Plotting protein surfaces

Michael L Connolly

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

This paper presents two methods for drawing solvent-accessible surfaces of proteins on a plotter. One method draws a stack of contours and the other draws a triangulated polyhedral surface. Both methods perform hidden-line elimination. These methods should be very useful in laboratories which do not possess vector or raster colour graphics equipment.

Keywords: protein, surface, plotting, contours, hidden-line elimination

received 17 September 1985, accepted 28 October 1985

The computer graphical display of the solvent-accessible surfaces of proteins is useful in understanding molecular interactions[1]. It is based on the ideas of Lee and Richards[2], Richards[3] and Greer and Bush[4]; they rolled a water-sized sphere over the molecule in order to define an outer surface envelope. These ideas have been implemented in an analytical fashion[5]. Methods for both the raster graphical[6,7] and vector graphical display of the analytical molecular surface have been developed. The dot surface numerical algorithm[8] is suited to vector graphical display[9], but dot surfaces are hard to interpret when plotted on paper. Methods for plotting van der Waals surfaces and sections are commonplace, but no good methods for plotting solvent-accessible molecular surfaces have been developed.

This paper presents two methods for plotting the solvent-accessible surface of a protein. The first method calculates intersection contours of the protein surface with a stack of planes. The second performs hidden-edge elimination on a triangulated polyhedral surface.

PLANAR CONTOURS

The product of this method is a set of contour lines representing the protein surface as seen from a particular viewpoint. To make the image more realistic, hidden-line elimination is performed. To keep the hidden-line elimination algorithm simple, the contours are calculated to lie in planes perpendicular to the z-axis, where the observer is located at positive values of z, and there is no perspective. The method consists then of two steps: (1) planar contour generation, and (2) hidden-line elimination. The names of the programs performing these two functions are, respectively, PC1 and PC2.

The PC1 program produces planar contours as output, given an analytical molecular surface disc file as input. The surface input file is created by passing the output of the AMS program[5] through the cusp trimmer (CT) program[10]. The spacing between the planes and a rotation matrix are given as input parameters to PC1.

Each planar contour is calculated by intersecting a plane with the surface. The surface is made up of pieces of spheres and tori, called faces, that join at circular arcs, called edges. An intersection contour is made up of the intersections of the plane with some of the faces of the surface. The intersection of the plane with a spherical face gives a circular arc. The intersection of the plane with a toroidal (saddle-shaped) face gives an arc of a fourth degree curve. These two types of intersection are handled differently. The mathematical equations of circle–plane and sphere–plane intersections will be published elsewhere as part of a more general intersection algorithm[11]. The saddle face intersections are discussed in more detail because they are more difficult to compute.

Before the intersection of a plane with any saddle face is calculated, the intersections of the plane with all spherical faces and all edges are calculated. Not just coordinates, but also tangent vectors are calculated at each edge–plane intersection point. The tangent vectors are tangent to the circular arcs that end at these points. Next, the saddle face intersections are calculated. Since each saddle face is bordered exclusively by spherical faces, every arc across a saddle face begins and ends at a place where an intersection point and tangent vector have already been calculated. Consider a particular arc on a saddle face (see Figure 1). Let the arc begin at a point with coordinates p_1 and tangent vector $\boldsymbol{t}_1$, and end at a point with coordinates p_2 and tangent vector $\boldsymbol{t}_2$. The arc of the fourth degree curve is approximated by an arc of a circle. This arc should join the two points and its tangent vectors at the endpoints should approximately match the two given (spherical face) tangent vectors. The arc is overdetermined, so the tangent vectors at the endpoints cannot be matched exactly. The centre and radius of the arc are computed as follows: Let $\boldsymbol{v} = (p_2 - p_1)/|p_2 - p_1|$ be a unit vector pointing from p_1 to p_2, and let $\boldsymbol{z}$ be a unit vector along the positive z axis. Let $\theta_1 = \arcsin(\boldsymbol{t}_1 \times \boldsymbol{v} \cdot \boldsymbol{z})$ be the angle between $\boldsymbol{t}_1$ and $\boldsymbol{v}$, and let $\theta_2 = \arcsin(\boldsymbol{v} \times \boldsymbol{t}_2 \cdot \boldsymbol{z})$ be the angle between $\boldsymbol{v}$ and $\boldsymbol{t}_2$. Let $\theta = (\theta_1 + \theta_2)/2$ be the average of these two angles. Let $d = |p_2 - p_1|/2$ and let $R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ be a 90° rotation matrix. Then the centre and radius of the arc are given by $c = (p_1 + p_2)/2 + d\cot\theta R\boldsymbol{v}$ and $r = d\operatorname{cosec}\theta$, respectively. Sometimes θ_1 and θ_2 will have opposite signs and the approximation scheme of Figure 1 will not be appropriate. In this case we join p_1 and p_2 by a straight line.

Sometimes a plane intersects a convex or concave face in a whole circle, but usually the circle is interrupted when the plane intersects some of the edges forming

0263-7855/86/020093-04 $03.00

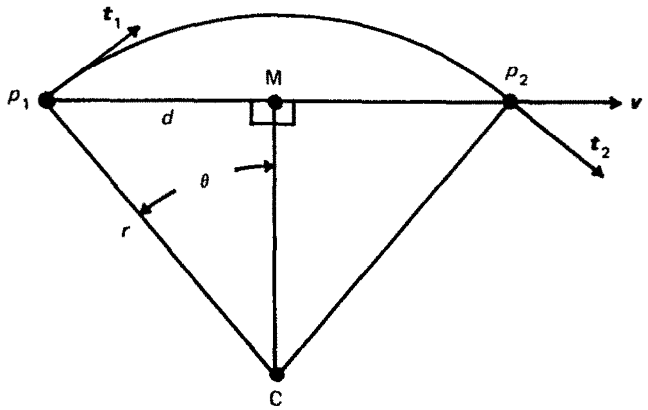

Figure 1. Calculation of arc approximating the intersection of a saddle face with the contour plane. The contour plane is the plane of the paper, and the saddle face is approximately perpendicular to the plane. The arc is determined from the tangent vectors of the preceding and succeeding arcs, which are on spherical faces. The midpoint between the two endpoints is denoted by M. *Ideally, the centre of the arc* c *would lie on three lines: the line perpendicular to* p_1p_2 *passing through* M, *the line perpendicular to* t_1 *passing through* p_1 *and the line perpendicular to* t_2 *passing through* p_2. *Since three lines do not generally intersect at a single point, the requirement of perpendicularity of the radial vectors to the tangent vectors has been relaxed. The angle between* t_1 *and* p_1C *is then approximately a right angle, so the angle between* t_1 *and* v *is the approximate complement of* $<Mp_1C$, *which, in turn, is the complement of* θ, *so* θ *should be approximately equal to the angle between* t_1 *and* v (θ_1). *Similarly,* θ *should be approximately equal to the angle between* v *and* t_2 (θ_2). *Therefore,* θ *is set equal to the average of* θ_1 *and* θ_2

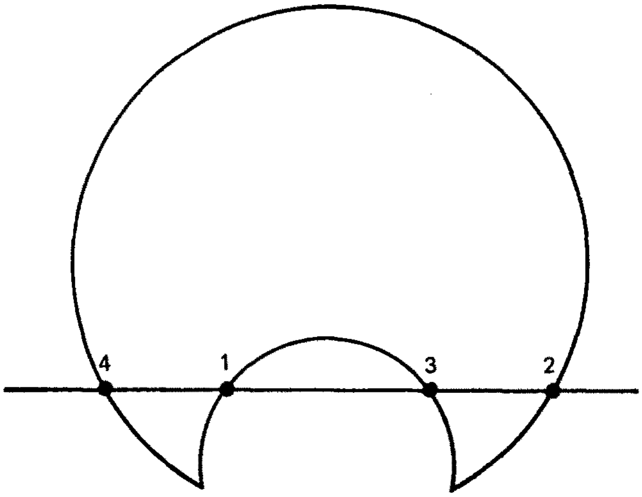

Figure 2. The plane (seen edge on) intersects the boundary of the face at four points. A pairing algorithm decides that 4 should be connected with 1, and 3 with 2

the border of the face. There are generally only two edge-plane intersection points per face, but sometimes there are more (see Figure 2). Clearly the number of such points must be even. If there are more than two points, they must be paired to form arcs. For spherical faces the pairing is carried out by sorting the points, by angle, along the circular arc. Adjacent points in the sorted list are connected. For saddle faces, however, the centres and radii of the arcs are not known until after the pairing. So for saddle faces an algorithm is used that pairs a point with the nearest point in the direction of its tangent vector.

The arcs of an intersection plane are linked together to form closed contours. Each contour can be described as a sequence of circular arcs and line segments. The contours are grouped to form the boundaries of regions using methods similar to those used for defining contact faces[5]. This is a necessary step because a region may be multiply connected, that is, be bounded by not only an outer contour, but also by one or more inner contours. All the regions of a given plane are collectively referred to as a section. These sections are written to a disc file.

In order to improve efficiency, a scan-line type of algorithm is used for choosing faces that may intersect the plane. The edges and faces of the surface are sorted by a z coordinate. There is a dynamic linked list of the edges that might intersect a particular plane, and a similar list for the faces. As the intersection planes are traversed from high to low z values, edges and faces leave and join the linked lists.

The contours produced by PC1 can be displayed directly on a vector graphics system with real-time rotation. For plotting purposes, a second program (PC2) reads the sections from disc and performs hidden-line elimination. This is done by starting at the top section and moving downward. First, the top section is drawn as it is; then the intersection of the top section with the contours of the next highest section is calculated. The z-coordinate is ignored; so that contours from different sections are treated as if they lie in the same plane. The goal is to determine the parts of the contours of the second section that lie outside the first section. These are the only lines in the second section that are visible.

The first step is to find an arc of the second section that lies outside all of the regions of the first section. If none is found, none of the contours of the second sections are drawn. An arc is tested for lying outside a region by calculating for each contour bounding the region its winding number[12] about the midpoint of the arc. If the winding number for the outer contour of the region is zero, the arc lies outside the region and is visible. Similarly, if the winding number of an inner contour is nonzero, the arc lies in a hole in the region and is visible. Arcs with midpoints near the boundary of the region are rejected because the arc might intersect the boundary. Once a visible arc is found, the formation of a new contour begins. This new contour will bound the union of the areas of the two sections (see Figure 3). The arcs of the second section are successively added until an arc is reached that is partially obscured by the first, higher section. If the two sections are considered to lie in the same plane, then this arc must intersect an arc of the first section. The two arcs that intersect are each subdivided into two shorter arcs, and the pre-intersection short arc of the second section is connected to the postintersection short arc of the first section (see Figure 4).

The arcs of the contour of the first section are then followed until an intersection point with a contour of

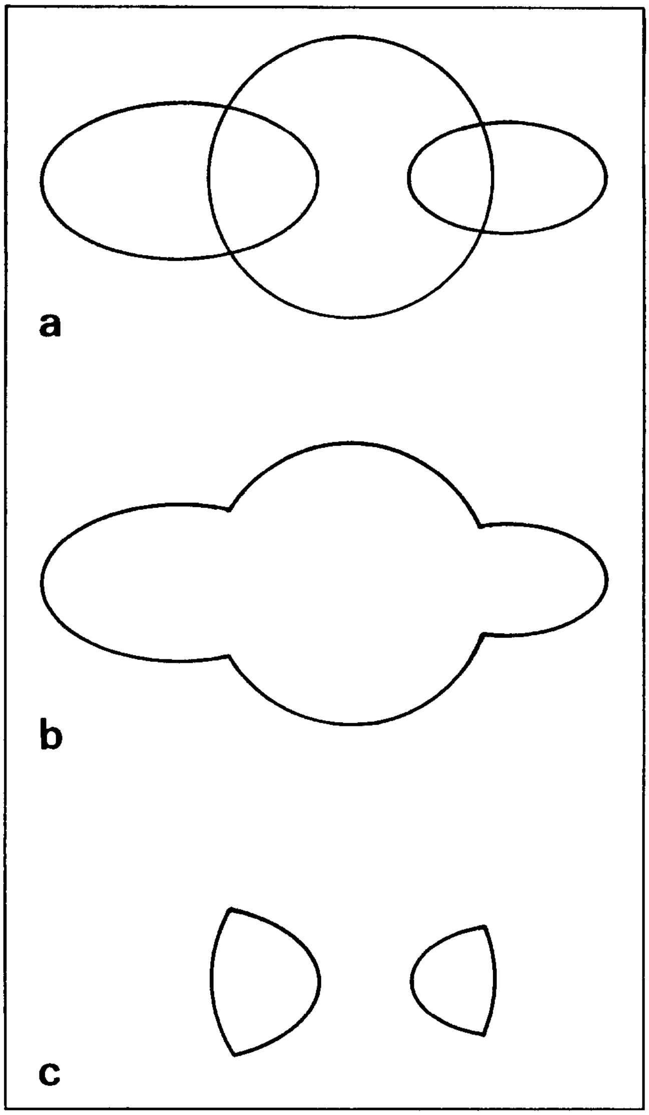

Figure 3. (a) Intersection of two sections, one of which is composed of one contour (circle) and the other of which has two contours (ellipses). There are four points of intersection. (b) Boundary contour of the union of the two sections is composed from the outer parts of the boundary of the original sections. (c) Boundary contour of the intersection of two sections is composed from the inner parts of the boundary contours of the original sections. This variation of the hidden-line elimination algorithm is used to draw the inner surface of clipped proteins (not shown)

the second section is reached. At which point a switch occurs back to a contour of the second section, and so on, until the original arc is reached. The new contour is now closed. The methods for calculating the intersections of a pair of circular arcs, a pair of line segments, or a line segment and an arc are not discussed here because they are special cases of a more general intersection algorithm that will be published elsewhere[11].

All the arcs of the new contour that came from the second section are marked as used, and a search is undertaken to find an unused arc of the second section that lies outside the first section. If such an arc is found, another new contour is created. These new contours are created until there remain no unused arcs of the second section lying outside (ie unobscured by) the first section. All the arcs of these new contours that come from the second section represent the visible part of the second section.

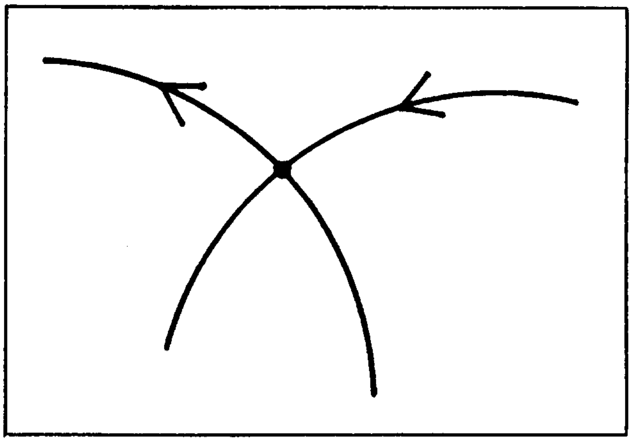

Figure 4. Intersection of two arcs from different contours. Each arc must be subdivided into two short arcs, so that the outer boundary contour (arrows) can be constructed

These newly created contours are grouped into regions and the regions accumulated as the algorithm progresses from plus to minus z are called the current mask, because they cover the contours below. The same procedure, described above, that was applied to the first and second sections, is applied to the current mask and the third section. This results in part of the third section being drawn, and an updated mask being formed. All the sections of the protein are successively processed in this manner.

A small variation in the method can be used to visualize the inner protein surface. A hither clipping plane is specified, and all sections above the clipping plane are discarded. The sections below the clipping plane are processed as described above, except that the first mask consists of the area of the plane lying outside of the regions of the first section, and for subsequent sections the lines that are drawn lie in the interior hole of the mask (see Figure 3(c)). This variation would be most useful on a plotter with two pen colours.

POLYHEDRAL SURFACE

A simple program (HEE) for performing hidden edge elimination on a triangulated protein surface has been written. The triangulated surface itself is calculated by a program (TS) described elsewhere[10]. Edges are not drawn if they lie underneath some triangle. The midpoint of the edge is used for this check. Edges on the border between two downward-facing triangles are not drawn. Edges on the border between an upward-facing triangle and a downward-facing triangle form an edge of the protein as viewed from above and partially obscure or cut edges directly below them. Each cut edge is subdivided into two edges, each of which is checked for whether it is under a triangle. This is how partially hidden edges are handled.

Plots were produced on a Talaris 1200 laser printer. The three programs PC1, PC2 and HEE are written in FORTRAN 77 and run on a VAX-11/750. They take about 2–5 min each to execute on a small protein. (AMS, CT and TS take a total of about 15 min for a small protein). The protein coordinates were taken from the

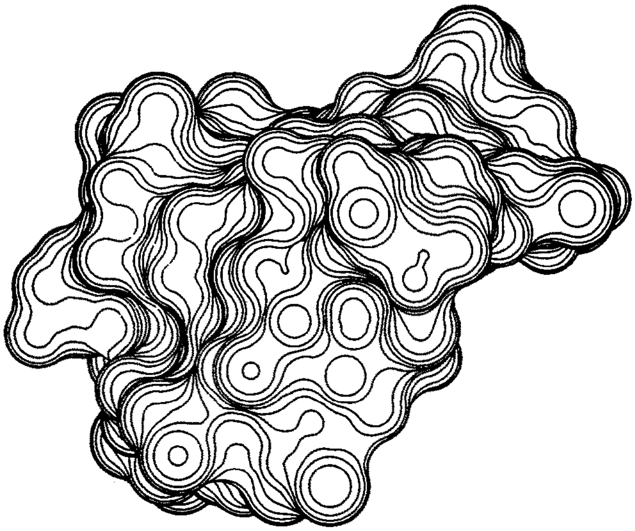

Figure 5. Crambin surface drawn with planar contours 0.35 Å apart

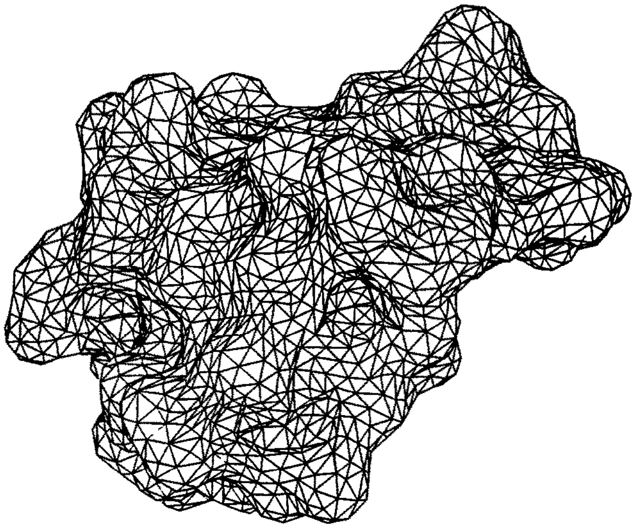

Figure 6. Triangulated surface of crambin

protein data bank at Brookhaven National Laboratory[13]. (Copies of the program can be obtained from the author.)

Figure 5 shows the planar contour surface of crambin[14]. Because the surface z coordinate is multiple-valued as a function of x and y, simply contouring a single-valued function defined by the highest value of z for each x,y would give a less realistic picture. Figure 6 shows the triangulated surface of crambin.

ACKNOWLEDGEMENT

This work was supported, in part, by NIH grant GM34338. The author thanks E Getzoff and J Tainer for useful comments on the manuscript.

REFERENCES

1 **Connolly, M L** *Science* Vol 221 (1983) pp 709–713

2 **Lee, B and Richards, F M** *J. Mol. Biol.* Vol 55 (1971) pp 379–400

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

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

5 **Connolly, M L** *J. Appl. Crystallogr.* Vol 16 (1983) pp 548–558

6 **Quarendon, P** *J. Mol. Graph.* Vol 2 (1984) pp 91–95

7 **Connolly, M L** *J. Mol. Graph.* Vol 3 (1985) pp 19–24

8 **Connolly, M L** (1981) *PhD Dissertation,* University of California, Berkeley, USA

9 **Langridge, R, Ferrin, T F, Kuntz, I D and Connolly, M L** *Science* Vol 211 (1981) pp 661–666

10 **Connolly, M L** *J. Appl. Crystallogr.* Vol 18 (1985) pp 499–505

11 **Connolly, M L** *Visual Computer* (1985)

12 **Flanders, H** *Differential forms* Academic Press, New York, USA (1963) pp 77–78

13 **Bernstein, F C et al.** *J. Mol. Biol.* Vol 112 (1977) pp 535–542

14 **Hendrickson, W A and Teeter, M M** *Nature* Vol 290 (1981) pp 107–113