Journal of Computer-Aided Molecular Design, **12:** 289–299, 1998.
KLUWER/ESCOM
© 1998 *Kluwer Academic Publishers. Printed in the Netherlands.*

A flexible triangulation method to describe the solvent-accessible surface of biopolymers

André H. Juffer* and Hans J. Vogel
Department of Biological Sciences, The University of Calgary, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada

Received 3 June 1997; Accepted 14 November 1997

Key words: boundary element method, continuum electrostatics, solvent-accessible surface, triangulation

Summary

A relatively simple protein solvent-accessible surface triangulation method for continuum electrostatics applications employing the boundary element method is presented. First, the protein is placed onto a three-dimensional lattice with a specified lattice spacing. To each lattice point, a box is assigned. Boxes located in the solvent region and in the interior of the protein are removed from the set. Improper connections between boxes and the possible existence of cavities in the interior of the protein which would destroy the proper connectivity of the triangulated surface are taken care of. The remaining set of boxes define the outer contour of the protein. Each free face exposed to the solvent of the remaining set of boxes is triangulated after the surface defined by the free faces has been smoothed to follow the shape of the macromolecule more accurately. The final step consists of a mapping of triangle vertices onto a set of surface points which define the solvent-accessible surface. Normal vectors at triangle vertices are obtained also from the free faces which define the orientation of the surface. The algorithm was tested for six molecules including four proteins; a dipeptide, a helical peptide consisting of 25 residues, calbindin, lysozyme, calmodulin and cutinase. For each molecule, total areas have been calculated and compared with the result computed from a dotted solvent-accessible surface. Since the boundary element method requires a low number of vertices and triangles to reduce the number of unknowns for reasons of efficiency, the number of triangles should not be too high. Nevertheless, credible results are obtained for the total area with relative errors not exceeding 12% for a large lattice spacing (0.30 nm) while close to zero for a smaller lattice spacing (down to 0.16 nm). The output of the triangulation computer program (written in C++) is rather simple so that it can be easily converted to any format acceptable for any molecular graphics programs.

Introduction

Electrostatic interactions play a role in the stability and function of proteins [1, 2]. For instance, they play a crucial role in enzyme action and the interaction between lipid membranes and proteins [3]. The finite difference method [4] is commonly employed to calculate the electrostatic potential in and around solvated proteins, e.g., for the calculation of acid dissociation constants of titrating sites in proteins [5]. The method depends on a three-dimensional grid onto which the protein/solvent system is positioned. The electrostatic potentials are determined from the solution of a differential equation (usually the linear Poisson–Boltzmann equation). As an alternative to the finite difference method, the boundary element method (BEM) was developed in recent years [6–9]. This method needs a properly triangulated surface which envelopes the macromolecule of interest. Instead of solving a differential equation, the BEM determines the potential from the solution of two coupled integral equations (that is, if ionic strength is included in the computation, otherwise only one integral equation needs to be solved) which are valid on the surface. Despite the fact that the BEM is a more accurate method in comparison to the finite difference method [6–10] the finite dif-

* To whom correspondence should be addressed.

ference method is generally preferred over the BEM, because the first is rather fast and simpler to implement in comparison to the BEM. In addition, for the latter the number of unknowns depends on the number of vertices of a triangulated surface which greatly reduces the efficiency of BEM algorithms (with ionic strength included, the number of unknowns is twice the number of vertices). In order to overcome this bottleneck of a large number of unknowns, a simple and flexible (that is, the number of triangles should be easily adjustable) triangulation procedure for biopolymers is required. Currently available triangulation methods occasionally result in a large number of triangles (e.g., [11]) or are not capable of generating a sufficiently accurate triangulated surface (e.g., [9]).

Several methods for the triangulation of the solvent-accessible surface varying in accuracy and simplicity have been developed in the past [6b, 9, 11–21]. The solvent-accessible surface is located at the center of a spherical solvent molecule in contact with the van der Waals surface [22]. Lorensen and Cline [21] introduced the technique of a 'marching-cube' in which cubes are used to define the shape of an object. Based on the number of vertices of a cube located in the solvent, a triangulation scheme for the cube is defined. In this way, a triangle topology using a 'divide-and-conquer' approach can be set up. Such an approach was employed by Bharadwaj et al. [10] for the creation of a molecular surface (this is the surface located between a spherical solvent molecule and the van der Waals surface) in their fast multipole boundary element method for molecular electrostatics. These authors perform a scaling of the triangle vertices onto the molecular surface, although the article does not describe the details of how this was accomplished. A variant on the marching-cube algorithm was recently developed by Zhexin et al. [16]. These authors superimpose triangular prisms instead of cubes on the protein/solvent system to define the protein solvent-accessible surface. The advantage of using triangular prisms instead of cubes is that the number of triangles is reduced and the algorithm is somewhat easier to implement with respect to the original marching-cube algorithm. Zauhar [12] recently developed a program called SMART (and its predecessor MOLSUF [13]) which delivers a smooth representation of a triangulated solvent-accessible surface. This program generates curvelinear triangular elements from the contact and reentrant portion of the solvent-accessible surface and enables the user to obtain very accurate estimates of the total area and volume of the protein under consideration. SMART is intended for use with the BEM but tends to deliver a rather large number of triangles for certain proteins (e.g., calmodulin [12]). Some algorithms [11, 15] have been developed based on the original Connolly algorithm [14] for which the shortcomings for boundary element applications have been discussed earlier [13]. There exists an algorithm which uses the concept of wavelets to triangulate the surface of some object [17] such as the human head [17b]. This method has not been applied to proteins, though. Other approaches use spherical harmonics to represent the molecular surface which may be useful for BEM applications [18]. These methods belong to the class of so-called 'surface simplification' methods [19].

An alternative approach is to map a set of triangles with known connectivity and orientation onto the solvent-accessible surface which has been defined by some other means, e.g., by a dotted surface. Such a mapping can be easily accomplished if the initial surface is spherical, since for a sphere a triangulated surface with known orientation (for the BEM, normal vectors are usually pointing outward) can be easily generated, e.g., based on a dodecahedron or an icosahedron. Using BEM, this approach was employed to calculate the shift in acid dissociation constants of histidines in the copper-containing protein azurin on oxidizing the copper [9]. Nevertheless, although simple and efficient, especially for small molecules, this triangulation method fails to detect crevices in proteins and in general will not work properly for non-globular proteins such as the dumbell-shaped calcium regulated protein calmodulin [23].

The purpose of this work is to outline in detail a triangulation method which was used recently to calculate acid-dissociation constants by means of the BEM [24]. The method may be employed also for other purposes such as displaying certain properties across the protein surface (e.g., hydrophobicity or the electrostatic potential by means of a coloring procedure), but it has no major advantages over sophisticated surface simplification methods mentioned above. The method presented here has some relation with the marching-cube algorithm but is rather different in many aspects. It uses the concept of mapping a triangulated surface with known connectivity and orientation onto a set of surface points determined by some other means. The method triangulates the solvent-accessible surface (or any other surface defined by the dotted surface). The method aims at delivering a triangulated surface with a relatively small number of triangles of a reasonable quality such that the surface can be utilized by any BEM program on a routine basis.

Materials and methods

The triangulation procedure essentially consists of five stages. First of all, the macromolecule is placed on a three-dimensional lattice. To each lattice point, a box is assigned. In the second step boxes located in the solvent region and in the interior of the protein are removed. The third step consists of the triangulation of the free faces of the remaining set of boxes. As a fourth step a smoothing operation is performed. Then, in the fifth and final step, the triangulated surface is mapped onto a dotted surface defined by some other means. Each of these steps are discussed in some detail below while applying the method to the protein calmodulin (in this work, entry 1OSA [25] in the Brookhaven Protein Databank [26] was employed). The X-ray structure of this protein has the characteristic dumbell structure [27], so it is very suitable for testing the triangulation algorithm.

First step: lattice

In the first step of the triangulation procedure, the protein is placed on a three-dimensional lattice. The lattice spacing is about the size of a solvent molecule. In general, the lattice spacing should be taken such that in the next step any cleft or invaginations will be detected. The lattice spacing defines the coarseness of the final triangulated surface. In Figure 1, the lattice is shown for calmodulin. To each lattice point, a box is assigned so that for $i \times j \times k$ lattice points, $(i-1) \times (j-1) \times (k-1)$ boxes are generated, where i, j and k are the number of lattice points along the x-, y- and z-axis respectively. The word 'box' instead of 'cube' is employed because the triangulation program (written in C++) adjusts the lattice spacing in each direction to meet the requested number of boxes in each direction based on the initial (user-specified) lattice spacing and the size of the protein. Note that it is essential to have at least two or more layers of boxes around the molecule to ensure that boxes located in cavities are properly identified (see below).

Second step: box removal

In this step, boxes located in the solvent region or in the interior of the protein are removed from the initial set of boxes. The criterion to decide if a box is presumably located in the solvent region is very simple: if the distance between an atom and the centre of a box is larger than the sum of the van der Waals radius of that atom and the radius of a solvent molecule (usually 0.14 nm for water), the box is assigned to be in the solvent region and is removed from the initial set of boxes (instead of using the centre of the box one could also use the vertices of the box). For the neighbours of the removed box, it means that one of their faces is not facing another box. Therefore, its free face is now available for triangulation. After deleting all boxes assigned to be in the solvent, all boxes with no free faces (so they are surrounded by other boxes in all directions) are assigned to be in the interior of the protein and are removed from the initial set of boxes also. The remaining set of boxes needs to be checked for two aspects.

First of all, boxes located in cavities which do not overlap with protein atoms (so were removed) are not discriminated from removed boxes assigned to be in the solvent. Since these boxes are affecting the free faces of their neighbours, these boxes in cavities are the cause of creating an isolated surface in the protein interior defined by the free faces of their neighbours. This will destroy the proper connectivity of the triangulated surface in the sense that the surface will not be simply connected. The boxes located in cavities can be identified using a similar algorithm as proposed by Zhexin et al. [16]. In short, one of the outer removed boxes serves as a starting point for the search and is flagged to be in the solvent. Then, its neighbours are checked. If a neighbouring box was removed also, then this box is located in the solvent also and correspondingly flagged. This is repeatedly done until no removed box is detected to be in the solvent. All boxes not flagged to be in the solvent but were removed are apparently located in a cavity and their neighbours should be corrected as a result. For a proper functioning of this part of the triangulation algorithm, it is necessary to include two or more additional layers of boxes around the molecule.

Secondly, it should be avoided that boxes are connected to each other only through a box edge or box edge point (see Figure 2). This would destroy the proper connectivity of a triangulated surface derived from the set of boxes. If such a connection is detected, one of the two boxes involved is simply removed from the current set of boxes. In fact, the detection of improper connections between boxes is the most complicated part of the program. This part is repeatedly done until no improper connections are detected anymore.

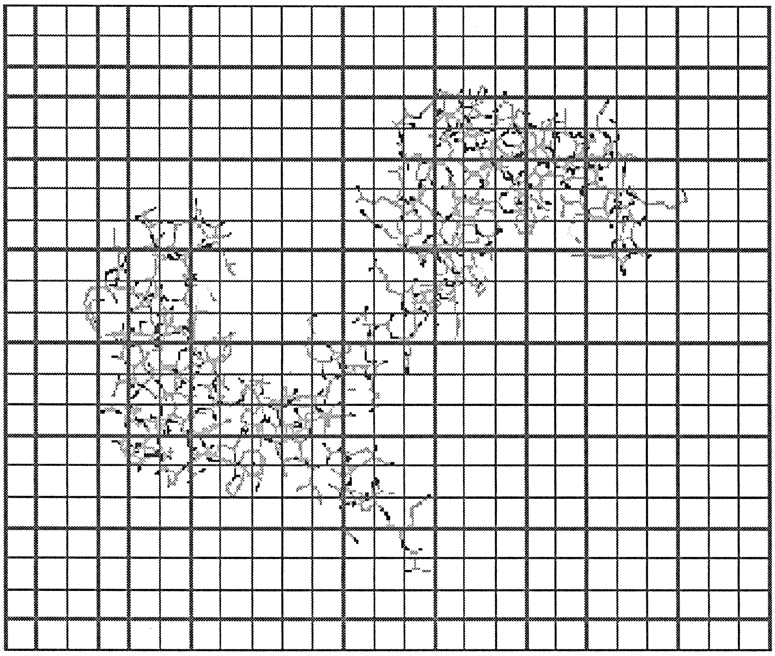

Figure 1. Three-dimensional lattice for calmodulin. Lattice spacing is 0.30 nm. Total number of lattice points is 11440, the total number of boxes corresponds to 9975. Three additional layers of boxes surround the molecule.

After these corrections have been performed, the final set of boxes define the outer contour of the protein (Figure 3). Their free faces will be employed for triangulation after the smoothing process discussed in the next step.

Third step: smoothing

The surface defined by the free faces of the final set of boxes follows only very roughly the shape of the protein. The final mapping procedure (discussed below) would be more difficult to perform. Without the smoothing step, the final triangulated surface is more irregular and occasionally 'stacked'. By repositioning neighbouring box vertices towards each other, the surface will be smoothed such that it will follow the shape of the protein much more accurately. The repositioning of box vertices is explained for the two-dimensional case in Figure 4, the result for calmodulin is demonstrated in Figure 5.

Fourth step: triangulation of free faces

The triangulation of the free faces is accomplished as follows. Each free face of a box is divided in two triangles by using one of the diagonals. For those boxes which may be located in deep 'valleys' (e.g., narrow but solvent-accessible clefts), four triangles are generated from the free face by using both diagonals. The boxes located in valleys may be identified by counting the number of neighbouring boxes. Boxes having 23 or more neighbours are assumed to be in a valley. A box completely surrounded by other boxes has 26 neighbours.

The orientation of the surface is obtained from the free faces of the boxes. Actually, one can set up the topology of boxes (definition of edges and faces in terms of the eight vertices of a box) such that each triangle generated from a free face always has a normal vector pointing into the solvent (outward). Given such a topology, for a triangle (assumed to be flat) the normal vector **n** is given from an outer product according to $\mathbf{n} = \mathbf{r}_{12} \times \mathbf{r}_{13}$, where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. The vector $\mathbf{r}_i$ is the position vector of vertex i of the triangle. The normal vector of a triangle vertex is taken to be the average of normal vectors of all triangles for which the vertex serves as a triangle edge point.

The resulting triangulated surface follows roughly the shape of the protein (Figure 5) and also the solvent-accessible surface. In the final step, the surface simply needs to be adjusted to fit the solvent-accessible surface more accurately.

Fifth step: mapping

This step essentially consists of moving triangle vertices along a line ('spoke') through the triangle vertex

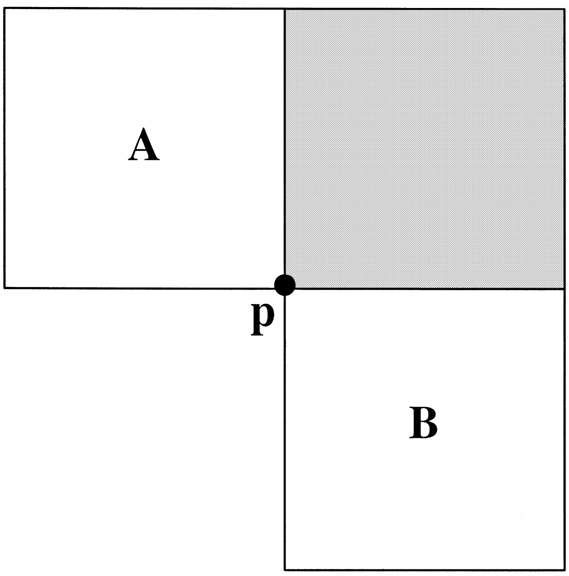

Figure 2. Illustration of improper connections between boxes in two-dimensional space. Improper connections would destroy the proper connectivity between the triangles. In two-dimensional space, the faces of a cube are the sides of a square and triangulation corresponds to splitting the side in two pieces. Suppose now that the shaded square would be omitted, than squares A and B would be connected only through point p. Therefore, the resulting curve (the 'surface', which envelopes the squares A and B) would not be simply connected anymore and the topology is destroyed. Also, the orientation is not clearly defined anymore. The same applies in three-dimensional space: no two boxes should be connected only through a common edge or edge point. They always should have a common neighbour.

towards the solvent-accessible surface (SAS). For this purpose, the NSC (Numerical Surface Calculation) program of Eisenhaber et al. [28] is employed to calculate the dotted surface of the protein under investigation. This program is fast and can be easily incorporated in the triangulation program since the source code (in C) is available upon request. It delivers a set of surface points (dotted surface) and also the total area of the solvent-accessible surface.

The basic idea of the mapping procedure is illustrated in Figure 6. Any triangle vertex p is moved along the spoke such that its final position at p'' (also on the spoke) will be as close as possible to the solvent-accessible surface. A spoke is given from the line through p and a suitable nearby atom (shaded in grey in Figure 6). This atom is the nearest atom within a cylinder with radius r (usually taken to be the radius of a solvent molecule) parallel to the normal vector delivered by the triangulated surface delivered from the previous step. Occasionally, it may happen that no atom can be found initially. For these cases, an increased value for the radius of the cylinder is used until an atom was found. Then, a point p' is determined from the position of s, a nearby surface point delivered by the NSC program. This can be done from the inner product of the position vector of s and p and by applying the cosines rule to the triangle $p'ps$. This

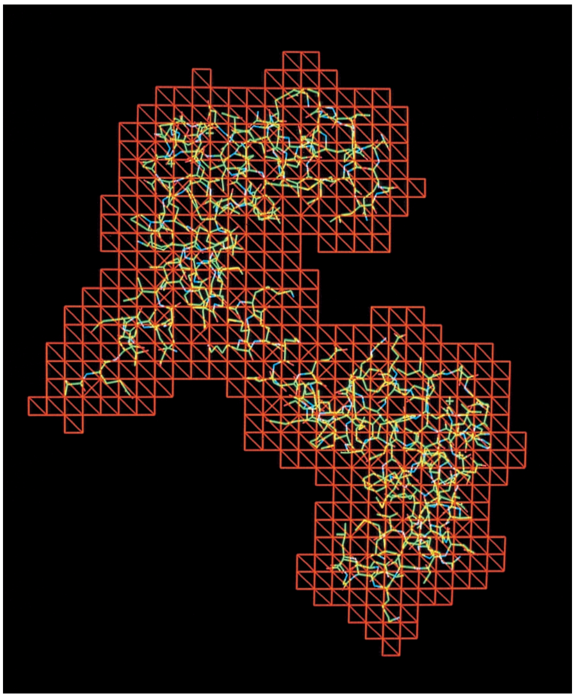

Figure 3. The set of boxes defining the contour of calmodulin. From the original 29160 boxes calculated for a lattice spacing of 0.20 nm, a total of 27511 boxes were removed, leaving 1649 boxes for triangulation. The free faces (pointing towards the solvent) of the boxes were triangulated.

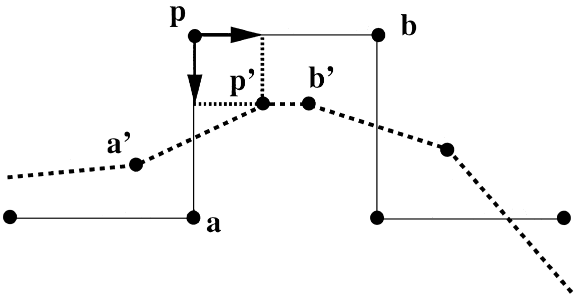

Figure 4. Illustration of smoothing the surface in two-dimensional space. The surface corresponds to the solid line through the points a, p and b. Consider now point p. It is connected to a and b. By moving p in the direction of the arrows, the resultant will generate a new point p'. The same applies to the other points, e.g. point a is repositioned at a'. The resulting 'surface' is smoothed as indicated by the dotted line.

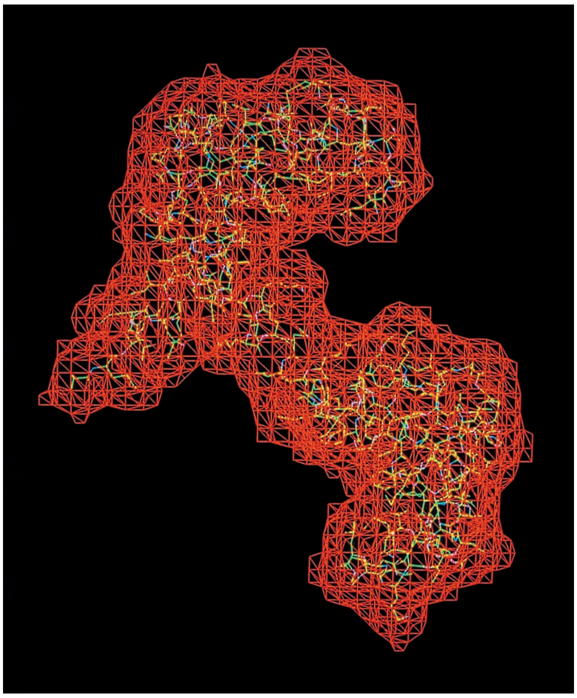

Figure 5. Smoothed triangulated surface for calmodulin. The number of triangles is 6908, the number of vertices is 3456. Lattice spacing was 0.20 nm.

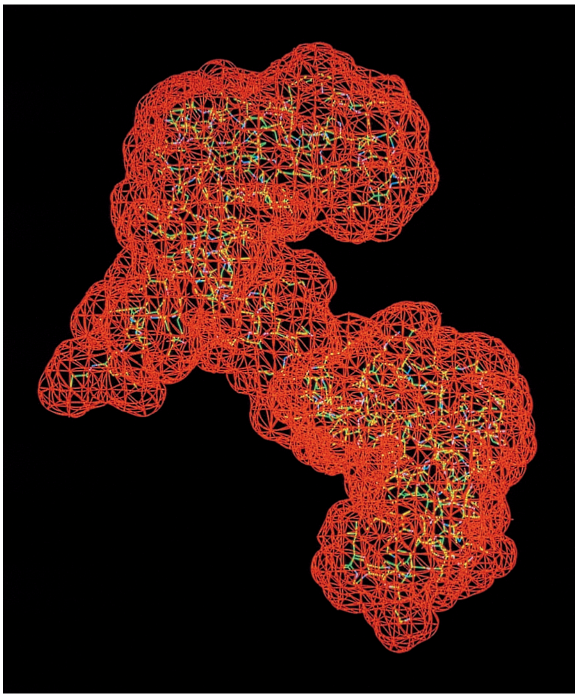

Figure 7. Final triangulated solvent-accessible surface of calmodulin. From 1649 boxes available for triangulation, 6908 triangles and 3456 vertices were generated. Each triangle was taken to be a curvilinear element. That is, each side of the triangle is described by the parametric representation $\mathbf{r}(t) = \mathbf{a} + \mathbf{b}t + \mathbf{c}t^2 + \mathbf{d}t^3$ where the vectors **a**, **b**, **c** and **d** can be derived from the coordinates of the two endpoints of the curve and the corresponding normal vectors at these endpoints [6b]. The parameter t goes from 0 to 1. The curve was divided in 10 pieces according to t=0, 0.1,…,1. Each piece was separately drawn as a straight line between the two endpoints as computed from $\mathbf{r}(t)$.

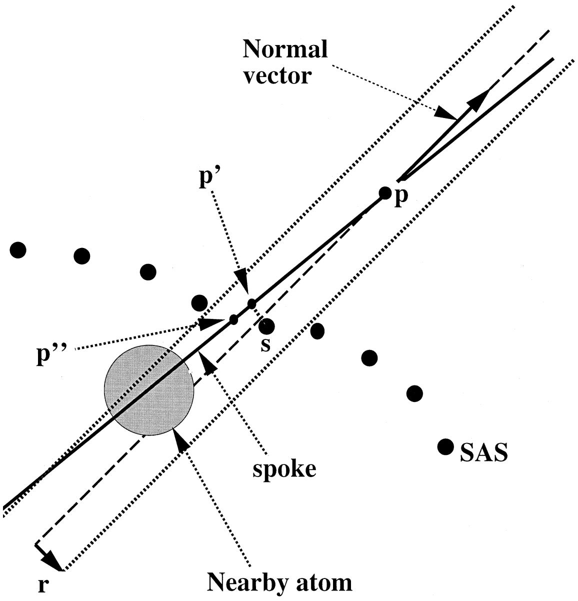

Figure 6. Illustration of the mapping of a triangle vertex onto the solvent-accessible surface in two-dimensional space. Point p is mapped onto point p''. The grey circle refers to a nearby atom, s is a nearby surface point and SAS stands for solvent-accessible surface. See the text for a detailed explanation.

will deliver the length of $p'p$ from which the location of p' can be calculated. Subsequently, the location of p' along the spoke is iteratively adjusted until the distance between a new point p'' and the nearby atom is about equal to the sum of the van der Waals atom radius and solvent radius. So, p'' is located on the solvent-accessible surface of the atom. The calculation of the point of intersection between the spoke and the set of points delivered by NSC may be computed analytically as well, since the dotted surface is essentially located on parts of the surfaces of a set of spheres. If the new point p'' is too far away from p', the new point is rejected and p'' will correspond to p'.

After the mapping, the unit normal vectors for each triangle are recalculated, which in turn are used to compute the unit normal vectors at the triangle vertices, in the same way as described earlier. The final triangulated solvent-accessible surface for calmodulin is shown in Figure 7.

It should be noted that the triangulation method does not require that the triangles are curvelinear elements. They are simply taken to be flat. When the triangulated surface is utilized by a BEM program, it depends on the BEM algorithm whether the triangles are taken to be curvelinear or flat. In Figure 7, the triangles have been drawn as curvelinear elements in accordance with the BEM algorithm in References [6b, 9] which ensures also that the normal vector is continuous across the surface.

To test the algorithm, the triangulation procedure has been applied to several molecules. Calculations were performed for a dipeptide (Lys-His) in the cis-conformation, a 25 residue long helical peptide (this peptide has one cysteine, three arginines and twenty alanines), calbindin (entry code 3ICB [29] in the Brookhaven Protein Data Bank [26]), lysozyme (entry 1LZT [30]), calmodulin (entry 1OSA [25]) and cutinase (entry 1CUS [31]). The dipeptide and the peptide were built with WHATIF [32] and the Insight II Biopolymer module [33] respectively, while van der Waals atomic radii were taken from Insight$^{©}$ II (for the peptide, calmodulin and cutinase) and Gromacs [34] (for the dipeptide and lysozyme, the radius of a hydrogen atom was taken to be 0.12 nm instead of 0).

Calculations have been performed for several values of the lattice spacing ranging from 0.16 to 0.30 nm. The total areas of the triangulated surfaces have been calculated and compared with the ‘real’ value as computed by the NSC program [28]. The total area was calculated by means of a numerical integration of a surface integral for which the surface is treated curvelinearly [6b,9].

Results

Table 1 summarizes the results. For all molecules, very reasonable values for the total area are found, especially for the smaller values of the lattice spacing. The relative errors are not bigger than 12%, while the error is around 1% when the lattice spacing is 0.16 (for the two smaller molecules, the error is almost zero). Clearly, an increase of the lattice spacing tends to underestimate the total area, since the total number of triangles decreases. Generally, the errors for the proteins are somewhat larger. Protein surfaces contain more (small) clefts which are more difficult to triangulate properly.

Figure 8 displays the correlation between the total areas according to the NSC program and the triangulation program for two lattice spacing values (0.30 and 0.16 nm). For both values, correlation coefficients are close to 1, the slope is 0.90 and 1.00 for 0.30 and 0.16, respectively. The value of 1.00 indicates almost full agreement between the NSC program and the triangulation algorithm. This figure together with Table 1 indicates that the mapping onto the NSC dotted surface is performed correctly for lattice spacings smaller than, say, 0.20 nm.

Figure 9 displays a typical distribution for the side length of triangles for the final triangulated surface (calbindin, lattice spacing 0.20 nm, number of triangles is about 4000). The lengths were calculated for flat triangles, so the triangle sides are considered to be straight lines. The distribution averages around 0.12 or so and has a Gaussian-like shape. The largest side is about 0.40 nm, but very small lengths are encountered too. For comparison, the distribution of side lengths (calculated in the same way) for a triangulated surface determined by SMART with about the same number of triangles is given also. As can been seen, the distributions are quite comparable, although SMART displays somewhat shorter edges.

An additional check on the quality of the triangulated surface is based on the Euler characteristics $\chi = v + t - e$ which should be equal to 2 for any closed triangulated surface in three-dimensional space (v is the number of vertices, t is the number of triangles and e is the number of edges). Of all 48 surfaces calculated, in four cases (see the caption to Table 1), this condition was not satisfied. That would indicate a hole in the surface. A triangulated surface which does not satisfy the Euler characteristics can obviously not be used for BEM applications.

Discussion

The triangulation method described in this paper is relatively simple and flexible. The flexibility arises since the number of triangles can be easily adjusted via the lattice spacing. It is simple because it can be implemented without great difficulties. In addition, the method has been proved to be consistent as shown in Figure 8 and Table 1 in the sense that the mapping onto the NSC dotted surface is performed well. In part, the success of the triangulation method can be explained from considering the triangulated surface as a smooth entity with a continuous normal vector. This will im-

Table 1. List of total areas (nm^2) for all tested molecules as computed by the triangulation program

Lattice	Lys-His	Peptide	Calbindin	Lysozyme	Calmodulin	Cutinase
0.16	6.28(0.21)	21.62(0.37)	47.34(1.98)	68.49(1.56)	102.50(0.86)	85.89(1.22)
0.18	6.31(0.77)	21.37(0.78)	46.51(3.70)	67.75(2.62)	101.60(0.03)	85.58(0.86)
0.20	6.23(0.49)	21.39(0.69)	47.07(2.54)	67.79(2.56)	100.30(1.33)	84.37(0.56)
0.22	6.23(0.62)	21.11(2.02)	46.38(3.96)	66.53(4.38)	98.48(3.10)	82.96(2.23)
0.24	6.15(1.84)	20.81(3.41)	45.08(6.65)	64.82(6.83)	98.16(3.41)	81.57(3.87)
0.26	5.87(6.27)	20.29(5.80)	44.08(8.74)	63.96(8.07)	95.56(5.97)	79.85(5.89)
0.28	5.77(7.78)	20.06(6.86)	43.73(9.46)	63.61(8.58)	94.46(7.06)	78.39(7.61)
0.30	5.89(5.95)	19.82(7.98)	42.68(11.63)	61.93(10.98)	92.45(9.03)	76.70(9.61)
NSC	6.27	21.54	48.30	69.57	101.63	84.85

The total number of triangles for the given lattice spacing (in nm) range from 152 to 668 for the dipeptide, from 622 to 2292 for the peptide, from 1226 to 4836 for calbindin, from 1698 to 6994 for lysozyme, from 2656 to 10802 for calmodulin and from 2264 to 9066 for cutinase. The total areas as computed by NSC [25] are given in the bottom line of the Table and were obtained using 100 points per atom. The values between brackets refer to relative errors calculated with respect to the NSC result according to $(A - A_{\mathrm{NSC}}) * 100/A_{\mathrm{NSC}}$, where A and A_{NSC} are the areas calculated by the triangulation program and NSC, respectively. In four cases, the triangulated surface did not satisfy the Euler characteristics (see text): peptide for a lattice space of 0.20 nm, calbindin for 0.22 nm, lysozyme for 0.26 nm and cutinase for 0.18 nm.

prove the quality of the surface since it will follow the solvent-accessible surface quit accurately as is seen from Figure 7. Therefore, rather good estimates of the total area while employing a low number of triangles can be obtained. Figure 9 indicates a Gaussian distribution for the triangle side length which points also to the existence of very small sides. These sides are located in clefts where the triangulation algorithm employs more triangles to describe the solvent-accessible surface. In four cases, it was noted that the triangulated surface did not satisfy the Euler characteristics. This was noted also for other triangulation methods [12, 35]. Clearly, such a triangulated surface cannot be used for any BEM application. It appears that this results from the removal of a box when an improper connection between boxes is detected (step 2). In practice, it is always possible (as may be clear from Table 1) to use a somewhat different value for the lattice spacing such that the new surface does satisfy the Euler characteristics.

The flexibility of the method is of importance for the BEM, since the number of vertices and triangles can be reduced without greatly affecting the quality of the triangulated surface. Therefore, the program is very suitable for BEM applications which include the effect of ionic strength. For these cases the dimension of a dense matrix involved in the computation corresponds to twice the number of vertices. For example, a triangulated surface consisting of 6000 triangles would have 3002 vertices. Therefore, the dimension of the matrix would be 6004 × 6004. In single precision, this would require about 140 MB (70 MB if ionic strength is not included). For comparison, the SMART program delivers for calmodulin between 11000 and 46000 triangles [12, 35] while in this work between 2656 to 10802 triangles were generated. On the other hand, SMART will deliver an almost constant value for the total area while the triangulation method presented in this work tends to underestimate the total area somewhat (with respect to the NSC program) when a coarser triangulated surface is employed. Another example: the algorithm of Sanner et al. [11] delivers for a small protein like crambin with 46 residues almost 5000 triangles; this work gives for calbindin with 75 residues between 1226 and 4836 triangles.

The question arises if there is need for a large number of triangles to describe the solvent-accessible surface for boundary element applications. First of all, although continuum electrostatics is successful when applied to solvated molecules [36], it is an approximation to the real potential of mean force of a solvated protein. Secondly, due to the variations of commonly employed van der Waals radii (e.g., Gromacs [34] uses 0.15 nm for the carbon atom while Insight II [33] employs 0.18 nm), an exact location of the solvent-accessible surface is not defined. In this context, it is interesting to note that Zhexin et al. [16] found cred-

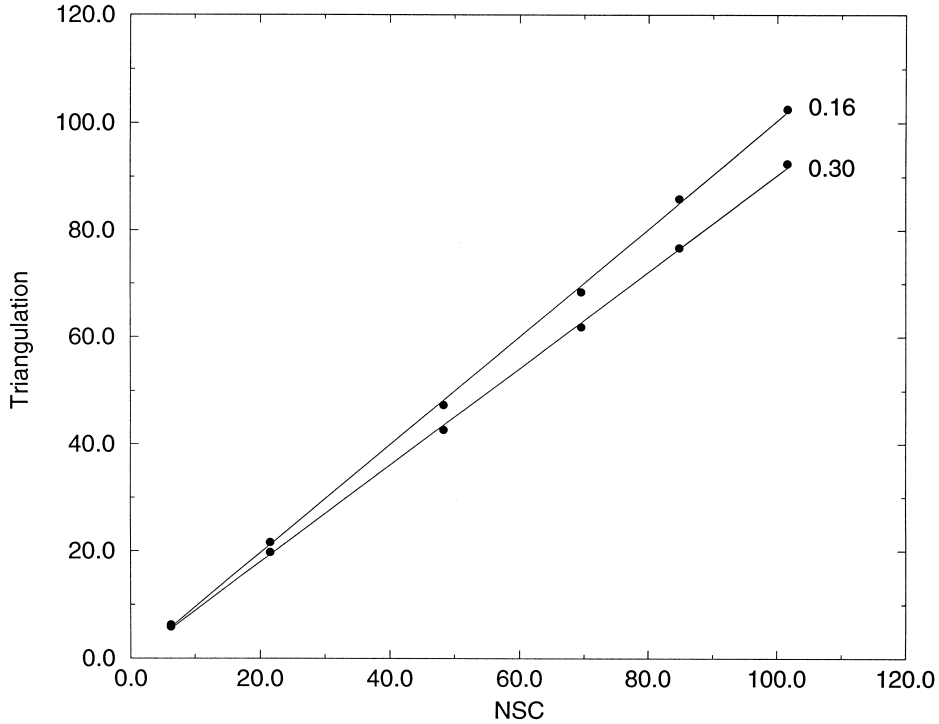

Figure 8. Correlation between the areas computed by NSC [28] and the triangulation program for a lattice spacing of 0.16 and 0.30 nm. For both cases, the correlation coefficient is virtually equal to 1. The slopes are 0.90 and 1.00 for 0.30 and 0.16, respectively.

ible results for the acid-dissociation constants (pK_a) for titrating sites in bovine trypsin inhibitor using only 400 triangles. Recently, Juffer et al. [24] obtained accurate estimates of pK_a values for four different proteins (including lysozyme and calbindin) employing the BEM in combination with the triangulation method presented in this work (the lattice spacing was 0.18 nm for all proteins employed in [24]: in all cases the triangulated surface did satisfy the Euler characteristics). This indicates the quality of the triangulation method. It is essential that the triangulated surface follows the solvent-accessible surface as closely as possible.

In step 2 of the triangulation procedure, it was stated that boxes located in cavities would be the cause of isolated surfaces within the protein molecule which destroys the proper connectivity of the triangulated surface. The decision to exclude these isolated surfaces is based on the following argument. The purpose of a triangulated surface in any BEM is to distinguish between bulk solvent and protein. The protein region may contain solvent molecules specifically bound to the protein molecule, e.g., water molecules trapped in a cavity. Since it is not expected that these solvent molecules behave as if they are in the bulk phase, it is desirable to include these molecules in the detailed description of the protein [9]. In addition, if one insists on using an isolated surface together with the solvent-accessible triangulated surface, this will result in a modification of the BEM formulation: e.g., the electrostatic potential in the protein region (outside any cavity which for this purpose is considered as a region with a different dielectric constant as the protein) is now to be expressed in terms of the potential and/or the normal component of the electric field on the solvent-accessible surface *and* all cavity surfaces. This complicates the calculation of the electrostatic potential considerably and decreases the efficiency of the algorithm. This all does not deny the importance of cavities for the structure and function of proteins [37].

Although the triangulation method originally was designated for the BEM, the output of the program is such that it can be easily converted into any format acceptable for a molecular graphics program such as WHATIF [32] or Insight II [33]. The triangulation program delivers the coordinates of vertices, the vector components of unit normal vectors at vertices and the topology for each triangle (indices of vertices serving

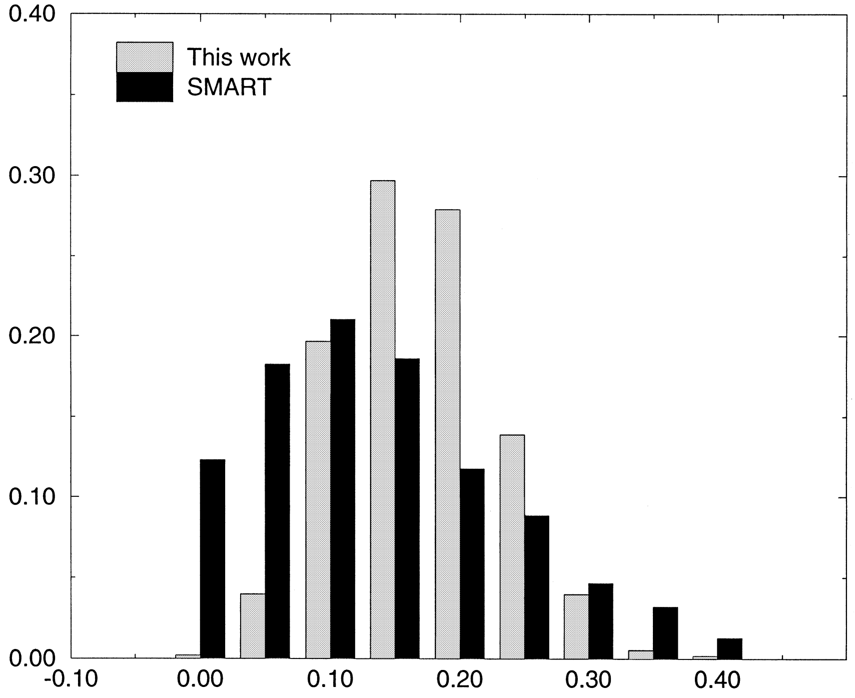

Figure 9. An example of a typical triangle edge length distribution. The figure applies to calbindin using 0.20 nm for the lattice spacing. For comparison, the distribution according to SMART [12] is presented also. SMART was employed with an angle parameter of 75 degrees (coarse representation). Both triangulated surfaces consist of about 4000 triangles. Although both surfaces are quite similar as can be seen from a visual inspection, strictly spoken, one cannot compare both surfaces so well: SMART calculates a somewhat different surface than the NSC program does. Therefore, there is a difference in the total area. The total area of the SMART surface is about 39 nm^2, while the total area of the triangulated surface according to this work is 47 nm^2 (Table 1). The main difference between the two surfaces is to be found in the clefts.

as edge point for a triangle). This has been illustrated in Figure 7 in which WHATIF was used to display the triangulated surface as a smooth entity together with the protein. Therefore, the program is suitable for molecular surface display purposes also.

The efficiency of the triangulation program is reasonable. For BEM, the triangulation itself is not the time determining step (it is totally negligible with respect to the total time required to perform a BEM calculation), so it is not essential for BEM application. Nevertheless, for the triangulation of calmodulin using a lattice spacing of 0.20 nm, about 30 min of CPU time on a R8000 Silicon Graphics computer with 224 MB main memory is required (calmodulin is the worst case, since it has the largest area of the proteins employed in this work). That is not extremely fast. The most time consuming steps are the box removal and mapping part. The latter can be made more efficient by reducing the number of surface points for the dotted surface.

Conclusions

In this paper, a flexible, relatively simple and consistent triangulation method for the solvent-accessible surface of a protein and other macromolecules has been presented. The corresponding computer program is very suitable to work with continuum electrostatic programs based on the boundary element method, since the number of triangles can be easily adapted to meet computer memory limits and efficiency requirements without greatly affecting the quality of the triangulated solvent-accessible surface. The program will also work fine with any molecular graphics program since the simple output file can be easily converted to a format required by the molecular graphics program. In addition, the surface can be viewed as smooth entity by simple means.

Acknowledgements

This work was supported by grants for A.H.J. from the Alberta Heritage Foundation for Medical Research

(AHFMR) and the Human Frontier Science Program (HFSP). H.J.V. is the recipient of a Scientist Award of the AHFMR. The computing facilities used were purchased through a grant from the AHFMR. We are indebted to Dr. Randy Zauhar from TRIPOS for making the SMART program available to us. We also thank Dr. Dean McIntyre for preparing the colour prints.

References

1. Harvey, S., Proteins Struct. Funct. Genet., 5 (1989) 78.
2. Davids, M.E. and McCammon, J.A., Chem. Rev., 90 (1990) 509.
3. Mantripragada, B. and March, D., In Watts, A. (Ed.) Protein-Lipid Interactions, Elsevier, Amsterdam, 1993, pp. 127–162.
4. Warwicker, J. and Watson, H.C., J. Mol. Biol., 157 (1982) 671.
5. Bashford, D. and Karplus, M., Biochemistry, 29 (1990) 10219.

6a. Zauhar, R.J. and Morgan, R.S., J. Mol. Biol., 186 (1985) 815.
6b. Zauhar, R.J. and Morgan, R.S., J. Comput. Chem., 9 (1988) 171.
7. Rashin, A.A., J. Phys. Chem., 94 (1990) 1725.
8. Yoon, B.J. and Lenhoff, A.M., J. Comput. Chem., 11 (1990) 1080.
9. Juffer, A.H., Botta, E.F.F., van Keulen, B.A.M., van der Ploeg, A. and Berendsen, H.J.C., J. Comput. Phys., 97 (1991) 144.
10. Bharadwaj, R., Windemuth, A., Sridharan, S., Honig, B. and Nicholls, A., J. Comput. Chem., 16 (1995) 898.
11. Sanner, M.F., Olson, A.J. and Spehner, J.-C., Biopolymers, 38 (1996) 305.
12. Zauhar, R.J., J. Comput.-Aided Mol. Design, 9 (1995) 149.
13. Zauhar, R.J. and Morgan, R.S., J. Comput. Chem., 11 (1990) 603.
14a. Connolly, M.L., J. Appl. Crystallogr., 18 (1985) 499.
14b. Connolly, M.L., J. Mol. Graphics, 11 (1993) 139.
15. Silla, E., Villar, F., Nilsson, O., Pascual-Ahuir, J.L. and Tapia, O., J. Mol. Graphics, 8 (1990) 168.
16. Zhexin, X., Yunyu, S. and Yingwu, X., J. Comput. Chem., 16 (1995) 512.
17a. Stollnitz, E.J., DeRose, T.D. and Salesin, D.H., IEEE Comput. Graph. Appl., 15(3) (1995) 76.
17b. Stollnitz, E.J., DeRose, T.D. and Salesin, D.H., IEEE Comput. Graph. Appl., 15(4) (1995) 75.
18a. Duncan, B.S. and Olson, A.J., Biopolymers, 33 (1993) 219.
18b. Timchenko, A.A., Galzitskaya, O.V. and Serdyuk, I.N., Proteins Struct. Funct. Genet., 28 (1997) 194.
19a. Certain, A., Popovic, J., DeRose, T., Duchamp, T., Salesin, D. and Stuezle, W., In Proceedings of SIGGRAPH 96, ACM SIGGRAPH, New Orleans, p. 91.
19b. Hoppe H., In Proceedings of SIGGRAPH 96, ACM SIGGRAPH, New Orleans, p. 99.
19c. Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F. and Wright, W., In Proceedings of SIGGRAPH 96, ACM SIGGRAPH, New Orleans, p. 119.
19d. Hoppe, H., In Proceedings of SIGGRAPH 97, ACM SIGGRAPH, Los Angeles, p. 189.
19e. Luebke, D. and Erikson, C., In Proceedings of SIGGRAPH 97, ACM SIGGRAPH, Los Angeles, p. 199.
19f. Garland, M. and Heckbert, P., In Proceedings of SIGGRAPH 97, ACM SIGGRAPH, Los Angeles, p. 209.
19g. Popovic, J. and Hoppe, H., In Proceedings of SIGGRAPH 97, ACM SIGGRAPH, Los Angeles, p. 217.
20. Akkiraju, N. and Edelsbrunner, H., Discrete Appl. Math., 71 (1996) 5.
21. Lorensen, W.E. and Cline, H.E., Comput. Graphics, 21 (1987) 163.
22. Lee, B. and Richards, F.M., J. Mol. Biol., 55 (1971) 379.
23. Vogel, H.J., Biochem. Cell Biol., 72 (1995) 357.
24. Juffer, A.H., Argos, P. and Vogel, H.J., J. Phys. Chem., 101 (1997) 7664.
25. Ban, C., Ramakrishnan, B., Ling, K-.Y., Kung, C. and Sundaralingam, M., Acta Crystallogr., D50 (1994) 50.
26. Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Brice, M.P., Rodgers, J.R., Kennard, O., Shimanouchi, T. and Tasumi, M., J. Mol. Biol., 112 (1977) 535.
27a. Babu, Y.S., Sack, J.S., Greenbrough, T.J., Bugg, C.E., Means, A.R. and Cook, W.J., Nature, 315 (1985) 37.
27b. Babu, Y.S., Bugg, C.E. and Cook, W.J., J. Mol. Biol., 204 (1988) 191.
28a. Eisenhaber, F. and Argos, P., J. Comput. Chem., 14 (1993) 1272.
28b. Eisenhaber, F., Lijnzaad, P., Argos, P., Sander, C. and Scharf, M., J. Comput. Chem., 16 (1995) 273.
29. Szabenyi, D.M.E. and Moffat, K., J. Biol. Chem., 261 (1986) 8761.
30. Hodson, J.M., Brown, G.M. and Sieker, L.C., Acta Crystallogr. B46 (1990) 54.
31. Martinez, C., De Geus, P., Lauwereys, M. and Matthyssens, G., Nature, 356 (1992) 616.
32. Vriend, G., J. Mol. Graphics, 8 (1990) 52.
33. Insight II, version 95.0, October 1995, Biosym/MSI, San Diego, CA.
34. van der Spoel, D., van Buuren, A.R., Apol, E., Meulenhoff, P.J., Tieleman, D.P., Sijbers, A.L.T.M., van Druunen, R. and Berendsen, H.J.C., Gromacs User Manual, State University of Groningen, 1996.
35. Zauhar, R.J. and Varnek, A., J. Comput. Chem., 17 (1996) 864.
36. Honig, B., Sharp, K. and Yang, A.-S., J. Phys. Chem., 97 (1993) 1101.

37a. Hubbard, S.J. and Argos, P., Curr. Opin. Biotech., 6 (1995) 375.
37b. Hubbard, S.J. and Argos, P., J. Mol. Biol., 261 (1996) 289.