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

Molecular Surface Triangulation

BY MICHAEL L. CONNOLLY

Department of Molecular Biology, Research Institute of Scripps Clinic, La Jolla, CA 92037, USA

(*Received* 15 *May* 1985; *accepted* 3 *July* 1985)

Abstract

A method is presented for triangulating the surface of a molecule. A triangulated surface is a polyhedron, all of whose faces are triangles. The triangles are created by subdividing the curved faces of an analytical molecular surface that has been precalculated by an earlier algorithm. The triangulated surface has many applications. Molecular areas and volumes may be calculated from it. Packing defects in proteins may be identified. It may be used to determine whether a particular water molecule lies in the interior of the protein or on the surface. The triangulated surface may be drawn on pen plotters, vector graphics systems and raster graphics terminals. Mathematical functions defined on the surface may be contoured. Local maxima and minima of functions may be located. While the triangulated surface is less accurate than the analytical molecular surface it is derived from, it has the advantage of being much simpler and easier to deal with. It combines the simplicity of a dot surface with the continuity of an analytical surface.

Introduction

A variety of methods have been developed for calculating and representing the molecular surface defined by Richards (1977). These include a grid algorithm (Greer & Bush, 1978), dot surface algorithms (Connolly, 1981; Pearl & Honneger, 1983), cube algorithms (Müller, 1983; Pavlov & Federov, 1983) and an analytical algorithm (Connolly, 1983*a*). The Voronoi construction (Richards, 1974; Finney, 1978) gives a rough polyhedral approximation to the molecular surface. Each of these algorithms has advantages and disadvantages. The present work describes a new polyhedral representation of a molecular surface. The faces of the polyhedron are all triangles, and the triangulated surface may be made to approximate the exact analytical surface arbitrarily closely by using smaller and smaller triangles.

Curved surface precursor

The triangulated surface is not calculated directly from atomic coordinates, but rather from a curved surface precursor. This curved surface is made up of pieces of spheres and tori that join at arcs of circles. It is calculated from atomic coordinates by means of an analytical molecular surface algorithm that has been previously described (Connolly, 1983*a*, *b*).

Before this curved surface is triangulated, it is processed in two ways. Firstly, the data structure used to describe the surface is changed from one that represents convex spherical, saddle-shaped toroidal and concave spherical faces in different ways to one where all faces are described in a similar manner. This enables a single algorithm to triangulate any of these shapes. Each face of the generalized curved surface data structure is defined by the algebraic variety (Gellert, Küstner, Hellwich & Kästner, 1975) it lies on (sphere or torus) and by a set of cycles of edges that form its boundary. Each cycle consists of a linked list of edges, with each edge identified by number. Since each edge helps form the boundary of two faces, each edge shows up in two cycles, once with a positive sign, and once with a negative sign. The negative sign indicates that the edge is to be traversed in the opposite direction. Each edge is defined by the circle it lies on and the starting and ending vertices.

Secondly, certain pathological situations of self-intersecting surfaces that commonly show up in deep grooves and cavities must be handled. It is necessary to trim away some regions of surface faces passing through other faces or themselves. This leaves cusps, places where the surface has a sharp edge or point (Fig. 1). This cusp trimming is very involved and is described in detail below. The equations for calculating the variables of the analytical molecular surface are given by Connolly (1983*a*). Familiarity with this earlier publication is assumed in the algorithm descriptions below. The descriptions below give only the equations for deriving the cusp variables in terms of the original surface variables: $\mathbf{a}_i$ (center of atom i), $\mathbf{u}_{ij}$ (unit vector from atom i to atom j), r_p (probe radius), $\mathbf{t}_{ij}$ (torus center), r_{ij} (torus radius), $\mathbf{p}$ (probe center).

Cusp trimming

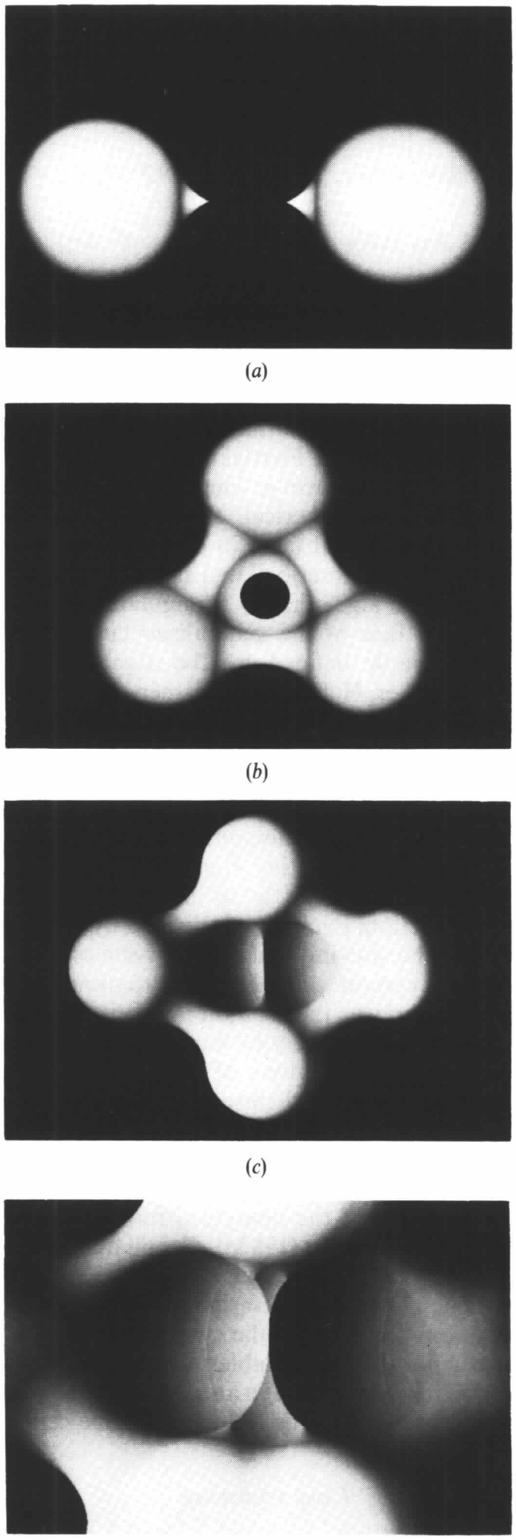
The simplest type of cusp consists of two sharp points (Fig. 1*a*). It occurs when two atoms are close enough so that they can be bridged by a probe sphere, but far enough apart so that the probe sphere intersects the line passing through the atom centers [Fig. 2(*a*); see also Fig. 6 of Connolly (1985*a*)]. Let us denote the coordinates of these cusp points by $\mathbf{q}_{ij}$ and $\mathbf{q}_{ji}$, and the distance from the torus center to a cusp point by d_{tq} (Fig. 2*b*). We may say:

$$d_{tq} = (r_p^2 - r_{ij}^2)^{1/2}$$

$$\mathbf{q}_{ij} = \mathbf{t}_{ij} - d_{tq}\mathbf{u}_{ij}$$

$$\mathbf{q}_{ji} = \mathbf{t}_{ij} + d_{tq}\mathbf{u}_{ij}.$$

Consider the situation where a probe sphere tangent to three atoms intersects the plane passing through the centers of the atoms. If there is no probe on the opposite side of the three atoms, then there will be no cusp, just a deep concavity. If there is an opposing probe, then the two probes will intersect at a cusp circle (Fig. 1*b*). Let us denote the distance between the probe centers by d_{pp}, the midpoint of the line joining



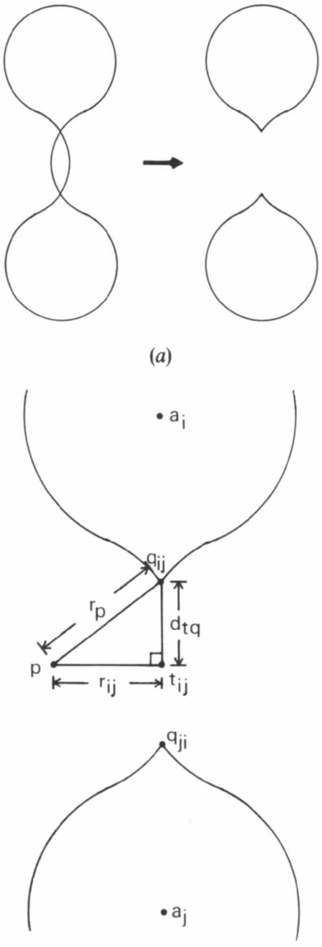
(*a*)

(*b*)

(*c*)

(*d*)

Fig. 1. Cusps of the solvent-accessible surface displayed on an AED 512 raster graphics terminal using the *RAMS* program (Connolly, 1985*b*). The cusps are at the central regions of the photographs. (*a*) Two atoms. (*b*) Three atoms. (*c*) Four atoms. (*d*) Five atoms.



(*a*)

(*b*)

Fig. 2. Cross section of point cusps. (*a*) The lens-shaped surface in the center does not form part of the boundary of the solvent-excluded volume, so it must be removed, leaving point cusps. (*b*) Application of the law of Pythagoras to determine the locations of the cusp points. The torus radius is less than the probe radius.

the two probe centers by **m**, and the radius of the cusp circle by r_q (Fig. 3). We may say:

$$d_{pp} = |\mathbf{p}_1 - \mathbf{p}_2|$$
$$\mathbf{m} = (\mathbf{p}_1 + \mathbf{p}_2)/2$$
$$r_q = (r_p^2 - d_{pp}^2/4)^{1/2}$$

This cusp circle forms a new edge that joins the two concave faces. Each concave face is now described by two cycles, one with three concave edges and one with one cusp edge.

It is possible for cusp points and a cusp edge to occur in combination (Fig. 1*c*). In this situation, a cusp edge consists of not a whole circle, but instead just part of a circle, and it is delimited by two cusp points. This is actually the most common type of cusp found on protein surfaces. In Fig. 4 it is shown how the data structure of the surface is modified for combination cusps.

The worst difficulty in dealing with cusps is the fact that cusp edges sometimes intersect each other. There are two kinds of intersection: axial and non-axial. Axial intersections occur when the probes creating the cusps are all tangent to a particular pair of atoms. In this situation all probe spheres lie inside the torus defined by a probe rolling around this pair of atoms, and all the cusp edges end at the same pair of points, the cusp points on the interatomic (torus) axis. Each cusp edge 'eats' a lesser or greater amount of the area

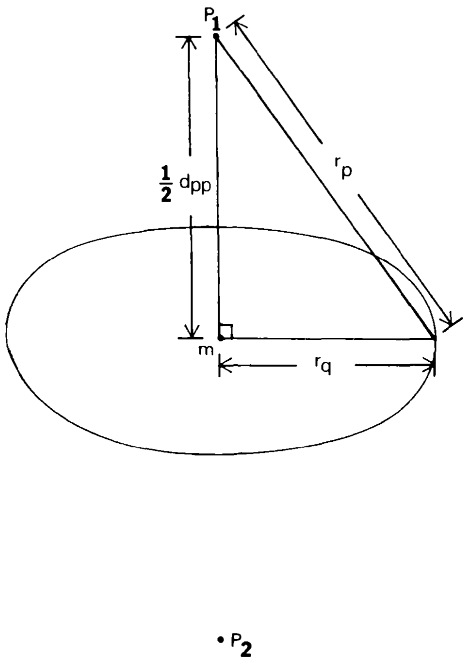


Fig. 3. Calculation of cusp circle radius. The two probe centers are separated by less than the sum of their radii, so they intersect, generating the cusp circle.

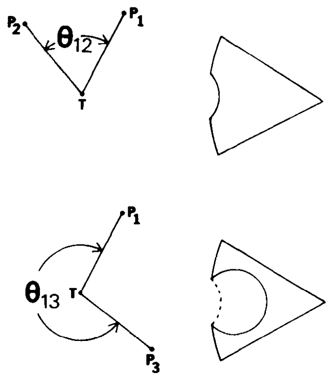


Fig. 5. Axial cusp intersections from two orthogonal viewpoints. Left: view down torus axis. Right: concave face of probe 1 with bites taken out by probe 2 (top) and probe 3 (bottom). The vector from the torus center to the center of probe 1 is compared to two other probe positions. Only the cusp edge resulting from the intersection of the concave faces associated with probes 1 and 3 is kept, because $\theta_{13} > \theta_{12}$ implies that probe 3 contains in its interior the cusp edge of probe 1 with probe 2 (dashed). This implication can be visualized by considering a continuum of second probe positions, starting with a position coincident with the first probe and rotating the second probe counter-clockwise around the torus axis. As the second probe rotates, so does the cusp circle of intersection of the two probes. As θ increases, the cusp circle eats more and more into the concave face associated with the first probe. Axial cusp intersections typically occur at mouths of surface invaginations or connections between internal cavities that are a little too narrow to allow passage of the probe sphere.

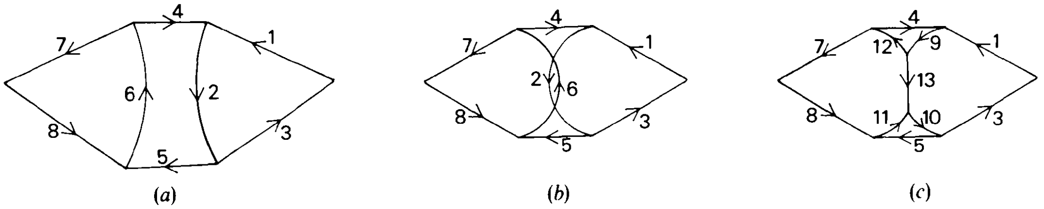


Fig. 4. Combination cusp. (*a*) Edge diagram of normal saddle rectangle flanked by two concave triangles. The cycles of edges for the concave faces are: $e_1e_2e_3$ and $e_6e_7e_8$. The cycle for the saddle face is $\bar{e}_2\bar{e}_4\bar{e}_6\bar{e}_5$, where the overbar represents traversal in the direction opposite to the arrow in the diagram. (*b*) Situation where two concave edges (e_2 and e_6) intersect, generating a combination cusp. (*c*) The corrected edge diagram. Edges e_2 and e_6 have been replaced by edges e_9, e_{10}, e_{11}, e_{12} and e_{13}. The concave triangle cycles have been changed to: $e_1e_9e_{13}e_{10}e_3$ and $e_{11}\bar{e}_{13}e_{12}e_7e_8$. The saddle rectangle has been split into two saddle triangles: $\bar{e}_9\bar{e}_4\bar{e}_{12}$ and $\bar{e}_{11}\bar{e}_5\bar{e}_{10}$. Here and in Fig. 1(*c*) the cusp edge is seen edge on.

of each associated concave face. The edge that eats the most will eat all the other cusp edges on this axis, so they are discarded from the surface data structure. The method for determining which cusp edge is kept is shown in Fig. 5.

Non-axial cusp-edge intersections occur when the cusp edges belong to different tori. The edges intersect not at the original cusp points on the torus axes, but rather at new points (Fig. 1*d*). Instead of thinking of this situation as the intersection of two cusps, it may alternatively be thought of as the intersection of three concave triangles. The middle triangle is split into two concave faces: a quadrilateral and a pentagon (Fig. 6). The very rare situation of four concave triangles intersecting is handled in a similar fashion. So far, an intersection of five concave triangles has not been encountered.

This algorithm has been implemented in the *CT* (cusp trimmer) computer program. It reads the output of the *AMS* program (Connolly, 1983*a*), and writes a disk file of a curved surface in the new more general data structure, with cusps trimmed. That is, all intersecting faces have been replaced by faces that meet at cusps.

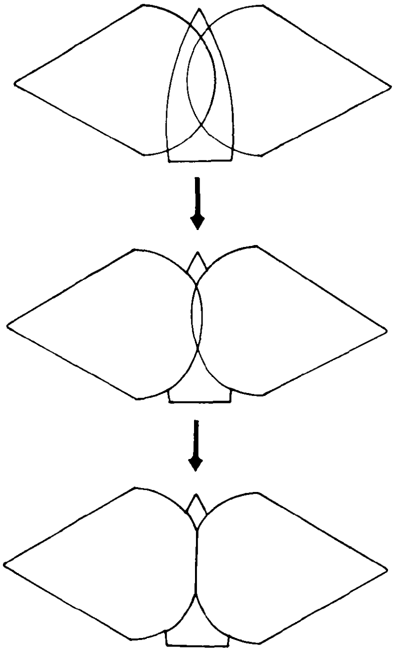


Fig. 6. Non-axial cusp intersections. Edge diagrams showing three intersecting concave triangles (top). The left and middle concave triangles intersect and the middle and right triangles intersect to form a pair of cusp edges (middle), which in turn intersect at two points joined by a cusp edge resulting from the intersection of the left and right concave triangles (bottom). The resulting surface can be viewed as part of the surface of the union of the three probe spheres. Non-axial cusp intersections typically occur in deep grooves on protein surfaces.

Triangulation

The output of the *CT* program is read by the *TS* (triangulated surface) program. After triangulating the surface, this program writes out a list of vertex coordinates and a list of triangles, each triangle being represented by three vertex numbers.

The basic idea behind the triangulation algorithm is recursive subdivision. Each face is divided into two faces. Each of these faces is, in turn, divided into two faces, and so on, until all faces have been replaced by a collection of triangles, each smaller than a specified size (Fig. 7).

The first step is to subdivide all the edges. The subdivision of an edge generates new edges and vertices. The number of subdivisions of an edge is chosen to be large enough so that each new edge is smaller than the maximum edge angle input parameter.

A face is subdivided by connecting two of its vertices by a new edge. Two of the criteria for choosing which pair of vertices to connect are that they should be close together and that the face should be divided into approximately equal halves. Let d_{ij} represent the distance (through space) from vertex i to vertex j. Let c_{ij} represent the distance from vertex i to vertex j going around the perimeter of the face in one direction, and let c_{ji} be the distance around the perimeter in the other direction (Fig. 8*a*). Let w_1 and w_2 be two input parameter weighting factors. Then i and j are chosen so that $w_1 d_{ij} + w_2 |c_{ij} - c_{ji}|$ is minimized, subject to the additional constraint that the two vertices lie on different arcs. The assignment of equal values of 0·5 to both weights has worked satisfactorily.

For spherical faces, the new edge is chosen to be an arc of a great circle, that is, a circle whose center coincides with the sphere center. The edge becomes part of two new cycles generated for the two new faces. If its angle is greater than the maximum edge angle parameter, it is subdivided.

When choosing a pair of vertices for saddle faces, there is an additional criterion, that only vertices at corresponding positions on opposite edges are connected (Fig. 8*b*). These are the only pairs of vertices

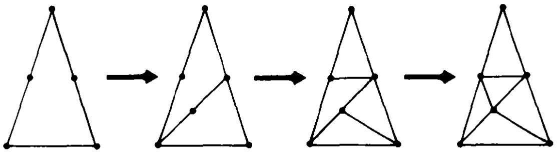


Fig. 7. Recursive face subdivision. First, all the edges longer that the maximum edge angle input parameter are subdivided. Then the face is bisected by a new edge. The new edge is subdivided, because it exceeds the maximum angle parameter. The new smaller faces continue to be recursively bisected, until all faces have three edges.

that can be joined by an arc of a circle; other vertex pairs could be connected only by arcs of fourth-degree curves. The requirement that only corresponding pairs of vertices be joined implies that opposing edges must have equal numbers of subdivisions. This is no problem for concave edges, but convex edges on atoms of different radii are not of equal length, although they always have equal angles. In order to ensure that there are equal numbers of subdivisions on the two convex arcs of a saddle rectangle, the subdivision input parameter is specified as an angle, rather than a distance. The subdivision of a saddle face produces a grid of small rectangles, each of which is subsequently divided into two triangles. The algorithm is modified slightly for saddle triangles and cones, which result from cusps (Fig. 1).

For spherical faces bounded by more than one cycle, the cycles are first connected together by new edges to form one large cycle (Fig. 9). This is done before the face is subdivided, but after its edges are subdivided.

The computer programs *CT* and *TS* are written in Fortran 77 for a VAX-11/750 operating under VMS. Scientists interested in obtaining copies of the programs should contact the author.

Area, volume and connectivity

Since all the faces are triangles, it is straightforward to calculate the area of the triangulated surface. The volume may also be calculated easily. An arbitrary point in space is chosen, for example, the centroid of the vertices. The volume of the polyhedron may be written as the sum of the volumes of a set of tetrahedra, where each tetrahedron joins a triangle of the surface to the central point.

The accuracy of the area and volume computed from the polyhedral surface may be determined by comparison with the analytical method (Connolly, 1983*a*, 1985*a*). For hen egg white lysozyme (Diamond, 1974), the analytical method gives an area of 5619 Å^2 and a volume of 16 243 Å^3. A surface triangulated with a maximum edge angle parameter of 1·2 rad has an area of 5371 Å^2 and a volume of 16 109 Å^3. A parameter of 0·6 rad gives an area of 5546 Å^2 and a volume of 16 226 Å^3. The protein coordinates used in this work were obtained from the protein data bank (Bernstein *et al.*, 1977), van der Waals radii were taken from McCammon, Wolynes & Karplus (1979), and a 1·5 Å radius probe sphere was used when calculating the surface.

Let us consider a related problem. Instead of looking at the volume of each tetrahedron, let us consider the solid angle of each tetrahedron at the central point. These solid angles may be calculated using spherical geometry (Millman & Parker, 1977). The sum of the solid angles will equal 4π if the central point is inside the polyhedron, and 0 if it is outside. This method is related to Gauss's law (Jackson, 1975) and is analogous to the concept of the winding number of a curve about a point (O'Neill, 1966). This solid angle computation gives us a simple automatic method for determining which water molecules of a crystallographic structure determination are buried inside the protein. Another application would be to fill space with small cubes and use this interiority algorithm to determine which cubes are inside the molecule. A set of cubes representing the solvent-excluded volume can be used to calculate X-ray scattering curves (Müller, 1983; Pavlov & Fedorov, 1983). There are analogous volume, interiority and cube algorithms based on dot surfaces, but they are less accurate, because of the inaccuracy of the areas associated with the surface points (Connolly, 1981).

The problem of finding packing defects or voids in protein interiors (Richards, 1979) can be approached as a problem in surface connectivity. Each packing defect is enclosed by a separate component of surface. Since connectivity is explicitly defined by the edges of the triangulated surface, the components are easily identified by algorithms for finding the connected components of graphs (Sedgewick, 1983). In contrast, the determination of the components of a protein dot surface requires an initial step of connecting together

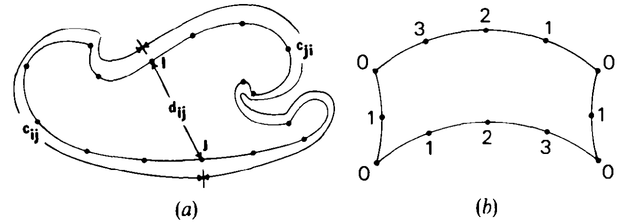


Fig. 8. Choosing a pair of vertices to connect with a new edge. (*a*) Basic criterion is based on direct and circumferential distances. (*b*) Saddle face subdivision. Vertices are numbered counterclockwise, starting at zero at each corner. A pair of vertices on two convex edges is chosen only if both numbers are non-zero and sum to the number of subdivisions of the convex edge. Concave edges are handled similarly.

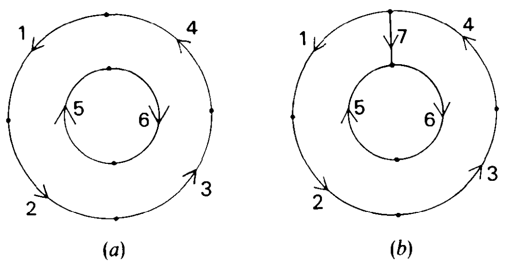


Fig. 9. Annular spherical face. (*a*) The face is originally bounded by two cycles: $e_1e_2e_3e_4$ and e_5e_6. (*b*) The addition of edge e_7 connecting the two cycles creates a single combined cycle: $e_1e_2e_3e_4e_7e_6e_5\bar{e}_7$. The overbar indicates that the edge is traversed in the reverse direction.

dots close in space and with normal vectors pointing in similar directions (Connolly, 1981).

Graphics and surface functions

Triangulated surfaces are particularly suited to plotting on paper, because the edges may be drawn as lines (Fig. 10). Since plotting hardware is less expensive than raster or vector graphics equipment, triangulated surfaces make the display of protein solvent-accessible surfaces more affordable.

The triangulated surface is an example of a polygon mesh, one of the most commonly used types of surface representation, for which many hidden-edge removal, hidden-surface removal and shading algorithms exist (Foley & Van Dam, 1982). In contrast, methods for hidden-surface removal and shading of solvent-accessible surfaces made up of parts of spheres and tori are rare and complex (Quarendon, 1984; Connolly, 1985*b*).

Of course, the edges of the surface may be displayed directly, without processing, on a vector graphics system with real-time rotation. For two interacting molecules, the connectivity of the triangulated surface enables the viewer to distinguish the lines belonging to different molecules, lessening the need for an expensive color calligraphic monitor, which was so useful in distinguishing dot surfaces (Langridge, Ferrin, Kuntz & Connolly, 1981).

Consider a real-valued mathematical function defined on the vertices of the surface. Such a function might quantitate some chemical or geometric property of the surface, such as shape or electrostatic potential. The edges provide a way of comparing the values of the function at neighboring vertices. The value of the function on the edges and triangles may be interpolated, so that surface-function contouring may be performed. On plotters and vector graphics equipment, contour lines can be determined by interpolating the function along the edges.

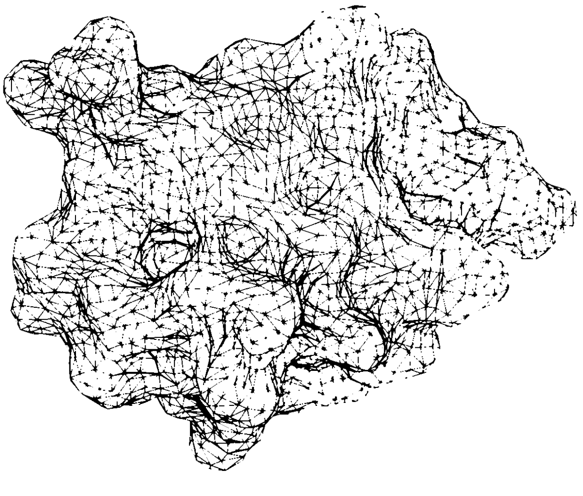


Fig. 10. Triangulated surface of insulin (Dodson, Dodson, Hodgkin & Reynolds, 1979) with hidden-edge removal.

For an example of contouring a function on a raster graphics system, consider the shape function defined in Fig. 11. This function measures the local convexity or concavity of the surface. For the insulin protein, it has been evaluated at each of the vertices of the surface, interpolated at each pixel, and each pixel has been colored according to the interpolated value (Fig. 12). An earlier method for coloring a polyhedral surface of a molecule according to a surface function (Nakamura, Kusunoki & Yasuoka, 1984) assigned a single homogeneous color to each polygon, rather than interpolating the function, pixel by pixel. Their method used a polyhedral approximation to the expanded accessible surface traced by the center of a probe sphere, rather than Richards's (1977) molecular surface. Methods for coloring dot surfaces on a vector graphics system according to electrostatic potential have been developed (Weiner, Langridge, Blaney, Schaefer & Kollman, 1982; Getzoff, Tainer, Weiner, Kollman, Richardson & Richardson, 1983). Procedures for calculating molecular electrostatic potentials on surface envelopes of nucleic acids and plotting the values of the surface function by means of grey levels have also been developed (Lavery & Pullman, 1981; Lavery, Pullman & Zakrzewska, 1982). The images produced by the molecular surface triangulation algorithm (Fig. 12) are perhaps more easily interpreted than these earlier results.

The local maxima, minima and saddle points of a surface function may be found by comparing the value of the function at each vertex to the values of the function at the adjacent vertices. Local minima and

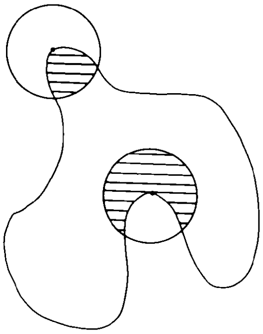


Fig. 11. Two-dimensional representation of shape function definition. The contour represents the protein surface and the circles represent spheres. The volume of the protein inside a sphere is shaded. The volume of the protein enclosed by a sphere centered at a point on the protein surface is large for concave regions and small for convex regions. The radius of the sphere determines the scale of the features that are identified. The volume value is assigned to the surface point at the sphere center. This defines a real-valued function that varies smoothly over the protein surface.

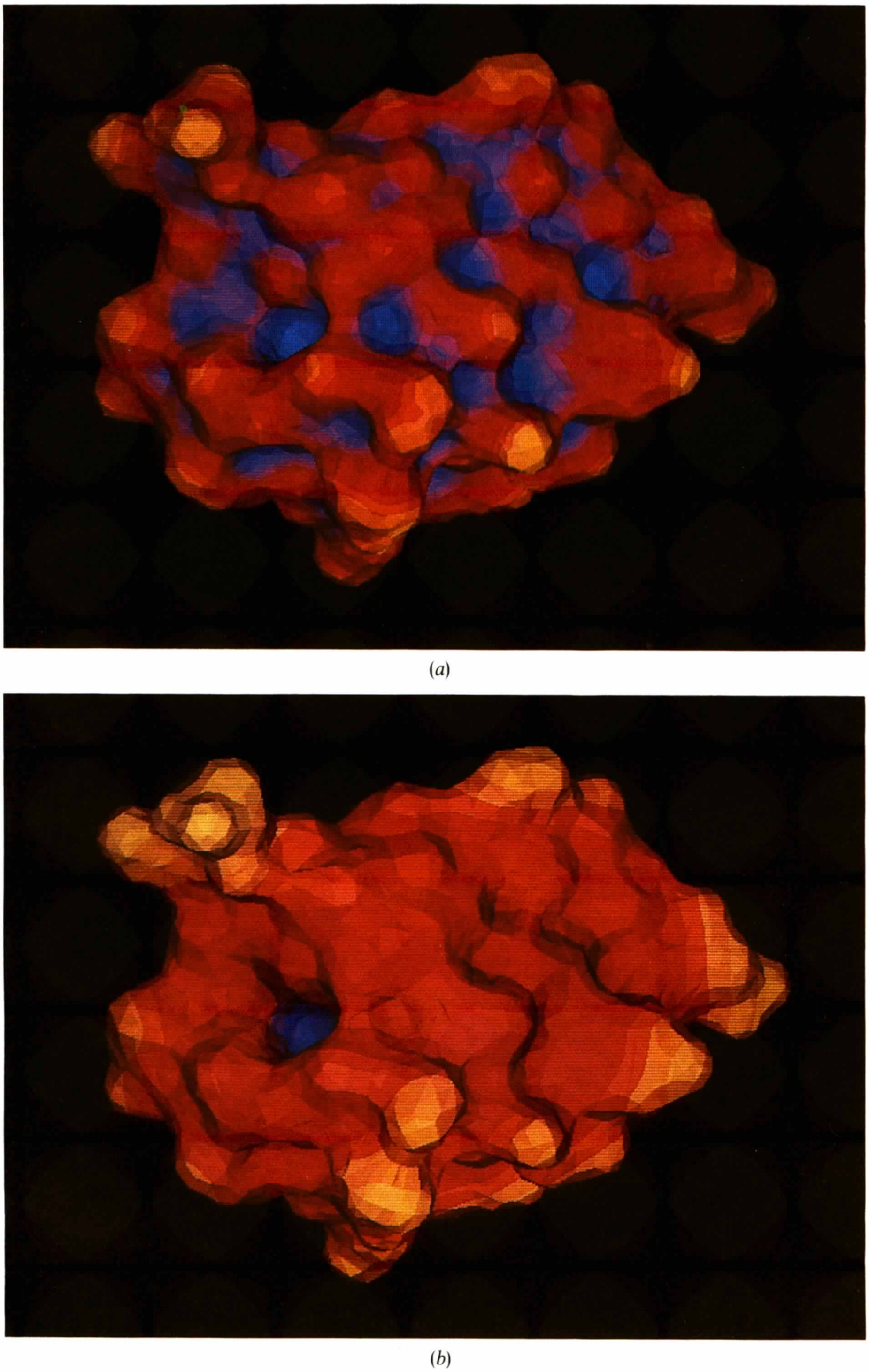


Fig. 12. Insulin surface color coded by shape. Same view as Fig. 10. Yellow: very convex; orange: moderately convex; red: approximately flat or saddle-shaped; magenta: moderately concave; blue: very concave. (*a*) 3 Å radius sphere. (*b*) 9 Å radius sphere. What is identified as a knob or hole is strongly dependent on the sphere radius.

maxima of the shape function correspond to knobs and holes, respectively, of the surface. This method for identifying topographical features of a protein surface may prove useful in the development of an algorithm for predicting protein–protein associations.

This work was supported, in part, by NIH grant GM 34338.

References

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Connolly, M. L. (1981). Thesis, Univ. of California at Berkeley.

Connolly, M. L. (1983*a*). *J. Appl. Cryst.* **16**, 548–558.

Connolly, M. L. (1983*b*). *Science*, **221**, 709–713.

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

Connolly, M. L. (1985*b*). *J. Mol. Graphics*, **3**, 19–24.

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

Dodson, E. J., Dodson, G. G., Hodgkin, D. C. & Reynolds, C. D. (1979). *Can. J. Biochem.*, **57**, 469–479.

Finney, J. L. (1978). *J. Mol. Biol.* **119**, 415–441.

Foley, J. D. & Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*, pp. 507–513, 553–570, 580–584. Reading, Massachusetts: Addison-Wesley.

Gellert, W., Küstner, H., Hellwich, M. & Kästner, H. (1975). *The Van Nostrand Reinhold Concise Encyclopedia of Mathematics*, pp. 673–675. New York: Van Nostrand Reinhold.

Getzoff, E. D., Tainer, J. A., Weiner, P. K., Kollman, P. A., Richardson, J. S. & Richardson, D. C. (1983). *Nature (London)*, **306**, 287–290.

Greer, J. & Bush, B. L. (1978). *Proc. Natl Acad. Sci. USA*, **75**, 303–307.

Jackson, J. D. (1975). *Classical Electrodynamics*, p. 32. New York: John Wiley.

Langridge, R., Ferrin, T. E., Kuntz, I. D. & Connolly, M. L. (1981). *Science*, **211**, 661–666.

Lavery, R. & Pullman, B. (1981). *Int. J. Quantum Chem.* **20**, 259–272.

Lavery, R., Pullman, B. & Zakrzewska, K. (1982). *Biophys. Chem.* **15**, 343–351.

McCammon, J. A., Wolynes, P. G. & Karplus, M. (1979). *Biochemistry*, **18**, 927–942.

Millman, R. S. & Parker, G. D. (1977). *Elements of Differential Geometry*, p. 262. Englewood Cliffs, New Jersey: Prentice-Hall.

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

Nakamura, H., Kusunoki, M. & Yasuoka, N. (1984). *J. Mol. Graphics*, **2**, 14–17.

O'Neill, B. (1966). *Elementary Differential Geometry*, p. 174. New York: Academic Press.

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

Pearl, L. H. & Honegger, A. (1983). *J. Mol. Graphics*, **1**, 9–12.

Quarendon, P. (1984). *J. Mol. Graphics*, **2**, 91–95.

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

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

Richards, F. M. (1979). *Carlsberg Res. Commun.* **44**, 47–63.

Sedgewick, R. (1983). *Algorithms*, pp. 381–386. Menlo Park, California: Addison-Wesley.

Weiner, P. K., Langridge, R., Blaney, J. M., Schaefer, R. & Kollman, P. A. (1982). *Proc. Natl Acad. Sci. USA*, **79**, 3754–3758.