorpion neurotoxin	(Zhao, 1992)
3cyt	104	Cytochrome C	(Takano & Dickerson, 1980)
3rn3	125	Ribonuclease A	(Howlin *et al.*, 1989)
4pti	58	Trypsin inhibitor	(Marquart *et al.*, 1983)
5mbn	154	Myoglobin (deoxy)	(Takano, 1984)
1arb	263	Achromobacter protease I	(Tsunasawa *et al.*, 1989)
1cau	424	Canavalin	(Ko *et al.*, 1993)
1cse	345	Subtilisin carlsberg	(Bode *et al.*, 1987)
1ecd	136	Hemoglobin (erythrocruorin, deoxy)	(Steigemann & Weber, 1979)
1icm	131	Intestinal fatty acid binding protein	(Eads *et al.*, 1993)
1mbd	153	Myoglobin (deoxy, pH 8.4)	(Phillips & Schoenborn, 1981)
1plc	99	Plastocyanin	(Guss *et al.*, 1992)
1rro	108	Rat oncomodulin	(Ahmed, 1993)
1thm	279	Thermitase	(Teplyakov *et al.*, 1990)
1ycc	103	Cytochrome C	(Louie & Brayer, 1990)
3sdh	292	Hemoglobin I	(Royer-Jr., 1994)
4gcr	174	Gamma-B crystallin	(Najmudin *et al.*, a)
5p21	165	C-*H-Ras p21 protein	(Pai *et al.*, 1990)

Table 4.2: Computed van der Waals surface (VS) area (in $Å^2$) of selected proteins.

protein	# res	VOLBL	MS/MSDOT	GEPOL
1eca	137	13928.2	13842.9	13909.4
1nxb	62	5673.7	5672.1	5683.3
1rbc	124	12077.2	11989.8	12040.3
1rbe	124	12126.1	12042.2	12120.8
1rbf	124	12083.7	11958.3	12025.2
1rbg	124	12224.3	12033.1	12106.8
1rbh	124	12223.6	12061.3	12160.8
1rbi	124	12187.5	12046.1	12102.7
2act	218	21288.5	21260.6	21162.0
2cha	248	22551.2	22521.6	22505.6
2lyz	129	12734.3	12700.7	12704.8
2ptn	230	21433.0	21439.9	21480.5
2rns	124	12050.8	11828.8	11867.7
2sn3	51	6293.0	6448.5	6431.8
3cyt	104	21171.2	21170.3	21125.1
3rn3	125	12390.3	12328.1	12287.9
4pti	58	5939.1	5939.4	5919.6
5mbn	154	16269.8	16192.0	16247.8
1arb	263	24796.8	24773.2	24861.1
1cau	424	37453.0	37452.7	37367.2
1cse	345	31905.8	31889.5	31955.6
1ecd	136	13889.6	13817.3	13862.2
1icm	131	14098.8	14108.9	14116.1
1mbd	153	16320.6	16220.5	16248.3
1plc	99	8856.6	9670.1	9721.2
1rro	108	11143.6	11144.1	11105.9
1thm	279	26174.1	26169.4	26123.8
1ycc	103	11152.2	11142.5	11102.6
3sdh	292	30172.1	29947.1	29968.8
4gcr	174	18875.3	18873.1	18895.6
5p21	165	17636.8	17611.3	17716.0

Table 4.3: Computed solvent accessible (SA) surface area (in $\AA^2$) of selected proteins.

protein	# res	VOLBL	MS/MSDOT	GEPOL
1eca	137	7099.9	7130.2	7111.1
1nxb	62	4035.3	4058.8	4027.7
1rbc	124	6688.0	6850.1	6721.7
1rbe	124	6627.3	6781.8	6618.2
1rbf	124	6616.1	6793.7	6661.8
1rbg	124	6701.7	6793.2	6581.7
1rbh	124	6668.8	6773.9	6647.4
1rbi	124	6665.1	6735.6	6573.8
2act	218	9152.4	9192.9	9124.4
2cha	248	10952.8	11073.1	10885.2
2lyz	129	6703.8	6766.8	6678.0
2ptn	230	9420.4	9463.6	9403.4
2rns	124	6693.2	6764.3	6669.7
2sn3	51	4411.7	4234.6	4176.6
3cyt	104	11647.3	11755.5	11633.5
3rn3	125	6884.1	6908.3	6814.9
4pti	58	3974.5	4019.6	3971.9
5mbn	154	8328.5	8415.6	8349.9
1arb	263	9707.9	9706.4	9733.9
1cau	424	20050.9	20171.5	20107.8
1cse	345	12729.4	12777.2	12718.8
1ecd	136	7068.2	7109.0	7089.8
1icm	131	7440.2	7519.5	7439.9
1mbd	153	8362.6	8500.9	8326.1
1plc	99	5330.5	5112.7	5052.4
1rro	108	5973.0	5999.7	5970.8
1thm	279	9879.2	9913.5	9873.9
1ycc	103	6498.8	6599.8	6372.7
3sdh	292	13670.0	13747.1	13700.0
4gcr	174	8910.2	8963.8	8974.2
5p21	165	8590.6	8639.4	8615.7

Table 4.4: Computed molecular surface (MS) area (in $Å^2$) of selected proteins.

protein	# res	VOLBL	MS/MSDOT	GEPOL
1eca	137	7005.0	6849.7	5970.6
1nxb	62	3437.1	3402.6	3196.5
1rbc	124	6044.9	5958.6	5337.0
1rbe	124	5886.5	5872.6	5342.5
1rbf	124	6028.7	5939.4	5351.0
1rbg	124	5975.3	5877.1	5380.4
1rbh	124	6006.1	5907.0	5391.6
1rbi	124	5947.4	5811.5	5389.1
2act	218	9094.5	9556.8	9281.8
2cha	248	11086.9	10842.4	9917.5
2lyz	129	6323.1	6293.7	5901.9
2ptn	230	9518.0	9537.8	8656.6
2rns	124	5917.2	5918.4	5922.1
2sn3	51	3800.3	3436.9	3526.5
3cyt	104	10905.7	10797.9	9796.3
3rn3	125	6130.7	6036.1	5601.3
4pti	58	3346.5	3346.0	3124.2
5mbn	154	8203.4	8048.1	6893.1
1arb	263	10009.3	9565.9	9249.1
1cau	424	20306.2	19957.6	18258.7
1cse	345	12649.5	12272.3	11610.0
1ecd	136	6950.9	6820.3	5888.1
1icm	131	7536.9	7390.0	6315.6
1mbd	153	8279.1	8152.8	6850.1
1plc	99	4586.5	4368.9	4128.9
1rro	108	5502.9	5379.5	4764.8
1thm	279	9692.5	9454.7	8228.1
1ycc	103	6060.4	5961.0	5292.8
3sdh	292	13937.2	13769.3	11763.7
4gcr	174	8618.7	8470.8	7378.7
5p21	165	8447.1	8202.3	7050.9

Figure 4.1: Residue contributions to SA area in deoxy myoglobin (5mbn) as computed with VOLBL and ANAREA.

Figure 4.2: Residue contributions to SA area in RNAase (2rns) as computed with VOLBL and ANAREA.

Figure 4.3: Residue contributions to SA area in pancreatic trypsin (2ptn) as computed with VOLBL and ANAREA.

Table 4.5: Computed van der Waals surface (VS) volume (in $Å^3$) of selected proteins.

protein	# res	VOLBL	VOLUME	GEPOL
1eca	137	13588.5	18542.2	13383.1
1nxb	62	5841.0	7443.7	5741.0
1rbc	124	11812.2	14580.9	11763.9
1rbe	124	11873.3	14366.9	11807.0
1rbf	124	11860.7	14596.9	11696.3
1rbg	124	11980.1	14658.7	11886.4
1rbh	124	11958.7	14688.8	11785.2
1rbi	124	11926.6	14659.3	11761.8
2act	218	20996.5	28069.5	21055.6
2cha	248	22431.3	31791.2	22460.5
2lyz	129	12681.7	17053.1	12594.1
2ptn	230	21031.0	29046.2	21305.9
2rns	124	11803.0	14270.5	11575.2
2sn3	51	8536.7	8335.1	6558.0
3cyt	104	20682.9	30331.4	20606.5
3rn3	125	12169.6	16120.3	12196.0
4pti	58	5836.5	7572.1	5830.2
5mbn	154	15924.6	21229.9	15944.2
1arb	263	24384.9	32636.1	16998.0
1cau	424	37337.7	52776.3	37335.9
1cse	345	31410.8	42659.3	31310.6
1ecd	136	13575.3	18555.0	13325.5
1icm	131	13579.7	19289.9	13519.3
1mbd	153	15961.5	22105.8	15764.2
1plc	99	12611.3	12580.7	9401.9
1rro	108	10687.5	14682.4	10769.9
1thm	279	25478.8	34946.0	25567.0
1ycc	103	10903.4	15043.3	10883.9
3sdh	292	29486.6	41242.6	28833.0
4gcr	174	18697.9	25479.6	18823.9
5p21	165	16987.1	23586.6	16998.0

Table 4.6: Computed solvent accessible (SA) volume (in $Å^3$) of selected proteins.

protein	# res	VOLBL	GEPOL
1eca	137	25132.8	25396.5
1nxb	62	11344.7	11273.0
1rbc	124	22068.9	21798.0
1rbe	124	22072.6	22040.4
1rbf	124	22033.9	21969.9
1rbg	124	22286.0	21819.7
1rbh	124	22226.9	21793.8
1rbi	124	22127.0	21890.7
2act	218	36926.2	36713.6
2cha	248	41058.5	40777.8
2lyz	129	23378.5	23540.1
2ptn	230	37563.5	37884.0
2rns	124	22044.1	21604.0
2sn3	51	14328.4	12048.1
3cyt	104	38744.7	38657.7
3rn3	125	22695.4	22433.7
4pti	58	11323.0	33776.8
5mbn	154	29433.7	29235.9
1arb	263	42314.7	42185.8
1cau	424	70369.1	70364.5
1cse	345	54619.0	54862.1
1ecd	136	25059.1	25396.5
1icm	131	25826.5	25898.8
1mbd	153	29441.1	29460.4
1plc	99	19914.7	17578.9
1rro	108	20034.6	19906.3
1thm	279	44096.1	43768.0
1ycc	103	20697.2	20240.6
3sdh	292	53079.8	51992.1
4gcr	174	33441.2	33776.8
5p21	165	31329.3	31251.5

Table 4.7: Computed molecular surface (MS) volume (in $Å^3$) of selected proteins.

protein	# res	VOLBL	GEPOL
1eca	137	16764.4	17728.2
1nxb	62	6879.1	7385.1
1rbc	124	14477.1	15174.7
1rbe	124	14601.0	15356.6
1rbf	124	14509.1	14947.2
1rbg	124	14709.2	15181.6
1rbh	124	14671.4	15282.9
1rbi	124	14616.5	15193.6
2act	218	26112.1	25362.0
2cha	248	28016.9	29121.9
2lyz	129	15627.9	16214.8
2ptn	230	26331.3	27302.5
2rns	124	14520.9	14904.2
2sn3	51	9437.5	7087.9
3cyt	104	25287.6	26771.7
3rn3	125	14927.1	15861.3
4pti	58	6964.2	7391.6
5mbn	154	19644.4	21147.4
1arb	263	30642.1	31445.2
1cau	424	46485.3	49967.3
1cse	345	39559.2	40773.6
1ecd	136	16733.4	17473.7
1icm	131	16952.0	18362.4
1mbd	153	19586.2	20773.0
1plc	99	13985.0	11957.9
1rro	108	13222.6	13932.9
1thm	279	32477.9	33996.8
1ycc	103	13231.7	13994.2
3sdh	292	36669.7	39286.8
4gcr	174	23037.3	24034.2
5p21	165	21262.8	22523.7

obtained with GEPOL and Richards' VOLUME. VOLBL and GEPOL compute volume in all three models, VS, SA and MS, while Richards's VOLUME as distributed in VADAR only computes VS volume.

5 Discussion

We see from tables 4.2 through 4.4 that VS, SA and MS area computed from VOLBL are comparable to areas computed by other analytical methods such as ANAREA and MS/MSDOT. Residuewise area contributions computed by VOLBL and ANAREA as depicted in figures 4.1, 4.2 and 4.3 also agree in general. On the other hand, VS volumes computed with VOLBL and VOLUME differ significantly, with the latter giving volumes always larger by 20 to 40%, see table 4.5. VOLUME is one of the earliest programs developed for molecular metric computation. It highlights the difference between the approximation and the analytical methods. Approximation methods are usually less accurate, because they do not rigorously deal with the complicated combinatorial problems of the molecular structures. To achieve fast speed, approximations are often made in these methods when errors are assumed to be tolerable, or when an accurate treatment is too involved or too costly. On the other hand, when fine grained discretization is used and longer running times are allowed, the accuracy of approximation methods can converge to that of analytical methods.

Among the analytical methods, the difference in computed SA area between VOLBL and Richmond's ANAREA, as well as the difference in computed VS and MS area between VOLBL and Connolly's MS/MSDOT are probably due to the (lack of) treatment of high order atom overlaps and different treatment of geometric degeneracy. We elaborate in the following two sections.

5.1 High order of atom overlap

Some of the available analytical methods ignore high order overlaps of atoms. For example, it is reported that Connolly's analytical algorithm and related methods ignore overlaps of 4 or more atoms (Petitjean, 1994). Petijean investigates this problem in detail. Using small molecules as examples, he discovers that the relative error is often around 30% for surface calculation and 5% for volume calculation when overlaps of four atoms and above are ignored (Petitjean, 1994). He also shows that in practical situations (e.g. aromatic compounds), overlaps of up to six atom balls occur frequently. Applications requiring high precision of surface area and volume computation should benefit from the use of VOLBL which treats overlaps accurately.

5.2 Degeneracy

A fundamental issue in practical geometric computing is robustness. Common sources for the lack of robustness are geometric primitive tests that are ambiguous in degenerate or near degenerate cases. For example, degeneracy occurs when 3 or more points are collinear, when 4 or more points are coplanar, or when 5 or more points are cospherical. The trouble starts when a primitive geometric test is applied to these points. Different outcomes of such a test lead the process into logically different branches of the program. An arbitrary decision would typically be acceptable if it is consistent with earlier ones. However, inconsistent decisions can lead the program into geometrically impossible states that cannot be resolved.

Here is a typical example. Consider an algorithm that needs to decide whether a point is to the right or to the left of a directed line passing through two other points. When the three points are collinear, the test is ambiguous, and suppose the outcome of the test is "left", maybe because of a slight bias caused by a small numerical error. When the same 3 points are encountered later, the test might assign an inconsistent "right" value, possibly because the points are presented to the test in a different order. This inconsistency is likely to ultimately crash the program. In principle, this problem cannot be fixed with improved numerical precision, since collinear points will always be collinear. What is needed is an arbitrary yet consistent decision.

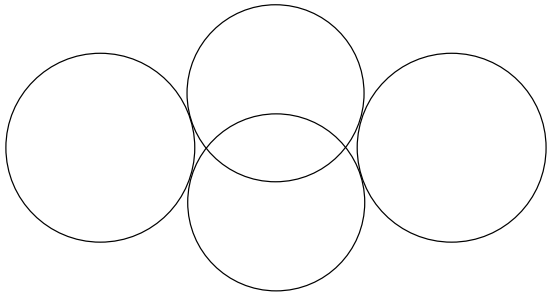

Figure 5.1: Cusp in the MS model. THIS FIGURE NEEDS TO BE IMPROVED. IT IS NOT CLEAR WHAT THE MS MODEL IS IN THIS CASE.

A popular method battling the problem of robustness is the (actual) perturbations of atom coordinates (Perrot *et al.*, 1992; Eisenhaber & Argos, 1993) and/or atom radii (Connolly, 1993). The hope is that a small perturbation will remove all degeneracies in the data. The drawback of such perturbations is they do not always work, and if they work, they do change the input and thus the output. Alternatively, one could write tests that unambiguously detect and classify degenerate cases. Such tests would have to rely on exact rather than floating-point arithmetic. This method leads to a large number of special cases, which have to be handled individually in a consistent manner. In the case of spherical balls in 3 dimensions this case analysis is likely to result in a nightmarish programming experience.

In the computational geometry community, several methods have been suggested to cope with geometric degeneracy (Edelsbrunner & Mücke, 1990; Yap, 1990; Emeris & Canny, 1991). The method of choice in our implementation is the symbolic perturbation of coordinates and radii. This is referred to as SoS (for "simulation of simplicity") and described in detail in (Edelsbrunner & Mücke, 1990). SoS symbolically perturbs coordinates and radii and systematically treats all special cases by a consistent reduction to the general case. Observe that only the construction of the Delaunay complex and the filter are prone to instable behavior if presented with inaccurate numerical computation. Floating-point operations in VOLBL are solely used to compute area and volume and not to derive any decisions impacting the flow of control. The above discussion of robustness and perturbation thus does not apply to VOLBL.

5.3 Riemannian interpretation of the MS model

Combinatorially, the SA and the MS model of a molecule are equivalent since they both are defined by the same solvent sphere rolling about the VS model. Indeed, they are both represented by the same alpha complex. Nevertheless, the area and volume associated with the two models may differ significantly. For example, the area and volume changes an octanol molecule experiences during a rotation can be different by 50% between the two models (Pascual-Ahuir & Silla, 1990). Motivated by these differences, it has been suggested that the molecular surface models certain physical/chemical phenomena more accurately than the solvent accessible surface (Jackson & Sternberg, 1993).

Unfortunately, the MS model suffers from an inherent deficiency. Surface smoothness lacking in the SA model was probably the motivation for the development of the MS model. However, when two probe spheres overlap in the process of reaching into a narrow tunnel from two sides, they form cusps which are local violations of smoothness, see figure 5.1. As a result, the overall smoothness of MS model is destroyed. To rid this dilemma, we propose to ignore cusps altogether. In other words, we consider the MS model as an emersed surface with self-intersections. Computing the volume still makes sense because the surface is orientable and locally inside/outside can unambiguously be defined. The surface bounds a 3-dimensional body, although the emersion of this body has flaps of self-overlap. It is like a piece of perfectly straight 3-dimensional space being placed into 4 dimensions: there can be flaps that lie side by side and overlap in their 3-dimensional projection. As an analogy, consider an

icosahedron with boundary made of paper. Cut the icosahedron open and lay it out in the plane. Depending on the choice of cuts the unrolled surface may overlap itself. These overlaps can be ignored since the refolding produces a perfect 2-dimensional manifold. This is why we call the cusp-free version of the MS model its *Riemannian* interpretation.

5.4 Removing self-overlap

JIE, SHOULD WE REALLY DISCUSS THESE OPERATIONS CONTRADICTION OUR PROPOSAL OF THE RIEMANNIAN INTERPRETATION? For area and volume computation cusps can be corrected by subtracting the self-intersecting part from the calculation (Connolly, 1985a; Connolly, 1993), if so desired. This is easy when there are only two atoms and gets quite difficult when 3 or more atoms are involved. The correction in area and volume is likely to be rather small. JIE, SHOULD WE DO THIS IN VOLBL IN SPITE OF OUR PROPOSAL?: In VOLBL, cusp correction is provided as an option which only considers the two atom cases.

Cusp-correction is problematic. We give a qualitative argument. For an MS model of a molecule, its surface is defined depending on the solvent: the surface is what the front of a solvent experiences while rolling about the molecule. A large solvent experiences a different surface than a small solvent. Since the MS surface is experienced by the solvent in a time averaged sense, we consider that a solvent approaching atoms from one side will not stay at the cusp regions for an extended period of time to exclude another solvent approaching the same region from the other side. The exclusion effect is present only for a fraction of the total time, and can be represented by a small probability. Cusp-correction would raise this small probability to certainty. As a result, cusp-correction may turn out to make the approximation of reality worse rather than better.

There is also a practical reason against cusp-correction. In the continuum approach for solvation and electrostatic studies, a protein is represented as a cavity of low dielectric constants in a high dielectric constant medium. The Poisson-Boltzmann equation is solved for electrostatic potentials (Sharp & Honig, 1990; Madura *et al.*, 1994; Sharp, 1994). For boundary value type of differential equations, such as the Poisson-Boltzmann equation, boundary conditions are an integral part of the problem directly influencing the computed electrostatic potential. Smoothness of the boundary is locally violated at the cusps and it is necessary to introduce extra conditions considering the cusp as a 1-dimensional boundary of a piece of the surface. In analysis, it is possible to derive these extra boundary conditions, e.g. from calculus of variation. The problem becomes more complicated in actual computation and none of the currently popular methods considers any such extra conditions.

JIE, I AM NOT SURE THERE IS A SATISFYING DEFINITION OF "CONTRIBUTION" IN THE MS MODEL. CAN WE REALLY CLAIM WE COMPUTED SOMETHING REASONABLE HERE? FOR NOW, I REMOVED THE DISCUSSION IN THE PAPER.

6 Conclusion

This paper describes an alpha shape based algorithm for computing molecular surface area and volume for both SA and MS models. Its implementation as the program VOLBL is described in some detail in the unpublished report (Edelsbrunner & Fu, 1994), see also (Edelsbrunner *et al.*, 1995). It belongs to the category of analytical methods and is combinatorial in nature, within the tradition of computational geometry (Preparata & Shamos, 1985; Edelsbrunner, 1987). Unlike previous analytical approaches, all computations are based on the dual topological structure of the molecule, which guarantees the combinatorial correctness of the computation. For example, it does not neglect cases when more than 4 atoms have a common intersection, although area and volume computations of such high order intersections are not actually performed. Efficient computation is achieved by precisely identifying atoms on the surface and their topological structure. Area and volume can be computed using the short inclusion-exclusion formula whose terms involve at most 3 intersecting balls at a time. The preparing computations by DELCX and MKALF employ a symbolic perturbation (Edelsbrunner & Mücke, 1990) of the geometric data. As a consequence, our software does not suffer from a lack of robustness caused by degenerate data. We have applied the

method to several proteins to demonstrate both the validity and the robustness of the alpha shape based method. In a companion paper, we continue the effort by describing the computation of voids and interfacial contacts in protein using the alpha shape method.

Acknowledgment

We thank Nataraj, Michael Facello, Patrick Moran and Marcus Wagner for interesting discussion on the topic of this paper and for help in the generation of 2- and 3-dimensional illustrations.

References

Ahmed, F. (1993). *J. Mol. Biol.* **230**, 1216–1224.

Alden, C. & Kim, S.-H. (1979). *J. Mol. Biol.* **132**, 411–434.

Baker, E. & Dodson, E. (1980). *Acta Crystallogr.,sect, A,* **36**, 559–572.

Birktoft, J. & Blow, D. (1972). *J. Mol. Biol.* **68**, 187–240.

Bode, W., Papamokos, E., & Musil, D. (1987). *Eur. J. Biochem.* **166**, 673–692.

Chotia, C. (1975). *J. Mol. Biol.* **96**, 721–732.

Connolly, M. (1985a). *J. Appl. Cryst.* **18**, 499–505.

Connolly, M. (1985b). *J. Am. Chem. Soc.* **107**, 1118–1124.

Connolly, M. (1986). *Biopolymers,* **25**, 1229–1247.

Connolly, M. (1993). *J. Mol. Graphics,* **11**, 139–141.

Connolly, T. (1983). *J. Appl. Cryst.* **16**, 548–558.

David, E. & David, C. (1982). *J. Chem. Phys.* **76**, 4611–4614.

Delaunay, B. (1934). *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk,* **7**, 793–800. In French.

Diamond, R. (1974). *J. Mol. Biol.* **82**, 371–391.

Eads, J., Sacchettini, J., A.Kromminga, & Gordon, J. I. (1993). *J. Biol. Chem.* **268**, 26375–26385.

Edelsbrunner, H. (1987). *Algorithms in combinatorial geometry.* Berlin: Springer-Verlag.

Edelsbrunner, H. (1993). In: *Proc. 9th Ann. Sympos. Comput. Geom.* pp. 218–231, New York: ACM Press.

Edelsbrunner, H., Facello, M., Fu, P., & Liang, J. (1995). In: *Proc. 28th Ann. Hawaii Int'l Conf. System Sciences* volume 5 pp. 256–264, Los Alamitos, California: IEEE Computer Scociety Press.

Edelsbrunner, H. & Fu, P. (1994). *Rept. UIUC-BI-MB-94-01, Molecular Biophysics Group, Beckman Inst. Univ. Illinois, Urbana, IL,* .

Edelsbrunner, H., Fu, P., Liang, J., Subramaniam, S., & Sudhakar, P. (1995). *In preparation,* .

Edelsbrunner, H., Kirkpatrick, D., & Seidel, R. (1983). *IEEE Trans. Inf. Theor. IT-29,* **4**, 551–559.

Edelsbrunner, H. & Mücke, E. (1990). *ACM Trans. Graph.* **9**, 66–104.

Edelsbrunner, H. & Mücke, E. (1994). *ACM Trans. Graphics,* **13**, 43–72.

Edelsbrunner, H. & Shah, N. (1992). In: *Proc. 8th Ann. Sympos. Comput. Geom.* pp. 43–52, New York: ACM Press.

Eisenhaber, F. & Argos, P. (1993). *J. Comput. Chem.* **14**, 1272–1280.

Emeris, I. & Canny, J. (1991). In: *Proc. 32nd Ann IEEE Sympos. Found. Comput. Sci.* pp. 405–413, Los Alamitos, California: IEEE Computer Society Press.

Finney, J. (1975). *J. Mol. Biol.* **96**, 721–732.

Freyberg, B., Richmond, T., & Braun, W. (1993). *J. Mol. Biol.* **233**, 275–292.

Gellatly, B. & Finney, J. (1982). *J. Mol. Biol.* **161**, 305–322.

Gibson, K. & Scheraga, H. (1987). *Mol. Phys.* **62**, 1247–1265.

Gibson, K. & Scheraga, H. (1988). *Mol. Phys.* **64**, 641–644.

Grand, S. L. & Jr., K. M. (1993). *J. Comput. Chem.* **14**, 349–352.

Guss, J., Bartunik, H., & Freeman, H. (1992). *Acta Crystallogr., Sect.B,* **48**, 790–811.

Hoffmann, C., Hopcroft, J., & Karasick, M. (1988). In: *Proc. 4th Ann. Sympos. Comput. Geom.* pp. 106–117, New York: ACM Press.

Howlin, B., Moss, D., & Harris, G. (1989). *Acta Crystallogr., Sect. A,* **45**, 851–861.

Jackson, R. & Sternberg, M. (1993). *Nature,* **366**, 638–638.

Joe, B. (1991). *Comput. Aided Geom. Des.* **8**, 123–142.

Kauzmann, W. (1959). *Advan. Protein Chem.* **16**, 1–63.

Kim, E., Varadarajan, R., & Wyckoff, H. (1992). *Biochemistry,* **31**, 12304–314.

Ko, T.-P., Ng, J. D., Day, J., Greenwood, A., & McPherson, A. (1993). *Plant Physiol.* **101**, 729–744.

Kratky, K. (1981). *J. Stat. Phys.* **25**, 619–634.

Kundrot, C., Ponder, J., & Richards, F. (1991). *J. Comp. Chem.* **12**, 402–409.

Langmuir, I. (1925). *Third Colloid Symposium Monograph.* New York: Chemical Catalog Co.

Lawson, C. (1977). In: *Mathematical software III* pp. 161–194. Academic Press New York.

Lee, B. & Richards, F. M. (1971). *J. M. Biol.* **55**, 379–400.

Louie, G. & Brayer, G. (1990). *J. Mol. Biol.* **214**, 527–55.

Madura, J., Davis, M., Gilson, M., Wade, R., Luty, B., & McCammon, J. (1994). In: *Reviews in Computational Chemistry*, (Lipkowitz, K. & Boyd, D., eds) volume V chapter 4, pp. 229–267. VCH Publishers New York.

Marquart, M., Walter, J., Deisenhofer, J., Bode, W., & Huber, R. (1983). *Acta Crystallogr., Sect. B,* **39**, 480–490.

Muller, J. (1983). *J. Appl. Cryst.* **16**, 74–82.

Munkres, J. (1984). *Elements of algebraic topology.* Redwood City, California: Addison-Wesley.

Najmudin, S., Nalini, V., Driessen, H., Blundell, T., Moss, D., & Lindley, P. (a). , .

Pacios, L. (1993). *Quantum Chemistry Program Exchange,* **132**.

Pai, E., Krengel, U., & Petsko, G. (1990). *EMBO J.* **9**, 2351–9.

Pascual-Ahhuir, J., Silla, E., & Tuñon, I. (1994). *J. Comp. Chem.* **15**, 1127–1138.

Pascual-Ahuir, J. & Silla, E. (1990). *J. Comput. Chem.* **11**, 1047–1060.

Pavlov, M. & Fedorov, B. (1983). *Biopolymers,* **22**, 1507–1522.

Perrot, G., Cheng, B., Gilson, K., Palmer, K., Nayeem, A., Maigret, B., & Scheraga, H. (1992). *J. Comp. Chem.* **13**, 1–11.

Petitjean, M. (1994). *J. Comput. Chem.* **15**, 507–523.

Phillips, S. V. & Schoenborn, B. (1981). *Nature,* **292**, 81–82.

Preparata, F. & Shamos, M. (1985). *Computational geometry: an introduction.* New York: Springer-Verlag.

Rajan, V. (1994). *Discret Comput. Geom.* **12**, 189–202.

Rashin, A., Iofin, M., & Honig, B. (1986). *Biochemistry,* **25**, 3619–3625.

Richards, F. (1974). *J. Mol. Biol.* **82**, 1–14.

Richards, F. M. (1977). *Ann. Rev. Biophys. Bioeng.* **6**, 151–176.

Richards, F. M. (1985). *Methods in Enzymology,* **115**, 440–464.

Richmond, T. (1984). *J. Mol. Biol.* **178**, 63–89.

Richmond, T. & Richards, F. M. (1978). *J. Mol. Biol.* **119**, 537–555.

Royer-Jr., W. (1994). *J. Mol. Biol.* **235**, 657–681.

Sharp, K. (1994). *Curr. Opinion Struct. Biol.* **4**, 234–239.

Sharp, K. & Honig, B. (1990). *Annu. Rev. Biophys. Biophys. Chem.* **19**, 301–332.

Sharp, K., Nicholls, A., Fine, R., & Honig, B. (1990). *Science,* **252**, 106–109.

Sharp, K., Nicholls, A., Friedman, R., , & Honig, B. (1991). *Biochemistry,* **30**, 9686–9697.

Shrake, A. & Rupley, J. (1973). *J. Mol. Biol.* **79**, 351–371.

Silla, E., Tuñon, I., & Pascual-Ahhuir, J. (1991). *J. Comp. Chem.* **12**, 1077–1088.

Steigemann, W. & Weber, E. (1979). *J. Mol. Biol.* **127**, 309–338.

Takano, T. (1984). *Meth. and Applic. Crystallogr. Comput.* , 262–272.

Takano, T. & Dickerson, R. (1980). *Proc. Nat. Acad. Sci. USA,* **77**, 6371–6375.

Teplyakov, A., Kuranova, I., & Harutyunyan, E. (1990). *J. Mol. Biol.* **214**, 261–79.

Tsernoglou, D. (1978). *Mol. Pharmacol.* **14**, 710–716.

Tsunasawa, S., Masaki, T., & Hirose, M. (1989). *J. Biol. Chem.* **264**, 3832–3839.

Varadarajan, R. & Richards, F. (1992). *Biochemistry,* **31**, 12315–12327.

Varshney, A., Brooks, F., & Wright, W. (1994). *IEEE Comput. Graphics Applications,* **14**, 19–25.

Voronoi, G. (1907). *Journal für die Reine und Angewandte Mathematik,* **133**, 97–178. In French.

Walter, J., Steigemann, W., Singh, T., Bartunik, H., Bode, W., & Huber, R. (1982). *Acta Crystallogr., Sect. B,* **38**, 1462–1472.

Wang, H. & Levinthal, C. (1991). *J. Comput. Chem.* **12**, 868–871.

Wishart, D., L. Willard, F. R., & Sykes, B. (1994). *University of Alberta,* **0.9**.

Wodak, S. & Janin, J. (1980). *Proc. Natl. Acad. Sci. USA,* **77**, 1736–1740.

Yap, C. (1990). *J. Symbolic. Comput.* **10**, 349–370.

Zhao, B. (1992). *J. Mol. Biol.* **227**, 239–52.