

Triangulation algorithms for the representation of molecular surface properties

Wolfgang Heiden, Michael Schlenkrich and Jürgen Brickmann

Institut für Physikalische Chemie, Technische Hochschule Darmstadt, Petersenstrasse 20, D-6100 Darmstadt, F.R.G.

Received 13 September 1989

Accepted 27 October 1989

Key words: Molecular surface; Molecular modelling; Interactive computer graphics; Triangle mesh

SUMMARY

A triangulation algorithm for a dotted surface (i.e. a surface defined by point coordinates in three dimensions) is given. The individual triangles are generated on the basis of a hierarchy of strategies according to increasing surface complexity. While for small molecules an elementary algorithm is sufficient to triangulate the surface, large molecules – like proteins – generally need all steps of the hierarchy. Although this program has been developed with the aim of triangulating molecular surfaces, it can in principle be applied to any surface defined by 3D point coordinates.

1. INTRODUCTION

Molecular graphics technologies are becoming increasingly important in many fields of molecular science [1]. This is on the one hand a direct consequence of the dramatic increase of super computer technology: many different molecular quantities can be calculated within a reasonable time. On the other hand, it is closely related to the fact that complex information can be visualized interactively using the new generation of graphics workstations. Conventionally, real-time interactive graphics applications have relied on calligraphic primitives – lines and points – to display 3D data and objects [2]. The capacity of modern workstations to manipulate shaded and transparent surfaces interactively enables the scientist to use solid spatial representations of molecular objects in addition to the standard molecular representations originating from calligraphic visualization systems. This can help to obtain qualitative and quantitative information on the molecular scenario in a more effective way. A typical example is the representation of the electrostatic potential on the van der Waals surface of a molecule [3]. This potential can be encoded by colouring the surface. In this case the dot surface representation is limited by the viewer's ability to concurrently assimilate an understanding of the molecule's skeleton, the 3D and the 1D surface quality (electrostatic potential for example) [4,5].

The visualization of a shaded surface with raster display hardware needs in any case an analytical representation of this surface. This representation can be the result of a piecewise analytical construction [6] or an interpolation in a pointwise given manifold. In the first case, there are generally no problems in visualizing the surface – for example according to a ray tracing algorithm. The explicit calculation, however, can become very cumbersome since graphical primitives, such as shaded polygons, cannot be directly applied. The second case is more problematic. If no other information except the positions of the surface points is available, topological complexity may lead to severe problems in the interpolation strategy.

In this paper we present a hierarchy of strategies for the surface interpolation based on surface triangulation. Among all polygons, triangles have been found to show several advantages in the representation of a surface:

(a) Flat polygons facilitate the analytical calculation of surface integrals. The triangle interpolation is the only general polygon representation for a set of points on a curved surface. The Gouraud-shading algorithm [7] can be used to generate a smoothly curved image of shaded surfaces given by a triangle mesh.

(b) Moreover, the representation of a surface by a triangle mesh is extremely efficient in speeding up the interactivity of the graphic display of large surfaces. In this approach all information for the generation of the picture is kept in storage once computed. Consequently, there are no further calculations necessary during the drawing process. A further acceleration can be achieved if the surface is modelled using series of triangles, each possessing a side in common with its immediate predecessor. Within such a ‘triangle mesh sequence’ (*tmesh* sequence) every new triangle is defined by just one new point. This reduces the number of transformations necessary for a shaded image to one third, which again increases the speed of graphic display to a dramatic extent. Furthermore, the triangle mesh is also generally accepted in most graphics libraries.

In recognition of these advantages several algorithms have been described for the generation of triangulated surfaces. A program designed by Connolly [8] generates triangulated ‘solvent-accessible molecular surfaces’ (following the definition by Richards [9]) directly from a set of atomic coordinates, while another method, applied by Kahn [10], generates points connected by triangles on van der Waals spheres around atomic centre coordinates. Both strategies are applicable only for molecular surfaces on the basis of general assumptions on molecular structure. A more general algorithm for triangulation of a given dotted surface is presented by Zauhar and Morgan [11], but because of problems arising with a certain grade of surface complexity this algorithm is limited to small molecules (up to oligopeptides).

In order to deal with these problems we have relied on a hierarchy of strategies, applying different triangulation conditions to different surface regions. Since it is not possible to identify all critical regions in advance with only the point coordinates given, we designed an automatism which detects areas in which triangulation errors have occurred. This then clears the areas of already generated triangles and omits them from subsequent triangulation, leaving them for special treatment.

In Section 2 the triangulation method is described, first presenting the elementary algorithm and then turning to additional strategies for triangulation of critical areas. Section 3 contains a few examples like Coulomb surfaces of proteins, while in the final section some conclusions are drawn.

2. TRIANGULATION ALGORITHMS

The *tmesh* strategy always starts from an already existing connection of two surface points (representing one side of a triangle), and generates a new triangle by connecting these two points with a third one (see Fig. 1).

Every triangle side is represented by a line connecting two surface points and a status specified for this line. This status can change during triangulation and has an influence on the further treatment by the progressing program of the points concerned. The status conditions are:

(i) *OUTER line*, represents the first connection of two surface points. It is involved in only one triangle, but may become part of another one. This happens when it is used as a basis for a new triangle, or when it completes another triangle with a different basis. During this process the status changes to

(ii) *INNER line*, which is part of two triangles, thus completely defining a surface region and no longer available for another triangle.

(iii) *BORDER line*: frames surfaces which are not closed. The line takes part in only one triangle, but is treated by the algorithm like an INNER line.

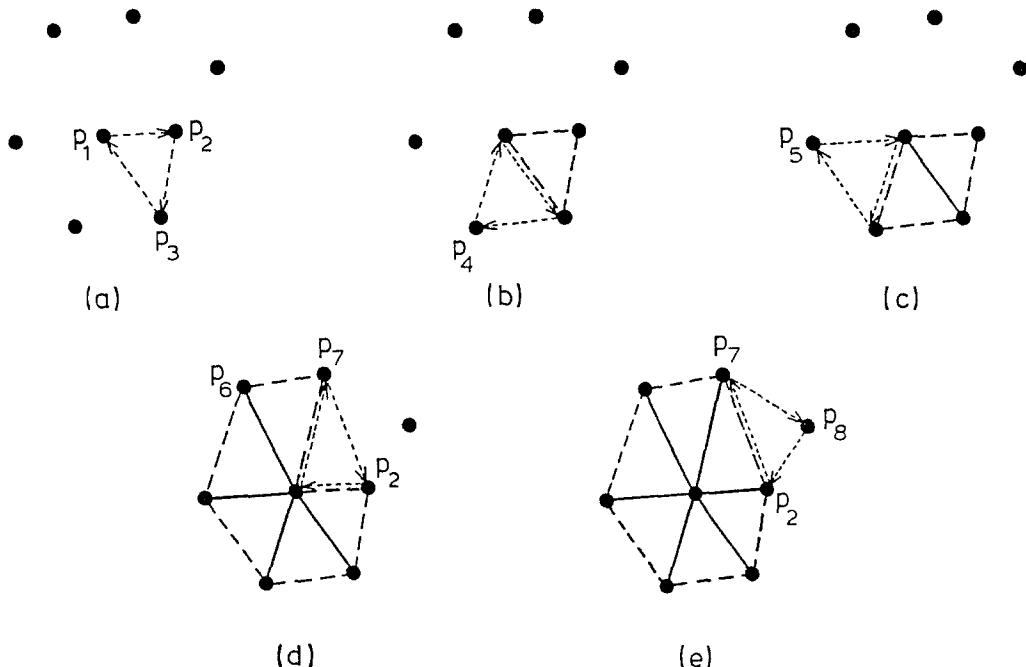


Fig. 1. Growth of a triangle mesh. Dotted arrows show the direction of the lines building the next triangle. This defines the orientation of the normal vector. (----) lines define OUTER connections, (—) lines define INNER connections. (a) Starting from P_1 the initial triangle is generated with the nearest points P_2 and P_3 . (b) The first triangle (P_1, P_2, P_3) is framed by OUTER lines, the last of which (P_3-P_1) is used as a basis for the next triangle with point P_4 . When P_1 and P_3 are connected for the second time, the direction of this line is inverted (P_1-P_3). (c) The common line of the first and the second triangle has changed to an INNER one. A new triangle is built with P_5 . (d) A circle of triangles around point P_1 closes upon itself with the triangle (P_1, P_7, P_2), finishing this part of the surface. (e) Triangulation continues from the last drawn OUTER line with point P_8 .

(iv) *RESTRAINED line*: is part of a ‘self-healing system’ which enables the program to detect and correct triangulation errors. This status is intermediate between OUTER and INNER since it allows the line to be used to complete a triangle with a different basis (then changing to INNER status), but not to form such a basis. For an example of a partially triangulated surface with lines of different status, see Fig. 7.

During the triangulation process a ‘dot normal vector’ is calculated for every surface point as the average of the normal vectors of all adjacent triangles. Triangle normal vectors $\vec{n}(1,2,3)$ are given by the vector product relation

$$\vec{n}(1,2,3) = \vec{V}_{12} \times \vec{V}_{23} / |\vec{V}_{12} \times \vec{V}_{23}|$$

for a triangle (P_1, P_2, P_3) , where \vec{V}_{ij} is a vector pointing from P_i to P_j . Therefore the numbering of these points is essential for the orientation of the normal vector (see direction of new triangle generation in Fig. 1). The correlation of normal vectors of neighbouring triangles is essential to our method. Loosely speaking, the normals must point ‘outward’ with respect to the object represented by the surface points.

For a better understanding we first describe the elementary triangulation algorithm before we turn to the special strategies of dealing with complex surfaces.

2.1. Elementary algorithm

The basic triangulation strategy of the surface as a sequence of triangles can be subdivided into several steps (see Diagram I):

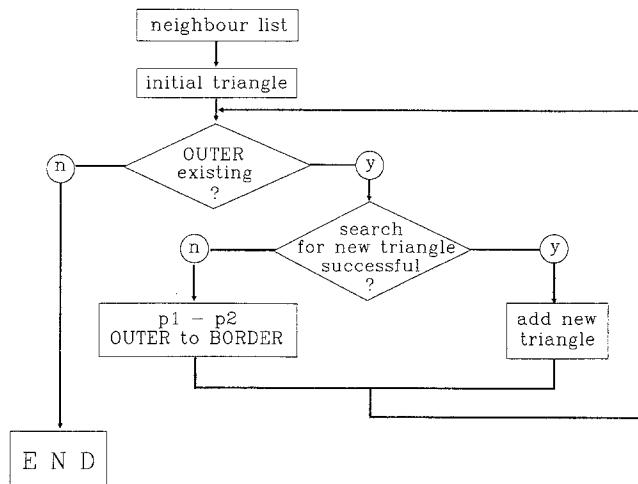


Diagram I. Elementary triangulation.

(a) Generation of a neighbour list for all points of the surface. This list is generated according to a distance criterion. The success of the triangulation of complex (folded) surfaces primarily hinges on a suitable choice of the critical distance. The actual value is related to the given point density. An optimum number of 20–50 neighbours was found in the explicit calculations.

(b) Definition of an initial triangle (P_1, P_2, P_3). The initial point P_1 is chosen as that furthest along one of the coordinate axes. This enables the program to define a unique orientation of the normal vectors (by inverting the numeration of the points if necessary) as pointing outward with respect to the centre of the given point set. The two points closest to the initial point P_1 are chosen as P_2 and P_3 .

(c) Growth of a triangle mesh. For smooth surfaces triangulation is straightforward. The individual steps can be described as follows (see Diagram II):

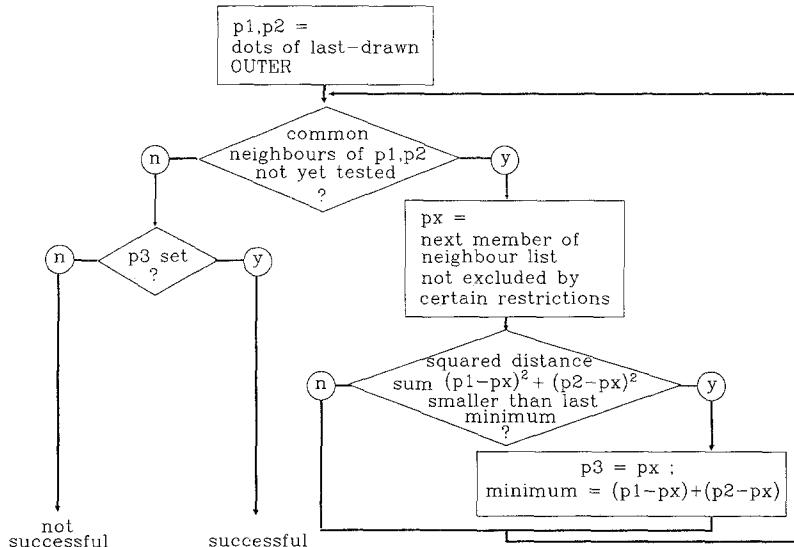


Diagram II. Search for new triangle.

(c1) Start from the last drawn OUTER line (connecting two points P_1 and P_2) as basis for a new triangle.

(c2) Find point P_x for a new triangle (P_2, P_1, P_x) with minimum perimeter. Minimizing the sum of the squares of the distances instead of the perimeter gives preference to the formation of more

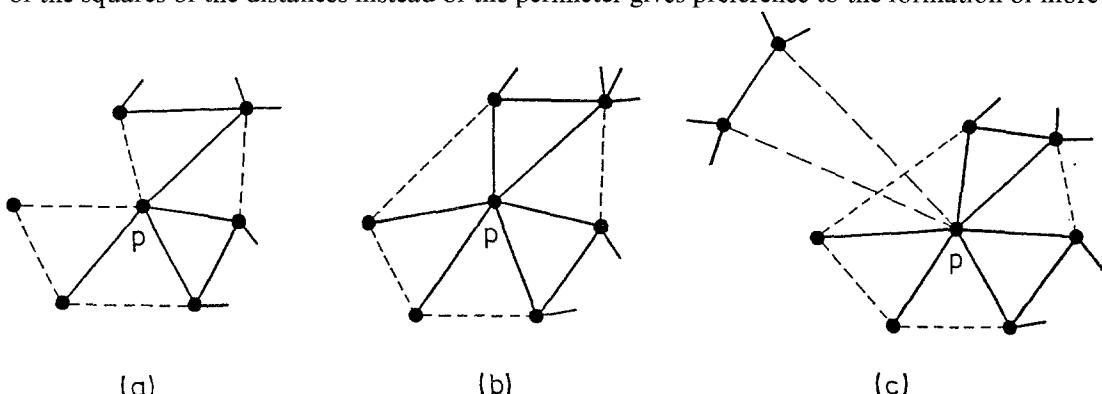


Fig. 2. Definition of *inner dots*. (a) P is not an *inner dot*, because it is still in contact with OUTER lines (---). There is still at least one correct triangle which can be built using P . (b) P is totally embedded in a surface, taking part in only INNER lines (—), which makes it an *inner point*. (c) An extraneous triangle addressing the *inner point* P , is topologically wrong.

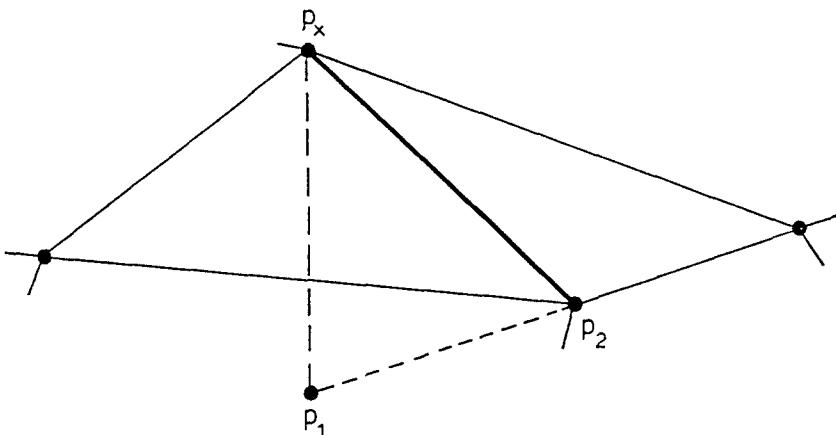


Fig. 3. No line may be incorporated in more than two triangles. A triangle with basis (P_1-P_2) including the INNER line (P_2-P_x) does not fit to the surface defined by the earlier triangles which have the line (P_2,P_x) in common.

or less equilateral triangles. P_x is chosen from the common neighbours of P_1 and P_2 (i.e. within a certain distance range from P_1 and P_2) which are not excluded from triangulation by certain restrictions. These restrictions are introduced in order to avoid the formation of triangles which would no longer represent a surface.

Points which are already totally embedded in a triangulated surface must not become part of any further triangle. Such ‘inner points’ are identified by their incorporation in INNER lines only (see Fig. 2).

None of the sides of the new triangle (P_2,P_1,P_x) may already exist as an INNER line, because such a triangle would represent a planar intersection with an existing surface (see Fig. 3).

The formation of a new triangle with a surface normal pointing ‘inward’ is prevented by comparing the surface normal with the point normals (see above) of the adjacent points. The new triangle is accepted only if the deviation is found not to exceed a set limit (*normal angle restriction*, see Fig. 4).

(c3) Add the new triangle (P_2,P_1,P_x) to the list if the search was successful (i.e. a new point P_x could be found), else change the basis line (P_1,P_2) to BORDER status. Then start a new cycle with (c1).

The elementary triangulation is finished when there are no more OUTER lines available.

2.2. Error handling

The elementary algorithm is sufficient for the triangulation of surfaces that are more or less homogeneous in dot density and curvature like the ones accessible for small molecules determined from the Connolly program [12]. With increasing surface complexity (for instance the van der Waals type surface of a protein), problems arise which make a triangulation of the complete surface impossible without an adjustment of parameters. Additional information besides the mere surface point coordinates is necessary. As an example, Fig. 5 shows a situation where it is impossible to avoid the formation of a topologically wrong triangle while using the same normal angle restriction limit for two different regions of the same surface.

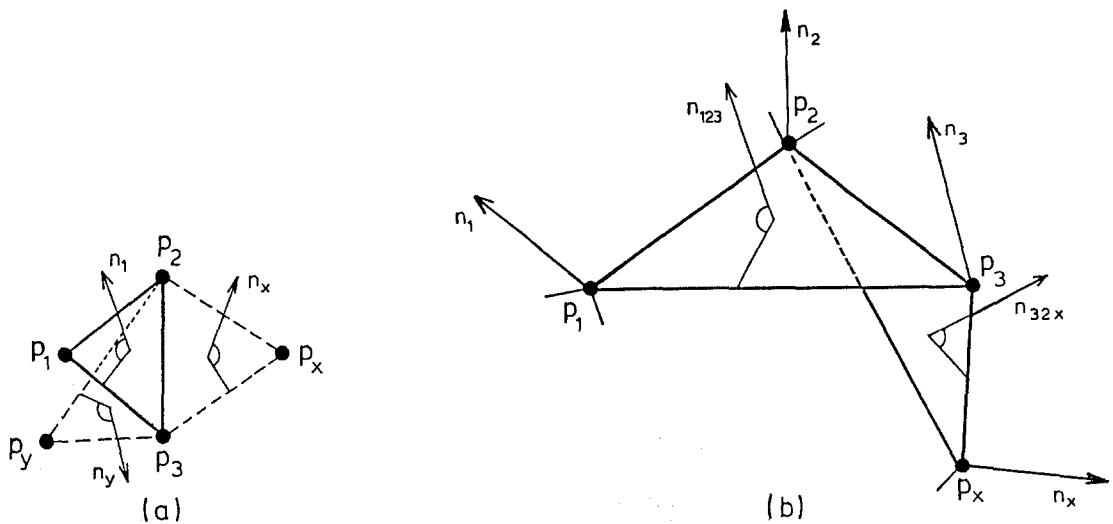


Fig. 4. Normal angle restriction. (a) There are two possibilities of forming a new triangle with the basis line (P_3-P_2). A correct triangle may be built with P_x , resulting in a normal vector n_x similar to n_1 on (P_1, P_2, P_3) . With n_y on (P_3, P_2, P_y) , the orientation of the surface would be inverted. (b) As the normal vector on a single triangle (like $n_{1,2,3}$) is not necessarily the best representation for the 'real' orientation of a surface, the test for normal angle with the new triangle's normal vector $n_{3,2,x}$ is done with the normal vectors on the apices (P_3, P_2, P_x), representing the average orientation of all adjacent triangles. (---) OUTER lines, (—) INNER lines, (--) 'hidden' lines (regardless of their status).

Furthermore, even triangles which are topologically correct can lead to a dead end by preventing the later formation of the only possible correct triangle in a region of low dot density. Without gradient information like that used by Dubois and his group [13], such errors cannot be precluded, though their consequences may in general be detected. So we provided the possibility to identify and remove triangulation errors, and retain critical regions for a treatment with modified conditions (see Diagram III).

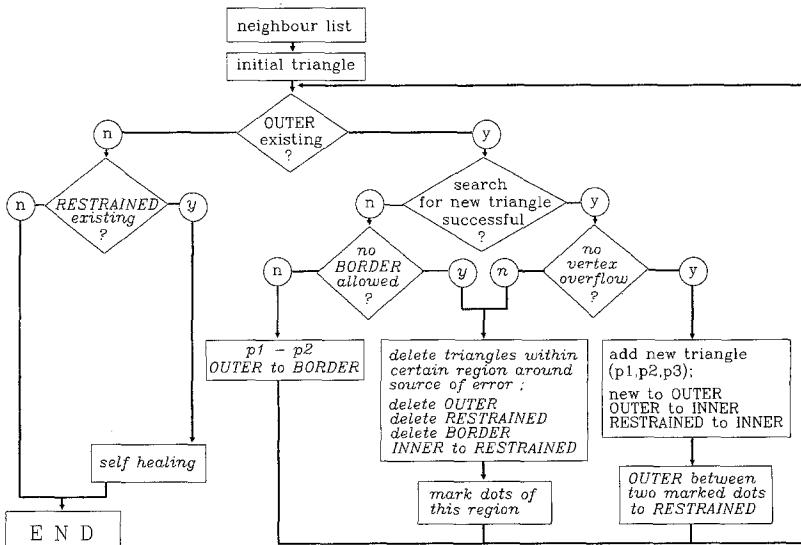


Diagram III. Triangulation including self healing.

Triangulation errors are detected by the occurrence of BORDER lines inside closed surfaces, or when the number of vertices at a surface point exceeds a given limit. The latter criterion in particular has proven most useful, since the first obvious consequence of a triangulation fault is often a cancer-like growth of triangles around a single surface dot.

Upon identification of triangulation errors all triangles in the neighbourhood of a critical one are removed from the triangle list one by one, and the status of each participating line is reset. During this procedure lines incorporated in only one triangle (i.e. OUTER, BORDER and RESTRAINED) are deleted, while INNER lines change to RESTRAINED. All surface points in this region are marked in order to separate them from the surface accessible to elementary triangulation by RESTRAINED lines. This is achieved by changing an OUTER line which connects two such marked points to RESTRAINED status. The special qualities of RESTRAINED lines [mentioned in (iv)] provide that small mistakes can be corrected at once. This may be achieved by generating another triangle with the line in question which is different from the one which led to the error. If the error-causing situation cannot be remedied without a change of parameters, the critical regions become framed by RESTRAINED lines, which retains them for a triangulation cycle yet to follow.

If RESTRAINED lines still exist after completion of the elementary triangulation, these critical areas are treated within a hierarchy of strategies (see Diagram IV).

(d) In a first step the critical areas are treated again with the elementary algorithm. Since the standard procedure is completed, this strategy is in many cases sufficient as new triangle sequences are built preventing the original error situation from recurring. In all following steps every critical

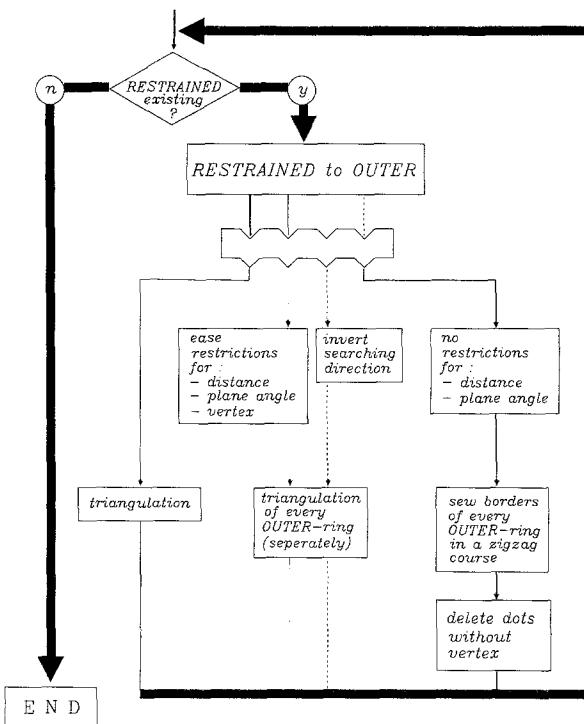


Diagram IV. Self healing.

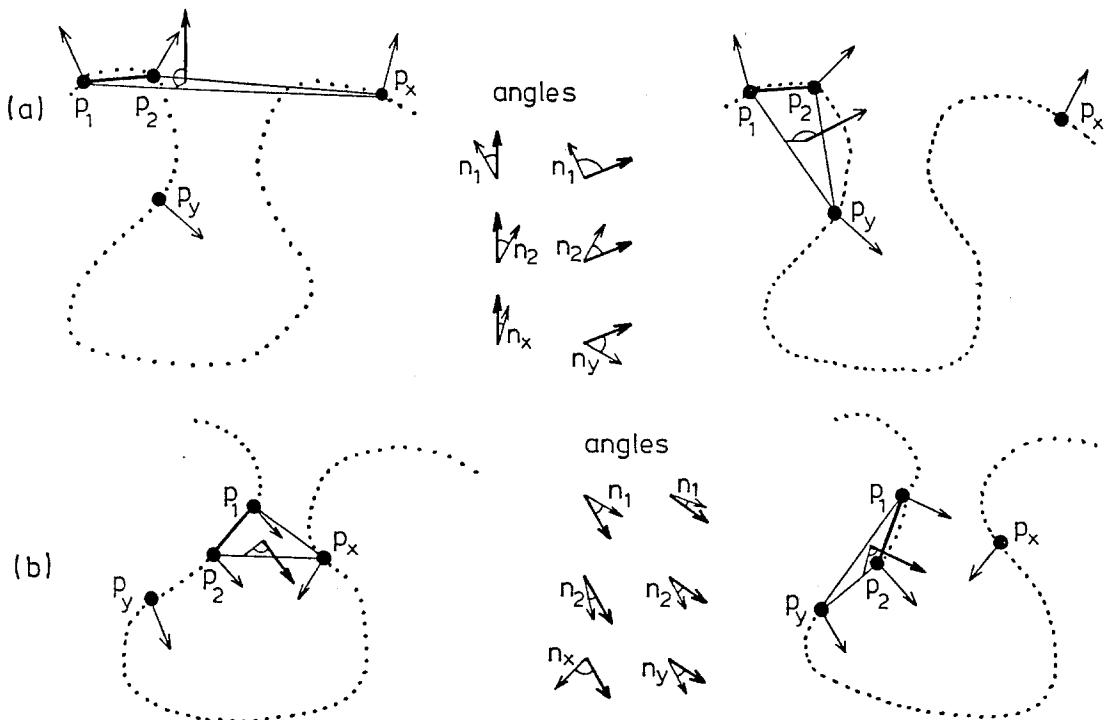


Fig. 5. Example for a situation where at least one topologically incorrect triangle is built inevitably, if the same normal angle restriction limit is applied to different regions of the same surface. The analytical surface is represented by dotted lines. P_1 and P_2 define the basis line for the new triangle, while P_x and P_y are another two surface points within their neighbour lists. Thin arrows show the normal vectors on the surface points, thick arrows are the normal vectors of the new triangle. In situation (a) the correct triangle is (P_1, P_2, P_x) . On the basis of the minimum perimeter its formation would also be preferred, but on the basis of the normal angle criterion its construction would nevertheless be prohibited in favour of (P_1, P_2, P_y) , if the allowed deviation is less than 90° . On the other hand, in situation (b) only a limit lower than 90° can prevent the formation of the wrong – but distance-preferred – triangle (P_1, P_2, P_x) .

area is treated separately in order to avoid connections between topologically different surface regions, now made possible by a lower distance restriction.

(e) If areas framed by RESTRAINED lines are still present the restrictions for distances (concerning the neighbour lists) and normal angles are moderated, and the elementary algorithm is applied.

(f) If this still does not lead to a complete triangulation the treatment of the point sequence is inverted and the algorithm is started again. This again may lead to triangle sequences avoiding a situation which led to a mistake in earlier cycles.

(g) If (d)–(f) have not totally succeeded in the triangulation of the critical areas, remaining gaps (framed by RESTRAINED lines) are closed omitting all isolated points by a zig-zag procedure (see Fig. 6).

2.3. Performance

All triangulations of dotted surfaces containing less than 60 000 points were performed on an Iris 4D 70/GT workstation. Typically the triangulation of a 20 000–30 000 dot surface like trypsin

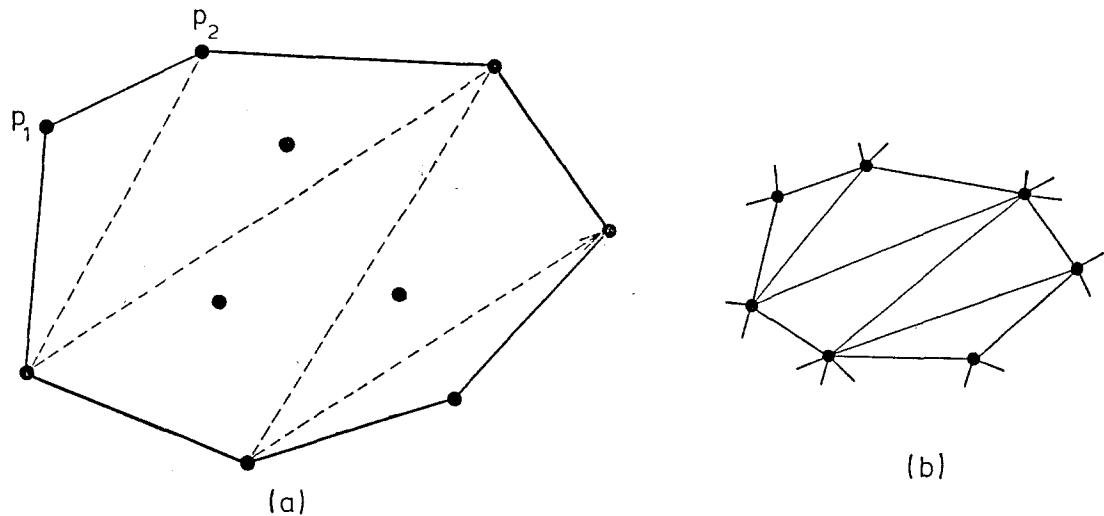


Fig. 6. Ultimate closing of surface gaps. (a) Starting from line (P_1-P_2) the gap (defined by RESTRAINED lines) is closed in a zig-zag course, omitting points without vertex. (b) This procedure results in a closed triangulated surface after deletion of the vertex-free points.

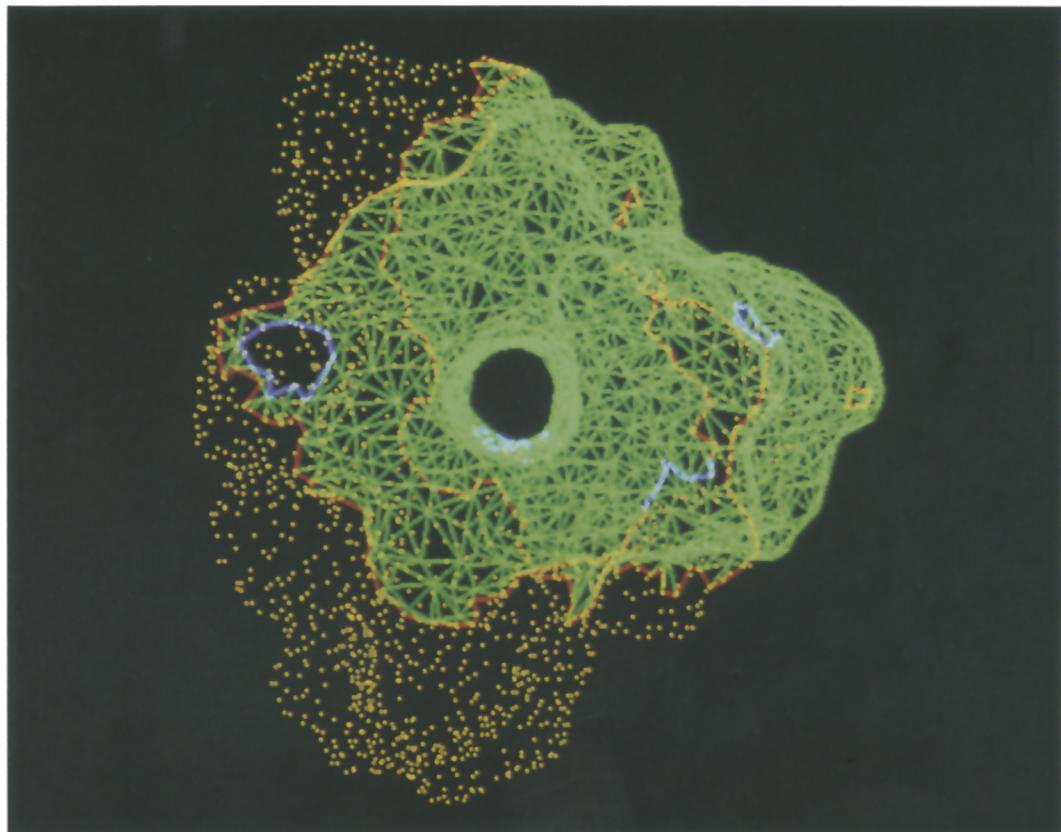


Fig. 7. Dotted surface of a gramicidin-A monomer during triangulation. OUTER connections are shown in red, INNER connections are shown in green, RESTRAINED connections are shown in pink.

(see Figs. 8a,c and d: 31 455 points) takes about 2 h, where the generation of a neighbour list needs only 10 min. The runtime for triangulation and neighbour list generation scales linearly with the number of surface points. The memory requirement for trypsin was about 12 Mbyte.

3. APPLICATIONS

The knowledge of a complete triangle mesh can be used to calculate almost all surface qualities. Surface integrals can be calculated on the basis of the given normal vectors. Also volume integrals can be solved by transferring them into surface integrals using Ostrogradski's formula [14]. The triangle representation opens effective strategies for volume and surface measures. Zauhar et al. use a triangulated surface for calculation of an electric potential [11]. The main advantage of a *tmesh* surface, however, is related to the fact that such a representation is supported by most of

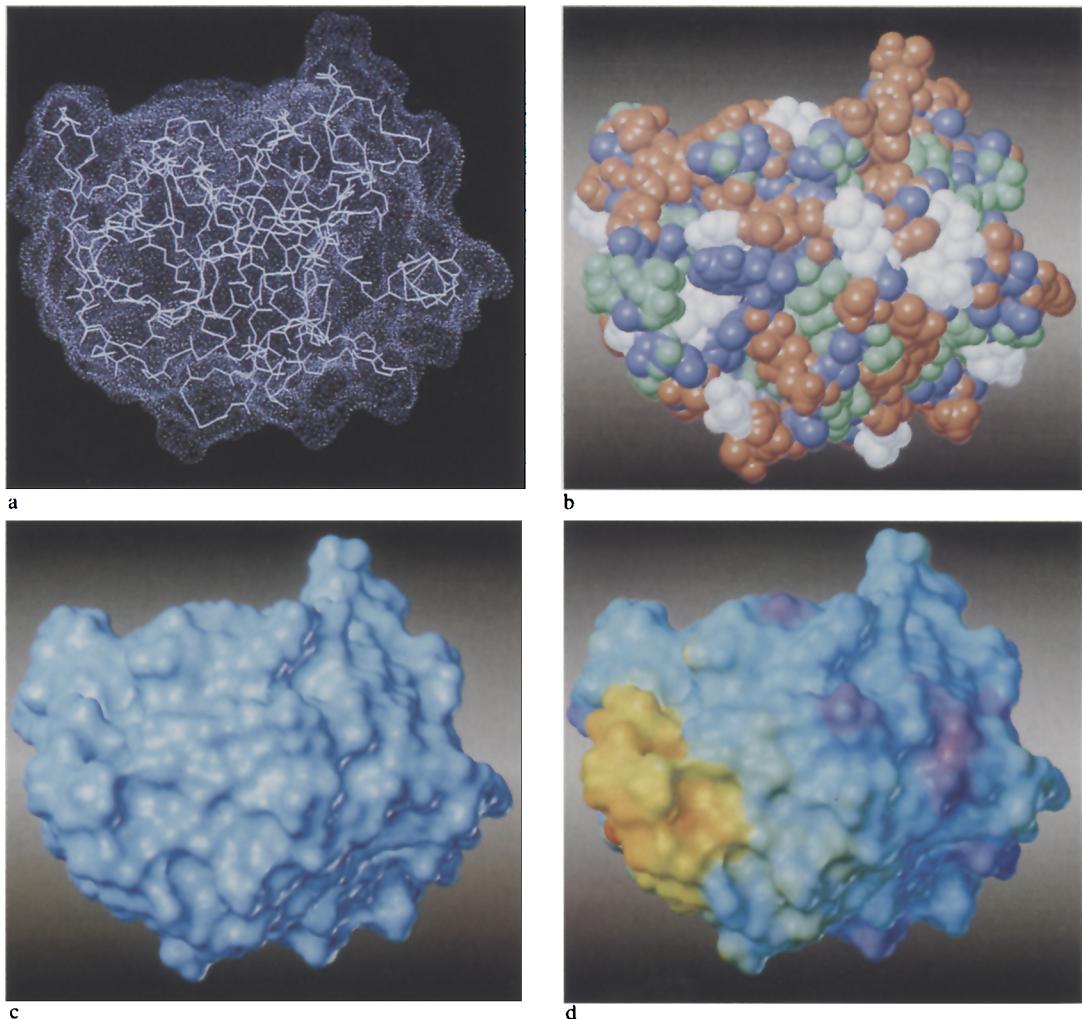


Fig. 8. Different display models of trypsin. (a) Dotted surface with peptide backbone as chicken wire. (b) CPK model. (c) Solid surface. (d) Solid, with electrostatic potential – varying from red (negative) to blue (positive).

the modern raster graphics hardware – a super workstation is able to generate roughly 10^5 Gouraud-shaded triangles per second if these triangles are given as *tmesh* sequences. Consequently, shaded images of even complex objects with a *tmesh* surface can be interactively manipulated on the screen. The solid surface representation of a molecule is definitely a powerful tool in certain studies within the field of computer-aided molecular modelling (CAMM). This is demonstrated in Figs. 8a–d with trypsin (a proteolytic enzyme) as an example. Possible receptor sites can be identified visually by searching for surface areas with topological complexity *and* a characteristic electrostatic potential. In the dotted surface representation the complexity of the screen image may camouflage the relevant information by interference of background and foreground dots, even if it is supported by a chicken wire representation of the protein main chain (Fig. 8a). Also a CPK (Corey, Pauling, Koltun) model (Fig. 8b) is of no great help for this retrieval. The solid-surface representation (Figs. 8c and d), however, reduces the amount of information to the relevant part. Using a colour code for the electrostatic surface potential, receptor sites can be discovered easily, as is shown in Fig. 9. The electrostatic potential was calculated on the basis of partial charges at the atom sites. The substrate recognition site ('specificity pocket') of trypsin [15] attracts attention by its topological and electrostatic complexity while the calcium-binding site is a deep and narrow pocket with a high concentration of negative charge. The specificity pocket may be identified even better if the magnitude of the electrostatic field is displayed instead of the potential (Fig. 10). Solid-surface representations with reactivity colour coding can also be helpful for an understanding of differences concerning the biological activity of structurally similar proteins. This can be seen by comparing trypsinogen (an inactive proenzyme which changes to active trypsin when the first 15 residues of the peptide chain are removed) (Fig. 11) and trypsin (Fig. 9). While most of the protein surface is quite similar in both molecules, in trypsinogen the specificity pocket is masked by a narrow entrance, and the negative potential at its bottom is shielded.

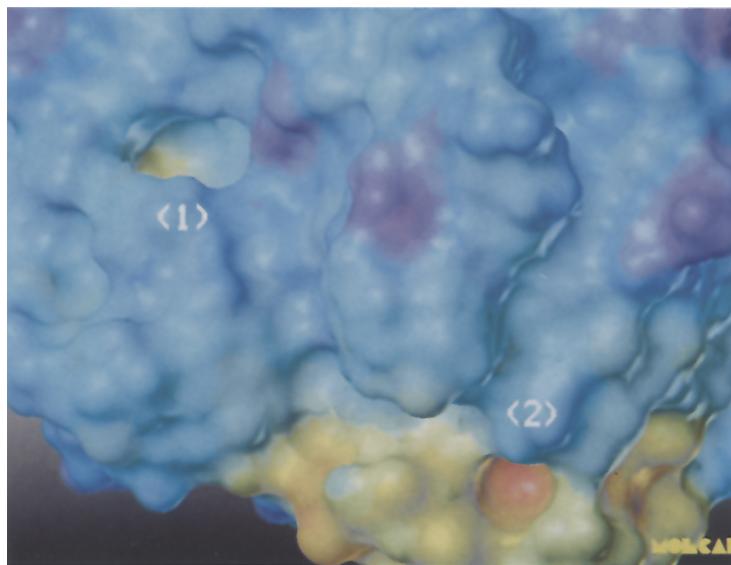


Fig. 9. Trypsin, solid model coloured according to electrostatic potential, showing two receptor regions: (1) specificity pocket, and (2) calcium-binding site.

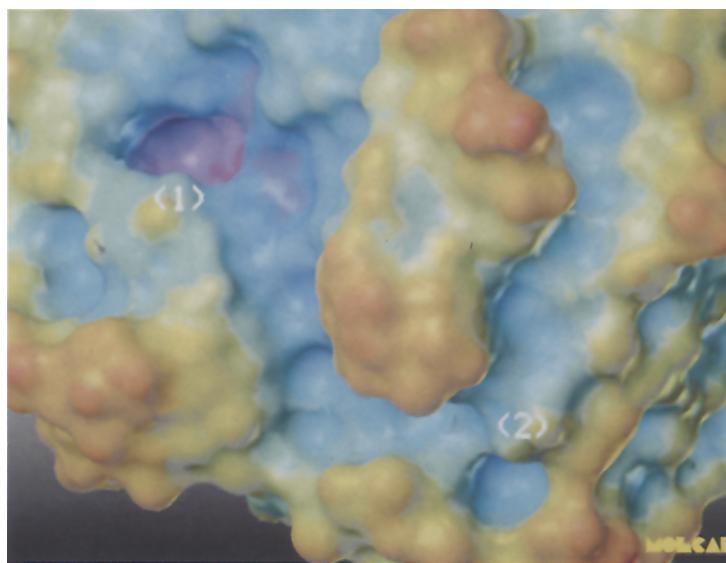


Fig. 10. Trypsin, solid model coloured according to electrostatic gradient, showing two receptor regions: (1) specificity pocket, and (2) calcium-binding site.

The solid-surface representation should not be seen as a replacement of the established techniques (Figs. 8a and b) used in CAMM (originating from hardware restrictions of the old vector graphics terminals) but as an extension. This is demonstrated in Fig. 12. The specificity pocket of trypsin (inhibited by benzamidine) is represented as a transparent surface while the adjacent parts of the protein are shown in standard chicken wire representation. With this combination of repre-

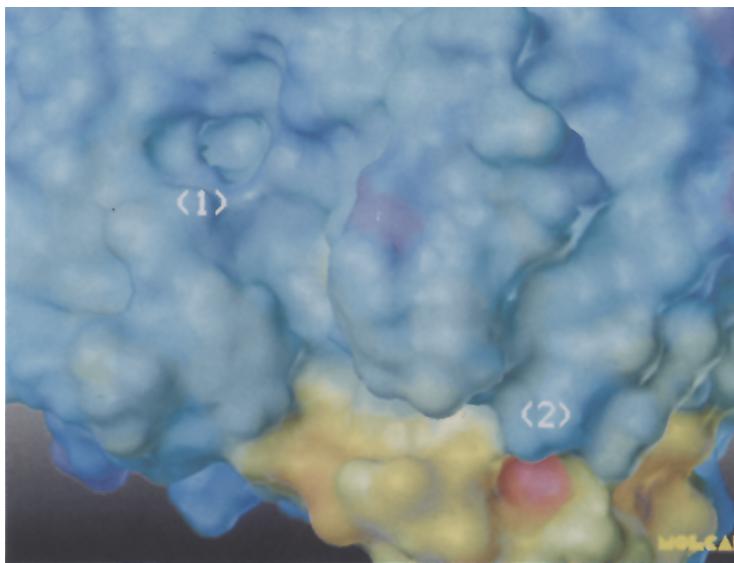


Fig. 11. Trypsinogen, solid model coloured according to electrostatic potential (the N-terminal residues, which are removed for activation of trypsinogen, are not shown). (1) Specificity pocket, and (2) calcium-binding site.

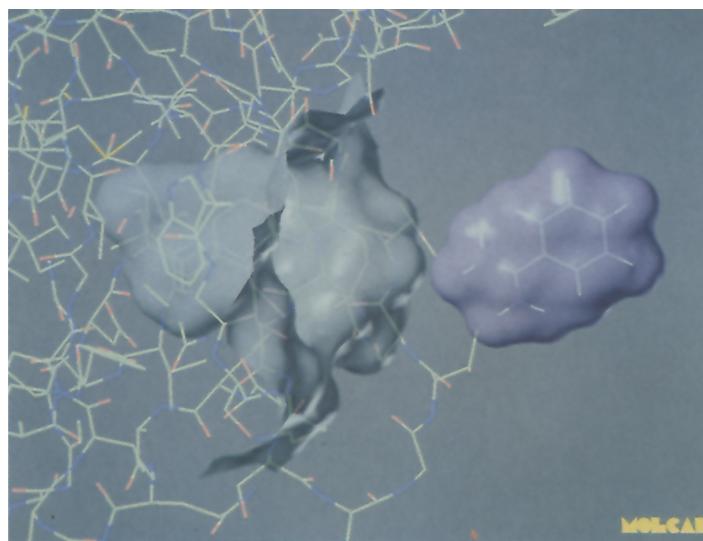


Fig. 12. Docking of the artificial inhibitor benzamidine into the specificity pocket of trypsin.

sentation techniques the advantages of both can be used: the chicken wire allows the identification of molecular details while the transparent solid surface makes interactive docking feasible.

4. CONCLUSION AND OUTLOOK

A hierarchy of strategies for an effective triangulation of surfaces on the basis of a set of surface points has been presented. Here the individual triangles are arranged in a triangle mesh, guaranteeing high interaction speed of solid surface images on modern raster graphics workstations. It has been demonstrated with a few examples that the extension of CAMM techniques by the introduction of colour-coded molecular surfaces dramatically increases the ability of these tools. Up to now there is no limit to be seen for the capacity of modern workstation technology. A display rate of 10^6 triangles per second in a *t*mesh arrangement seems to be possible within the near future. With this capacity new types of representation in CAMM will appear. The future visualization strategy will probably not try to display as much information as possible but to reduce the image to the relevant information. The use of colour-coded (possibly transparent) surfaces is a step into this direction.

ACKNOWLEDGEMENTS

The images shown in this paper are generated using the display package MOLCAD which has been developed in the group of the authors. We like to thank Martin Knoblauch and Michael Waldherr-Teschner for cooperation. This work is supported by the Bundesminister für Forschung und Technologie, Bonn, the Fonds der Chemischen Industrie, Frankfurt, and Silicon Graphics Inc. Mountain View (U.S.A.), Munich. We particularly thank Dr. W. Bode for provid-

ing the coordinates of bovine β -trypsin and its complex with benzamidine, as well as Dr. P. Bopp and Miss A. Milne for reviewing the manuscript.

REFERENCES

- 1 Max, N., J. Mol. Graph., 2 (1984) 8.
- 2 Langridge, R., Ferrin, T.E., Kuntz, I. and Connolly, M.L., Science, 211 (1981) 661.
- 3 Quarendon, P. et al., J. Mol. Graph., 2 (1984) 4.
- 4 Getzoff, E.D., Tainer, J.A., Weiner, P.K., Kollman, P.A., Richardson, J.S. and Richardson, D.C., Nature, 306 (1983) 287.
- 5 Weiner, P.K., Langridge, R., Blaney, J.M., Schaefer, R. and Kollman, P.A., Proc. Natl. Acad. Sci. U.S.A., 79 (1982) 3754.
- 6 Connolly, M.L., J. Appl. Cryst., 16 (1983) 548.
- 7 Gouraud, H., IEEE Trans. Comput., C-20(6) (1971) 623.
- 8 Connolly, M.L., J. Appl. Cryst., 18 (1985) 499.
- 9 Richards, F.M., Annu. Rev. Biophys. Bioeng., 6 (1977) 151.
- 10 Kahn, S.D., Iris Universe, Spring (1989) 24–30.
- 11 Zauhar, R.J. and Morgan, R.S., J. Comput. Chem., 9(2) (1988) 171.
- 12 Connolly, M.L., Science, 221 (1983) 709.
- 13 Yue, Shi Yi, Modélisation Calligraphique de Formes Moléculaires Electroniques et Géométriques dans une Structure Grille, Thesis, 1987.
- 14 Huron, M.J. and Claverie, P., J. Phys. Chem., 76 (1972) 2123.
- 15 Bode, W. and Schwager, P., J. Mol. Biol., 98 (1975) 693.