MSEED: A Program for the Rapid Analytical Determination of Accessible Surface Areas and Their Derivatives

G. Perrot,* B. Cheng, K.D. Gibson, J. Vila,† K.A. Palmer, A. Nayeem, B. Maigret,‡ and H.A. Scheraga**

Baker Laboratory of Chemistry, Cornell University, Ithaca, New York 14853-1301

Received 9 April 1991; accepted 15 July 1991

An algorithm for the rapid analytical determination of the accessible surface areas of solute molecules is described. The accessible surface areas as well as the derivatives with respect to the Cartesian coordinates of the atoms are computed by a program called "MSEED," which is based in part on Connolly's analytical formulas for determining surface area. Comparisons of the CPU time required for MSEED, Connolly's numerical algorithm DOT, and a program for surface area determination (ANA) based on Connolly's analytical algorithm, are presented. MSEED is shown to be as much as 70 times faster than ANA and up to 11 times faster than DOT for several proteins. The greater speed of MSEED is achieved partially because nonproductive computation of the surface areas of *internal* atoms is avoided. A sample minimization of an energy function, which included a term for hydration, was carried out on MET-enkephalin using MSEED to compute the solvent-accessible surface area and its derivatives. The potential employed was ECEPP/2 plus an empirical potential for solvation based on the solvent-accessible surface area of the peptide. The CPU time required for 150 steps of minimization with the potential that included solvation was approximately twice as great as the CPU time required for 150 steps of minimization with the ECEPP/2 potential only.

INTRODUCTION

In empirical energy calculations on polypeptides and proteins in water, it is necessary to include the effect of the solvent on the conformation.[1] Since inclusion of explicit water molecules greatly increases the computational cost, simplified models, e.g., the solvent-shell model[1-5] or models based on exposed surface area,[1,6-8] have been developed. Even these models, however, are computationally expensive when attempts are made to minimize the combination of empirical potentials and solvation free energies in searching for low-energy structures of proteins. A resolution of this difficulty, utilizing the algorithm of Perrot and Maigret[9] for determination of surface area, is now available. This algorithm, which is based in part on earlier work of Connolly,[10-13] makes use of a routine, MSEED, which rapidly traces out the surface and computes its area, together with analytical derivatives of that area. In this article, we describe MSEED and illustrate its applicability to some polypeptide and protein computations.

MSEED identifies the solvent-exposed surface area of the protein, and then employs an empirical potential based on the solvent-accessible surface area to estimate the free energy of solvation. There are basically three definitions of protein "surface" currently in use.[1] All these definitions[14,15] represent the protein as a union of spheres, with the spheres having the van der Waals radii of the constituent atoms. The surfaces are defined by rolling a spherical test probe over the atomic spheres. The *molecular* surface is traced out by the direct contact between the surface of the probe and the surface of the protein. Where the probe cannot make contact, the bottom-most part of the probe is used to define the molecular surface, e.g., the saddle-shaped surface in Figure 1. If the radius of the probe is set to zero, the molecular surface becomes the *van der Waals* surface. The *solvent-accessible* surface is the surface traced by the center of the probe as the probe is rolled over the van der Waals surface of the molecule[14] (Fig. 1). In this article, we will be concerned only with the solvent-accessible surface. However, it should be noted that the analytical determination of the accessible and the molecular surfaces is identical; therefore, MSEED could, in principle, be used to calculate molecular surface

*On leave from I.B.M. France, Service 3393, 40 rue Dussoubs, 75002 Paris, France, 1990.

†On leave from the National University of San Luis, Faculty of Science and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Instituto de Matemática Aplicada—San Luis. Ejécito de Los Andes 950, (5700) San Luis, Argentina, 1988–1991.

‡Laboratoire de R.M.N. et de Modélisation Moléculaire, Institut Lebel, Université L. Pasteur, 4 rue Blaise Pascal, 67000 Strasbourg, France.

**Author to whom all correspondence should be addressed.

Journal of Computational Chemistry, Vol. 13, No. 1, 1–11 (1992)

CCC 0192-8651/92/010001-11$04.00

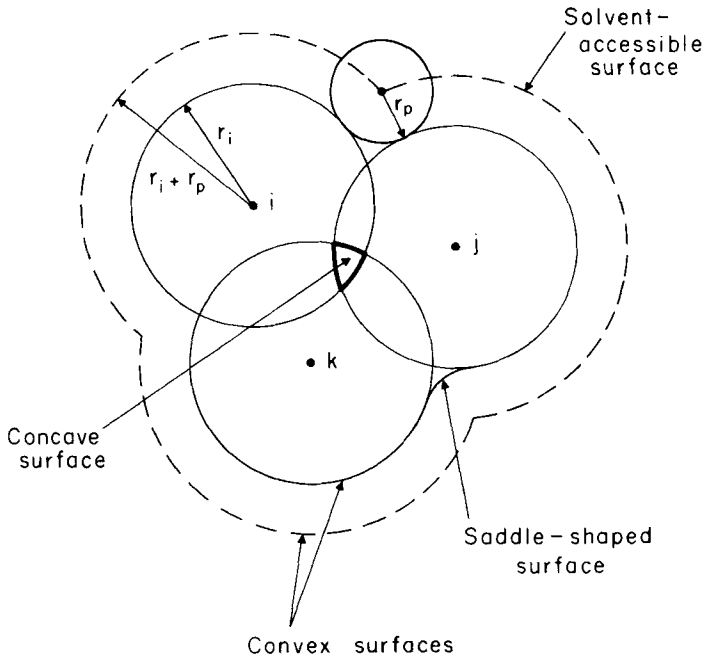

Figure 1. The solvent-accessible surface (shown as a dashed line) for atoms i, j, and k with a probe radius of r_p. The accessible surface is composed of three pieces of convex surface (three convex faces). The different components (convex, saddle-shaped, concave) of the molecular surface are also shown. The concave surface (outlined in heavy black line) is formed by the bottom of the probe when the probe touches all three spheres (i, j, k) simultaneously.

areas with slight modifications. On the other hand, MSEED cannot be used to determine van der Waals surfaces (see Results).

The usual procedure for applying empirical solvation models to peptides and proteins is, first, to find a low-energy conformation using a potential that does not include solvent, such as the Empirical Conformational Energy Program for Peptides (ECEPP/2).[16,17] Next, the accessible surface (or shell volume) of the energy-minimized structure is calculated, and the solvation free energy is estimated by summing the solvation free energy terms that are a function of exposed surface areas (or shell volumes). Such estimates of the free energy of the system may be used to predict protein structures, for example, in modeling short segments (loops) of homologous proteins.[18] It has been suggested that hydration effects can be included in energy-minimization or molecular dynamics simulations by calculating the surface area during each step of the simulation.[8,19,20] In order to make effective use of such an approach, it is necessary to have a rapid and accurate method for calculating the accessible surface area. Furthermore, to utilize this augmented potential effectively as part of a scheme for energy-minimization or molecular dynamics, the derivatives of the accessible surface must be computed analytically.

Until recently, no algorithm for determining the accessible surface area with sufficient speed and accuracy was available. The fastest approximate methods[21,22] for estimating the accessible surface area are generally not accurate enough to be combined with a good solvation potential function. A numerical method that uses a uniform distribution of dots to represent the surface[10] rivals the fastest of the approximate methods for estimating surface area, but only when low dot densities are used. Even at higher dot densities, this method is not as accurate as the analytical method of Connolly.[8,12] Additionally, the calculation of the derivatives of the area with respect to atomic positions using the numerical method is slow and inaccurate (unpublished observations in this laboratory). Until now, Connolly's analytical method for calculating surface areas[12] appeared to be the only algorithm of sufficient accuracy to be incorporated into energy-minimization routines. Unfortunately, the calculation of accessible surface areas using this program is significantly more expensive than is the computation of the energy (without inclusion of the solvent), thus rendering its use impractical in applications involving conformational searches.

By contrast, MSEED, the program presented here, computes solvent-accessible surface areas and solvation free energies in less CPU time than that required for computation of intramolecular potentials. Therefore, estimates of hydration free energy can now be incorporated into methods for exploring large parts of conformational space, without increasing the computation time greatly. The addition of

terms for hydration free energy is unlikely to change the overall number of local minima on a typical energy surface significantly.[23] Accordingly, the efficiency of search methods, which include the effects of hydration, should be comparable to that of methods that use potentials that do not include solvent.

MSEED computes surface areas using equations that are the same as, or similar to, the equations presented by Connolly[12] and Richmond.[24] However, the CPU-intensive portion of Connolly's algorithm is the determination of the portion of the surface which is defined as solvent-accessible. In Connolly's method, all convex faces are checked for solvent accessibility, including those faces that are buried in the interior of the molecule. A program that follows Connolly's method rigorously[8,12] spends a large fraction of its time generating the intermediate quantities necessary for the calculation of interior surface areas that are ultimately found to be zero. For globular proteins, the problem becomes more severe as the size of the molecule increases, because a larger fraction of the convex faces will not be accessible to the surface.

By contrast, MSEED eliminates the unnecessary computation of intermediate results by considering only those convex faces that are *on the accessible surface.* To do this, MSEED takes advantage of the fact that each convex face on the surface is defined by the set of arcs of circles of intersection of two spheres, and of points of intersection of three spheres that surround it. This set of arcs and points encloses a curvilinear polygon on the surface of one sphere; the entire accessible surface is made up of contiguous polygons, on adjacent spheres, that are defined in this manner. By exploring only these arcs and points of intersections, and ignoring all other arcs and points of intersection that lie *inside* the molecule, MSEED reduces the computational effort needed to define the surface by a very large factor, as will be shown below.

The search method used by MSEED is not completely general. To find the *surface* polygons, MSEED searches for points of intersection of three spheres (triple points or vertices). However, some parts of the solvent-accessible surface may not have any triple points; for example, the intersection of only two spheres generates two convex surfaces, each bounded by a complete circle of intersection (Fig. 2). There are no triple points on this circle of intersection. Because MSEED carries out a search by proceeding from one triple point to the next, it will be unable to reach this circle of intersection, and will not calculate an accurate total surface area in such a case (another obvious example of a structure for which surface area cannot be calculated using MSEED is a completely free sphere). Fortunately, such cases occur rarely in surface-area calculations for proteins when using typical van der

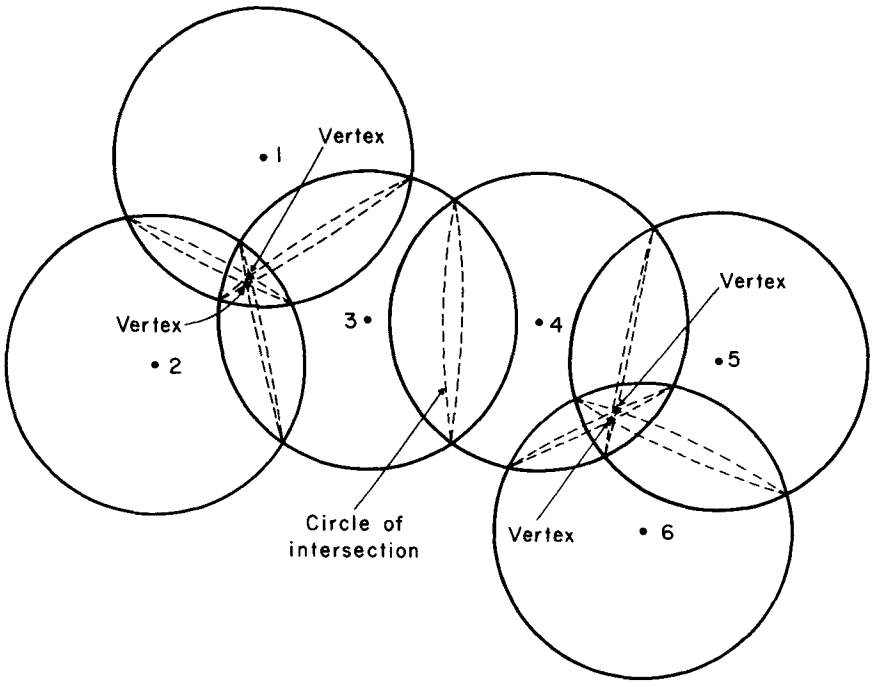

Figure 2. Circle of intersection formed by the intersection of the surfaces of two spheres. Atoms 3 and 4 form a circle of intersection shown in dashed line. Vertices are the points of intersection of the surfaces of three spheres. The triplet of atoms (1, 2, 3) and (4, 5, 6) each form two vertices per triplet. The short arcs of the circles of intersection that join two vertices are referred to as edges. No edges connect the vertices formed by atoms 1, 2, and 3 with the vertices formed by atoms 4, 5, and 6.

Waals radii[7] for the atoms, together with a probe having a radius of 1.4 Å. However, if the combined van der Waals radii plus probe radius are too small (about 2 Å), there may be many intersections of only two atoms, and the corresponding areas will not be calculated properly. Occasionally, a molecule may have two (or more) domains connected by only two overlapping spheres. Because there are no vertices connecting the two domains, the search procedure cannot cross this barrier (Fig. 2) and the resulting surface area will be incorrect. This problem would be overcome if the MSEED program were to restart the search with a new seed in the unexplored domain; however, the current version of MSEED contains no provision for this. Fortunately, this problem has not been observed under the conditions discussed in this article. In the Results section, we demonstrate that, for applications utilizing a probe having a radius of 1.4 Å, the error in the calculation of solvent-accessible surface areas arising from this cause is so small, and the saving of CPU time is so large, that MSEED is clearly superior for many significant applications to calculations of macromolecular structures. In that section, we compare the performance of MSEED with those of Connolly's dot surface algorithm[10] and a version of Connolly's analytical algorithm (ANA)[8] for determining surface area. In addition, we present timings and some results of ECEPP/2 energy minimizations where hydration free energies were calculated using a surface-based potential[8] and in which the accessible surface areas were computed by using either MSEED or ANA.

METHODS

The major difference between MSEED and Connolly's method[12] is the use of "SEED," which is an algorithm similar to that presented by Perrot and Maigret.[9] A brief summary of Connolly's algorithm will be presented to facilitate comparison with the new algorithm, and the MSEED approach to the determination of solvent-accessible surface areas will then be described. In the following, arcs of circles of intersection of two spheres will be referred to as "edges," and points of intersection of three spheres will be referred to as "vertices."

Connolly's Algorithm

In order to obtain the solvent-accessible surface area, it is first necessary to locate the vertices and edges on the surface. All of the vertices and edges may be found by searching through all possible triplets of spheres (S_i, S_j, S_k) and determining the full set of double and triple intersections (a sphere j intersects, i.e., is a neighbor of, sphere i if the distance between their centers is less than the sum of their radii). Next, it is necessary to distinguish the subset of vertices and edges which are on the surface. A vertex is on the surface if and only if it is not inside any other sphere S_l where $l \neq i, j$, or k. If n is the total number of spheres (atoms or united atoms), and if k_i is the number of spheres intersecting sphere S_i, and $\langle k \rangle$ is the mean value of $k_{i(i=1,n)}$, then the overall complexity of the search for all vertices scales as $n\langle k \rangle$.[3]

MSEED

The MSEED program operates in two distinct steps. A routine called SEED locates all of the vertices and edges on the accessible surface. A second routine, AREAC, calculates the areas of the convex faces and the Cartesian derivatives of the areas, using the vertices and edges on the surface located by the SEED routine. With the exception of some changes in the presentation of the equations, the intermediate quantities calculated in AREAC are nearly identical to those presented by Connolly.[12]

Step 1

Following Vila et al.,[8] the set of centers of spheres representing atoms and united atoms is first placed into cubes of appropriate size to facilitate the determination of a list of neighbors for each sphere. This constitutes the "initialization step." The next step is to locate a vertex on the accessible surface. This is achieved as follows.

Since the molecule is represented as a finite union of spheres in Cartesian space, the point with the smallest y coordinate must lie on its surface. Call this point P_1, located on sphere S_i, ($= S_1$ in Fig. 3). Each circle of intersection of S_i with one of its neighbors, S_j, contains a point P_j whose y coordinate is the smallest among all points on that circle. Let P_2 be the P_j whose y coordinate is smallest; then clearly P_2 also lies on the surface. The point, P_2, is the "seed" from which the search for vertices and edges begins. To find a vertex on the surface, a search is conducted by starting at P_2, and exploring along the circle of intersection on which P_2 lies in either of the two possible directions until the first vertex is encountered (Fig. 3). It is necessary to search in only one direction. We choose the direction that is right-handed with respect to rotation about the unit vector $\mathbf{D}_{ij}$, where $\mathbf{D}_{ij} = (\mathbf{C}_j - \mathbf{C}_i)/|\mathbf{C}_j - \mathbf{C}_i|$, and $\mathbf{C}_i$ and $\mathbf{C}_j$ are the coordinate vectors of the centers of S_i and S_j. Only the k neighbors of S_i have to be searched in order to find the seed, P_2, and the first vertex. All other vertices on the surface are located in the following manner. Since each vertex is a point on the surface of three spheres, it is also a point of intersection of three edges, and each edge leads to an-

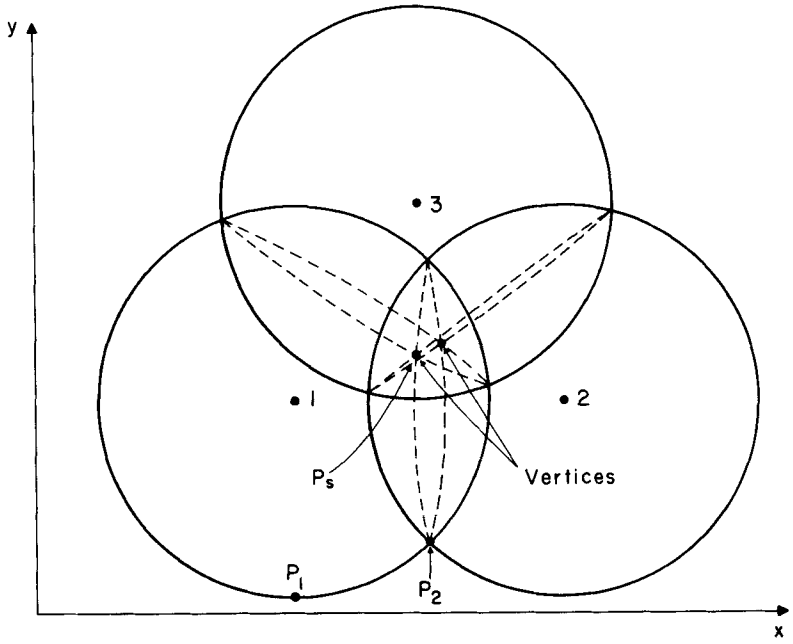

Figure 3. Diagram to locate points P_1 and P_2. Sphere 1 is the seed sphere. Spheres 2 and 3 intersect sphere 1, and circles of intersection have been drawn for all atoms, i.e., spheres 2 and 3, that intersect sphere 1. P_1 is the point with the smallest y coordinate on sphere 1. P_2 is the point on the circle of intersection of spheres 1 and 2 with the smallest y coordinate. P_s is the first vertex.

other vertex (see Fig. 4 for an illustration). All vertices on the surface can be located by exploring along edges leading away from known surface vertices. The search along each edge stops as soon as another vertex is encountered on that edge. Since the surface defined by the edges and vertices, considered as a closed two-dimensional manifold in Euclidean space, is orientable, each edge on the surface needs to be traversed only once. However, as will be mentioned later, numerical instability resulting from insufficient arithmetical precision can cause the surface defined by MSEED to appear to be nonorientable; in this case, an edge may be traversed twice, or may appear to contain only one vertex (see the section entitled "Numerical Stability" under Results and Discussion).

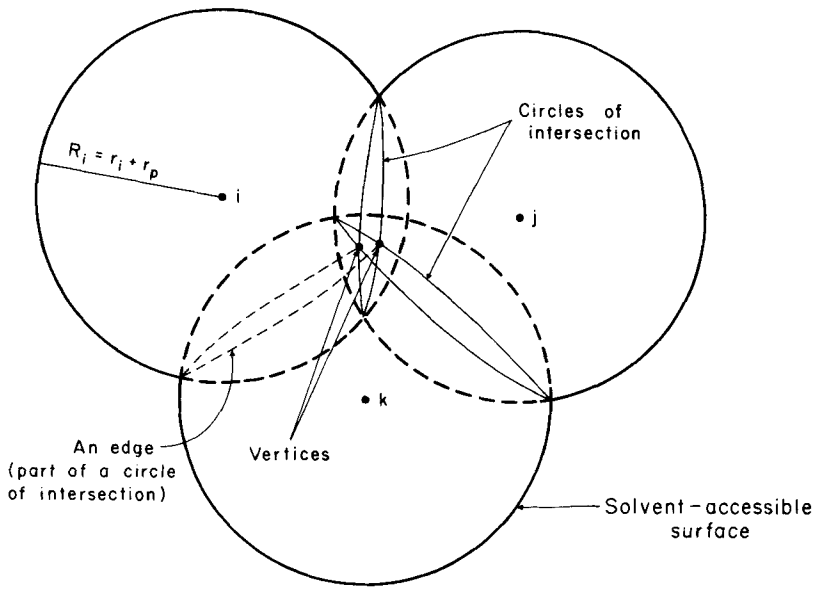

Figure 4. Method to locate vertices. Three circles of intersection cross at a vertex. The radius of the sphere is R_i where R_i is the sum of the van der Waals radius for the atom i and the probe radius.

To find the adjacent surface vertex along an edge, the following procedure is used. The edge is an arc of the circle of intersection of two spheres: let these be S_i and S_j, with centers at the points with coordinate vectors $\mathbf{C}_i$ and $\mathbf{C}_j$. Also, let the current surface vertex be the point of intersection of S_i and S_j with the sphere S_k, and let its coordinate vector be $\mathbf{P}_{ijk}$. Then, if $\mathbf{y}$ and $\mathbf{z}$ are the vectors

$$\mathbf{y} = \mathbf{P}_{ijk} - \mathbf{C}_i \tag{1}$$

$$\mathbf{z} = \frac{(\mathbf{C}_j - \mathbf{C}_i)}{|\mathbf{C}_j - \mathbf{C}_i|} \tag{2}$$

then the vector $\mathbf{u}_k$ defined by

$$\mathbf{u}_k = (\mathbf{y} - (\mathbf{y} \cdot \mathbf{z})\mathbf{z})/|\mathbf{y} - (\mathbf{y} \cdot \mathbf{z})\mathbf{z}| \tag{3}$$

is a unit vector in the plane of $\mathbf{C}_i$, $\mathbf{C}_j$, and $\mathbf{P}_{ijk}$, perpendicular to the line joining the centers $\mathbf{C}_i$ and $\mathbf{C}_j$ of the spheres and pointing towards the vertex $\mathbf{P}_{ijk}$ (Fig. 5). For any other sphere, S_l, that intersects S_i and S_j, a vector $\mathbf{u}_l$ can be defined in a similar manner. The surface vertex adjacent to $\mathbf{P}_{ijk}$ along the edge defined by S_i and S_j can be found by computing the angle between $\mathbf{u}_k$ and $\mathbf{u}_l$, measured in the positive (right-handed) direction, for all spheres S_l that intersect both S_i and S_j, and choosing the sphere for which this angle is smallest. Expressions for the coordinate vectors $\mathbf{P}_{ijk}$, in terms of the coordinate vectors $\mathbf{C}_i$, $\mathbf{C}_j$, $\mathbf{C}_k$ of the spheres S_i, S_j, S_k and their radii, were given by Connolly;[12] in the present work, these expressions have been simplified and modified, using an approach based on barycentric coordinates, to improve their computational efficiency.

Computational Cost

The procedure for finding P_1, the point having the smallest y coordinate, requires on the order of n operations, where n is the number of atoms. To find each new vertex, the neighboring vertices must first be listed; this requires a number of tests proportional to k_i, using the cubes defined at the beginning of step 1. If a list of all vertices found on each S_i is maintained, the cost of checking whether a vertex is new is independent of the total number of vertices and of $\langle k \rangle$. Thus, the total cost of the search for all of the vertices is on the order of $m\langle k \rangle$, where m is the number of convex faces and $\langle k \rangle$ is the average number of neighbor spheres that must be searched to find the adjacent vertices.

Since the surface edges were determined by MSEED during the search for surface vertices, all of the convex faces bounded by the edges were determined at that stage of the search procedure. These faces, therefore, constitute the accessible surface (if free spheres and two-sphere intersections are ignored). By adding up all of the operations required for the initialization, it can be seen that the search for the seed vertex and the search for all of the vertices scale as $m\langle k \rangle + n$. For compact molecules such as globular proteins, m^3 is approximately proportional to n^2, since m is proportional to the area and n is proportional to the volume; therefore, SEED scales as $n^{2/3}\langle k \rangle + n$. The exponent in the first term is less than one, and the second term is linear, thus making this an unusually efficient search procedure. For comparison, it may be noted that the determination of a full pairwise interatomic potential scales as n^2. The sublinearity with respect to n and linearity with respect to $\langle k \rangle$ are key features of this algorithm, which are independent of the number of spatial dimensions.

Step 2

Connolly[12] deduced a simple formula for computing the area of each face, by applying the Gauss-Bonnet formula from differential geometry[25] and using the

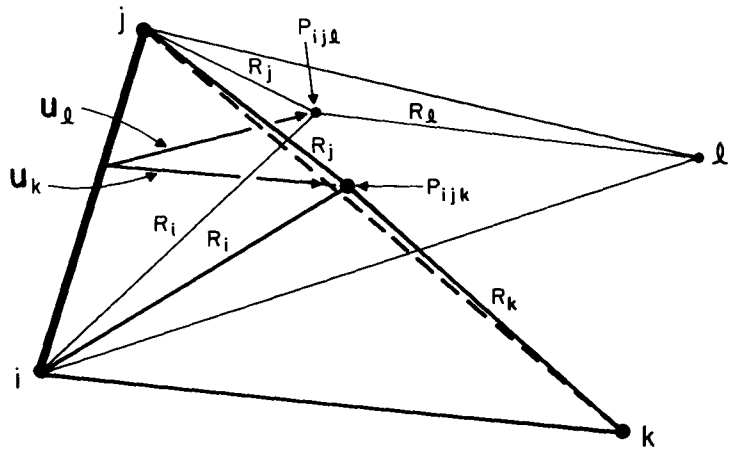

Figure 5. The vertices, $\mathbf{P}_{ijk}$ and $\mathbf{P}_{ijl}$, formed by the two sets of triplets of spheres (S_i, S_j, S_k) and (S_i, S_j, S_l) are shown. The vectors $\mathbf{u}_k$ and $\mathbf{u}_l$ are perpendicular to the line between atoms i and j and are pointed towards $\mathbf{P}_{ijk}$ and $\mathbf{P}_{ijl}$, respectively.

facts that (1) the Gaussian curvature of a sphere is constant and (2) the geodesic curvature of a circle on the surface of a sphere is constant. The final form of Connolly's formula for the area of a convex face is

$$A = r_i^2 \left[2\pi + \sum_e \phi_e \cos \theta_{ei} - \sum_v \beta_v \right] \quad (4)$$

where r_i is the radius of the sphere S_i on which the face lies. In eq. (4), the first sum is over all edges that make up the boundary of the face and the second sum is over the vertices at which these edges intersect; ϕ_e is the arc length of edge e, and θ_{ei} is the polar angle of edge e on sphere S_i (measured from the line joining the center of S_i to the center of the sphere on the other side of the edge); β_v is the exterior angle at vertex v, defined by the two edges of the face that meet at this vertex.

Efficient minimization algorithms require the first derivative of the function to be minimized. The derivative of the area of each face, with respect to the Cartesian coordinates of the centers of the spheres that define the edges and vertices of that face, can be computed without increasing the degree of complexity of the computation, if certain intermediate quantities are computed and stored first. Each of the quantities ϕ_e, θ_{ei} and β_v in eq. (4) depends on the coordinates of the centers of four, two, or three spheres, respectively. Let S_1, S_2, S_3, and S_4 be four partially exposed spheres, whose positions determine the two vertices, $\mathbf{P}_1$ and $\mathbf{P}_2$ and the edge E (Fig. 6). Denoting the coordinate vectors of the spheres by $\mathbf{C}_1$ to $\mathbf{C}_4$ and their radii (extended by the probe radius) by R_1 to R_4, the quantities in eq. (4) depend on the following unit vectors:

$$\mathbf{z} = (\mathbf{C}_2 - \mathbf{C}_1)/|\mathbf{C}_2 - \mathbf{C}_1| \quad (5)$$

$$\mathbf{u}_i^{(1)} = (\mathbf{C}_i - \mathbf{P}_1)/R_i \qquad (i = 1, 2, 3) \quad (6)$$

$$\mathbf{u}_i^{(2)} = (\mathbf{C}_i - \mathbf{P}_2)/R_i \qquad (i = 1, 2, 4) \quad (7)$$

$$\mathbf{v}_k^{(1)} = (\mathbf{u}_i^{(1)} \times \mathbf{u}_j^{(1)})/|\mathbf{u}_i^{(1)} \times \mathbf{u}_j^{(1)}|$$
$$(i, j, k \text{ a cyclic permutation of } 1, 2, 3) \quad (8)$$

$$\mathbf{v}_k^{(2)} = (\mathbf{u}_i^{(2)} \times \mathbf{u}_j^{(2)})/|\mathbf{u}_i^{(2)} \times \mathbf{u}_j^{(2)}|$$
$$(i, j, k \text{ a cyclic permutation of } 1, 2, 4) \quad (9)$$

Then

$$\cos \beta_k^{(1)} = -\mathbf{v}_i^{(1)} \cdot \mathbf{v}_j^{(1)}$$
$$(i, j, k \text{ a cyclic permutation of } 1, 2, 3) \quad (10)$$

$$\cos \beta_k^{(2)} = -\mathbf{v}_i^{(2)} \cdot \mathbf{v}_j^{(2)}$$
$$(i, j, k \text{ a cyclic permutation of } 1, 2, 4) \quad (11)$$

$$\cos \theta_{e1} = -\mathbf{u}_1^{(1)} \cdot \mathbf{z} \quad (12)$$

$$\cos \theta_{e2} = -\mathbf{u}_1^{(2)} \cdot \mathbf{z} \quad (13)$$

$$\cos \phi_e = \mathbf{v}_3^{(1)} \cdot \mathbf{v}_4^{(2)} \quad (14)$$

Derivatives of the quantities in eqs. (5) to (9) are built up in steps and stored as intermediate quantities; an example of such an intermediate quantity is the array of derivatives of the unit vector $\mathbf{z}$, with components z^1, z^2, z^3, with respect to the coordinates of spheres S_1 and S_2, the components of which are

$$\frac{\partial z^l}{\partial C_1^k} = -\frac{\partial z^l}{\partial C_2^k} = \frac{(C_2^l - C_1^l)(C_2^k - C_1^k)}{|\mathbf{C}_2 - \mathbf{C}_1|^3} - \frac{\delta^{lk}}{|\mathbf{C}_2 - \mathbf{C}_1|} \quad (15)$$

where δ is the Kronecker delta, which is zero unless $l = k$. Each of the angles in eqs. (10) to (14) depends on two of the quantities in eqs. (5) to (9); hence, its derivatives can be found immediately from the derivatives of the quantities in eqs. (5) to (9). Finally, the derivatives of the area of one face are sums of contributions from expressions of the type given in eqs. (10) to (14); the number of terms in this sum depends on the number of vertices surrounding the face and is independent of the *total* number of vertices or the average number of neighbors. Evaluation of the derivatives of the expressions in eqs. (5) to (9) can be achieved in a manner that scales linearly

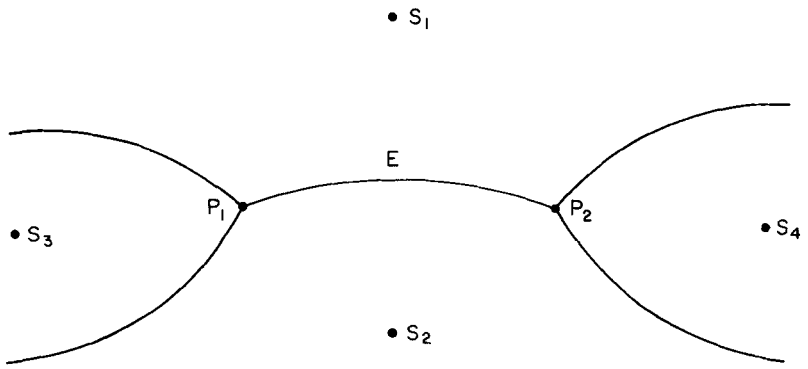

Figure 6. S_1 through S_4 are the centers of spheres 1 through 4. Vertex P_1 is a triple intersection point of spheres S_1, S_2, and S_3; vertex P_2 is a triple intersection point of spheres S_1, S_2, and S_4. Edge E is an arc of the circle of intersection of spheres S_1 and S_2.

Table I. Comparison of CPU times for various programs used to calculate the accessible surface area of different molecules. The probe radius, r_p, was set to zero for all three programs MSEED, ANA, and DOT.

Molecule	Number of residues	Number of united atoms	C.P.U. time (s) DOT (10 dots/Å^2)	MSEED[a] (vec)[b]	MSEED[a] (novec)[c]	ANA (vec)
Met-enkephalin[d]	5	48	0.8	0.06	0.08	0.17
Crambin[e]	46	327	3.01	0.36	0.51	8.27
BPTI[f]	58	454	4.12	0.50	0.71	12.13
Phosphatidyl-2-acylhydrolase[g]	123	957	8.67	0.97	1.41	34.51
Bovine pancreatic ribonuclease A[h]	124	951	8.95	0.93	1.98	44.69
Thermolysin[i]	316	2432	21.96	2.02	2.95	145.33

[a]MSEED computes both the accessible surface area and the derivatives of the area with respect to the Cartesian coordinates.
[b]MSEED (vec): MSEED run with the vector option.
[c]MSEED (novec): MSEED run without the vector option.
[d]Coordinates of the global minimum found by Li and Scheraga.[26]
[e]Coordinates of entry 1CRN in the Brookhaven Protein Data Bank.[27]
[f]Coordinates of entry 5PTI in the Brookhaven Protein Data Bank.[27]
[g]Coordinates of entry 1BP2 in the Brookhaven Protein Data Bank.[27]
[h]Coordinates of an ECEPP/2 minimized structure from Palmer and Scheraga.[28]
[i]Coordinates of entry 3TLN in the Brookhaven Protein Data Bank.[27]

with the number of faces; hence, the overall determination of the Cartesian derivatives scales in the same way. Once the derivatives with respect to Cartesian coordinates are known, derivatives with respect to any other type of variable, such as dihedral angles, can be found using established formulas.

RESULTS AND DISCUSSION

Tests of the Algorithm

MSEED is written in V.S. 2.4 I.B.M. FORTRAN and has been optimized for vector processing. CPU times for sample runs of MSEED on one processor of an I.B.M. 3090/600J are given in Table I. Table I also compares the performance of MSEED with that of two other programs that compute accessible surface area, DOT,[10,11] and ANA,[8] a program based on Connolly's analytical algorithm.[12] The numerical algorithm DOT was run using a density of 10 dots per Å^2. With all three programs, MSEED, ANA, and DOT, and for all of the runs described in this work, the probe radius, r_p, was set to zero and the van der Waals radius was extended by 1.4 Å, which ensured that the accessible surface area was computed by all three programs. The van der Waals radii and definitions of united atoms were those described by Ooi and co-workers.[7]

In Table I, all of the CPU times for the MSEED program are for the calculation of the accessible surface area, as well as the Cartesian derivatives of the accessible surface area. The Cartesian derivatives of the accessible surface areas were not calculated by ANA or DOT.

Table II. Comparison of accessible surface areas for one conformation of each molecule obtained from MSEED, ANA, and DOT.

Molecule	Area (Å^2) MSEED	ANA	DOT
Met-enkephalin[a]	898	898	897
BPTI[b]	4000	4014	4020

[a]The accessible surface area was calculated for the structure obtained by Li and Scheraga,[26] with the probe radius, r_p, set to zero.
[b]The accessible surface was calculated for the ECEPP-native analog of BPTI-5,[29] with the probe radius, r_p, set to zero.

Accuracy of MSEED

Accessible surface areas were calculated for 5000 conformations of the pentapeptide enkephalin and for at least 100 conformations of the 58-residue peptide bovine pancreatic trypsin inhibitor (BPTI).

Enkephalin

The areas computed by MSEED and DOT for one particular conformation of enkephalin were very similar (Table II). Statistical data for accessible surface areas for 5000 conformations of enkephalin, generated randomly, are given in Table III. The largest difference in computed areas found by MSEED and DOT for the same conformation was 14.82 Å^2, i.e., approximately 2% of the average total surface area of enkephalin. When MSEED and ANA were used to compute the accessible surface areas of an additional 5000 randomly generated conformations of enkephalin, the largest difference in surface area for the same conformation was 3.08 Å^2 (Table III), i.e., less than 0.5% of the average total surface area. Table III shows the average difference between the

Table III. Statistical quantities related to the surface areas calculated by MSEED, DOT and ANA for 5000 conformations of Met-enkephalin.

	Difference in surface areas (Å^2)	
	MSEED and DOT	MSEED and ANA
Maximum difference[a]	14.82	3.08
Average difference[b]	1.24 ± 1.00	0.02 ± 0.11

[a]The maximum difference is the largest difference in surface area computed for the same conformation of Met-enkephalin by MSEED and DOT, or by MSEED and ANA, with the probe radius, r_p, set to zero.

[b]The average difference is the average difference in surface area computed for the same conformation of Met-enkephalin by MSEED and DOT, or by MSEED and ANA, with the probe radius, r_p, set to zero.

surface areas determined by MSEED and DOT, and MSEED and ANA, as well as the standard deviation from that average. The table shows that the results from MSEED and ANA differ only slightly; thus, the approximations in the MSEED algorithm do not affect the final results significantly.

BPTI

The areas computed by MSEED and DOT, or MSEED and ANA, for any given conformation of BPTI, were not identical, although the discrepancies were small (Table II). The reasons for these small discrepancies are explained in the section entitled "Special Properties of MSEED."

CPU Comparisons between Surface Determination Programs

The analytical program ANA has not been optimized for vectorization. However, this does not change the analysis of how the program scales as the number of atoms increases. It is clear from the comparisons of CPU times between the different programs (Table I) that MSEED requires much less time than either DOT (with a dot density of 10 dots per Å^2) or ANA. MSEED computes accessible surface areas, and derivatives of these surface areas, about two (for enkephalin) to 70 (for thermolysin) times faster than ANA and eight (for crambin) to 11 (for thermolysin) times faster than DOT, in the tests reported here. In these tests, derivatives of the accessible surface area were not computed by either ANA or DOT. These timing comparisons support the contention that MSEED scales much better as the number of atoms increases than do ANA or DOT. The underlying reason for the increased speed of MSEED is that it scales as $n^{2/3}\langle k\rangle + n$, as mentioned above, whereas ANA scales as $n\langle k\rangle^3$ and DOT scales as $n\langle k\rangle^{5/3}$.

There is another numerical surface determination algorithm,[30] which we shall refer to as the "lattice cell algorithm." The approach used, although numerical, is similar in spirit to the approach used by MSEED. The lattice cell algorithm defines the accessible surface, ignoring all buried atoms, before the area is computed; therefore, the lattice cell algorithm and MSEED should scale similarly. Since a comparison between the lattice cell algorithm and the DOT program has already been presented,[30] a direct comparison between MSEED and the lattice cell algorithm will not be included here. However, it is appropriate to point out that the scaling factor for the lattice cell algorithm suffers from the drawback of all numerically based surface-determination algorithms, namely, that it depends on the desired level of accuracy; as the desired accuracy increases, such algorithms rapidly become slower. Additionally, derivatives must be computed numerically and this necessitates great accuracy in the computation of the areas. For these reasons, the lattice cell algorithm is unlikely to be as suitable as MSEED for energy minimization using algorithms that require derivatives.

CPU Comparison between MSEED and ECEPP

The CPU time required by the ECEPP/2 program.[16,17] for one energy function evaluation plus one gradient evaluation for BPTI was 3.5 seconds, while the calculation of the surface area and its derivatives for this molecule using MSEED required about 0.6 CPU seconds. As noted under Methods, the CPU time required for an evaluation of ECEPP/2 scales as n^2, while the time required for computing solvation energy using MSEED scales with a factor between $n^{2/3}$ and n. Thus, the calculation of solvation energy using the MSEED algorithm will certainly not be rate-limiting in applications to energy minimization of proteins of any size.

Monte Carlo Minimization of Enkephalin

Three series of energy minimization, using the Monte Carlo-minimization (MCM) procedure of Li and Scheraga,[26] were carried out using Met-enkephalin as a model. In one series, the energy function was the ECEPP/2 potential, with no additional term for hydration. In the second series, the ECEPP/2 potential was augmented by an empirical term for the solvation based on surface area, in which the JRF set of parameters of Vila et al.[8] was employed and accessible areas were computed using the program ANA. The third series employed exactly the same energy function as the second series, but accessible surface area was computed using MSEED. The starting structure for the minimization was the global energy minimum located by Li and Scheraga[26] in the absence of solvent. For each run, 150 starting conformations (Monte Carlo moves) were generated. These runs were too short to locate the global min-

ima in the second and third MCM series; also, because MSEED and ANA do not generate identical surfaces, the trajectories of these two MCM runs were somewhat different. The CPU times required for these three series of minimizations were 3.3 minutes for the first series of 150 conformations (ECEPP/2 without solvation), 42 minutes for the second series of 150 conformations (ECEPP/2 plus solvation, using ANA) and 8 minutes for the third series of 150 conformations (ECEPP/2 plus solvation, using MSEED). Thus, even for a molecule as small as enkephalin, for which the scaling properties of MSEED are not especially advantageous, the cost of a conformational search using a potential function that allows for the effect of solvation is scarcely more than double the cost of a search that ignores solvation, with the new algorithm. By contrast, calculating solvation energy with ANA, a program based on Connolly's full analytical algorithm, increased the CPU time by at least an order of magnitude.

Special Properties of MSEED

As pointed out previously, the SEED algorithm in MSEED does not locate edges having no vertices or spheres that have no intersections (free spheres). Therefore, MSEED cannot be used to calculate areas for surfaces in cases where edges without vertices or free spheres are common, i.e., van der Waals surfaces. This problem could in principle be corrected by adding a search for edges without vertices, but for calculation of solvation free energy based on accessible surface area, the small increase in accuracy that would be obtained by including such a search scarcely justifies the additional CPU time. Test runs with enkephalin and BPTI (Table II) indicate that the current version of MSEED is adequate for calculation of solvation free energy of peptides and protein molecules, given the uncertainty inherent in current estimates of solvation free energy per unit area.

MSEED calculates the area of one connected piece of the accessible surface. This means that buried surfaces will not be found. In computation of the potential energy of a protein, cavities that do not contain buried water molecules contribute nothing to the solvation energy, and the solvent-accessible surface is irrelevant. When buried water molecules are found, empirical hydration models based upon exposed surface areas are unlikely to be appropriate. A better approach is to include these water molecules explicitly as part of the interior molecular structure.

Buried surfaces do occur frequently in globular proteins.[15,31] As an example, BPTI contains three accessible surfaces that are not connected to each other; these surfaces have 287, 14, and, 2, vertices, respectively. Because only the exterior surface (with 287 vertices) is determined by MSEED, the total surface area computed by MSEED, is 0.4% smaller than the total area determined by ANA (Table II). Enkephalin has only a single piece of accessible surface, and the surface areas computed by all three programs for this molecule are essentially identical.

Numerical Stability

Errors can occur during the exploration of the surface by MSEED, if there is insufficient accuracy in the test that determines the identity of the next vertex along an edge. In general, exploration along an edge will locate a few widely separated vertices along that edge, and there is no doubt which of these vertices lies closest to the current vertex. However, when the surfaces of more than three atoms meet at a single point (as in a phenyl ring), there exists, in principle, a vertex at which more than three edges meet. MSEED takes advantage of the finite arithmetic of computer calculations to separate this ideal "multiple" vertex into several "simple" vertices (four in the case of the phenyl ring), at each of which the surfaces of only three atoms meet. The edges joining the simple vertices will necessarily be very small (in a typical computation involving a phenyl ring, their arc length can be 10^{-5} radians or less, depending on the accuracy to which the coordinates of the centers of the atoms are calculated); consequently, the dihedral angle measured between any pair of the vertices will also be small, and could easily be smaller than the machine precision. In such a case, there is a strong possibility that, during the search for the next vertex along one of these small edges, the wrong vertex will be chosen; once this occurs, the error is propagated during subsequent searches along edges and the surface is defined incorrectly. This type of error is rare, as long as the computations are performed using double precision arithmetic. When it does occur, a change of ± 0.01 Å in the position of one or more atomic centers will allow the surface to be redefined correctly, with a negligible change in area.

Continuity of the Gradient

Another type of problem can occur when MSEED is used to compute solvation energy during minimization of the total energy. This problem occurs because of changes in the number of vertices and edges that may accompany a change of conformation. When a new vertex or edge appears on the surface, the area of each atomic surface changes continuously but the gradient has a discontinuity. If this discontinuity is large enough, it can cause the minimization algorithm to fail to find a conformation with lower energy because the local search, which uses an interpolative method to locate a new test point, does not function correctly. The same problem

would probably occur with any minimization algorithm that requires a gradient, since such algorithms almost always require that the gradient be continuous. In practice, we have found that this problem does occur very occasionally, but that it is only significant when the conformation is close to a local minimum (so that all contributions to the gradient are small). It is possible to restart the computation from a slightly altered conformation and continue to convergence if desired.

CONCLUSIONS

It is well established that solvation plays a major role in determining conformations of peptides and proteins. For applications involving conformational searches for macromolecules, models based on solvent-accessible surfaces or volumes are probably the most economical methods for including the effect of solvent in the potential function.[1-8,32,33]

For large molecules, molecular dynamics simulations or energy minimizations that employ a potential that does not include solvent are computationally rather expensive. The use of empirical surface area- and volume-based solvent models represents an attempt to add solvation effects to such potentials, without incurring the very large extra demand for CPU time that an explicit treatment of water entails. However, with previous programs for determining surface area, the computational expense of carrying out a conformational search with a potential that included one of these empirical solvation potentials was invariably found to be excessive. The great advantage of the MSEED algorithm is that it determines the accessible surface area in less time than is required for the determination of a pairwise interatomic potential. With the ready availability of the new algorithm, we may now proceed with extensive testing and improvement of empirical hydration potentials based on surface area.

This work was supported at Cornell University by research grants from the National Institute of General Medical Sciences of the National Institutes of Health (GM-14312) and from the National Science Foundation (DMB84-01811). J.V. was supported by a fellowship from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) of Argentina (1988–1990). Support was also received from the Groupement Scientifique I.B.M.-C.N.R.S. The computations were carried out at the Cornell National Supercomputer Facility, a resource of the Cornell Center for Theory and Simulation in Science and Engineering, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute. The program is available from the Quantum Chemistry Program Exchange as QCPE 610.

References

1. W.G. Richards, P.M. King, and C.A. Reynolds, *Protein Engineering*, **2,** 319 (1989).
2. Y.K. Kang, G. Némethy, and H.A. Scheraga, *J. Phys. Chem.*, **91,** 4105 (1987).
3. Y.K. Kang, G. Némethy, and H.A. Scheraga, *J. Phys. Chem.*, **91,** 4109 (1987).
4. Y.K. Kang, G. Némethy, and H.A. Scheraga, *J. Phys. Chem.*, **91,** 4118, (1987).
5. Y.K. Kang, K.D. Gibson, G. Némethy, and H.A. Scheraga, *J. Phys. Chem.*, **92,** 4739 (1988).
6. D. Eisenberg, and A.D. McLachlan, *Nature*, **319,** 199 (1986).
7. T. Ooi, M. Oobatake, G. Némethy, and H.A. Scheraga, *Proc. Natl. Acad. Sci.*, **84,** 3086 (1987).
8. J. Vila, R.L. Williams, M. Vásquez, and H.A. Scheraga, *Proteins: Structure, Function, and Genetics*, **10,** 199 (1991).
9. G. Perrot and B. Maigret, *J. Mol. Graph.*, **8,** 141 (1990).
10. M. Connolly, *Quantum Chem. Prog. Exchange Bull.*, **1,** 75 (1981).
11. M.L. Connolly, *Science*, **221,** 709 (1983).
12. M.L. Connolly, *J. Appl. Cryst.*, **16,** 548 (1983).
13. M.L. Connolly, *J. Appl. Cryst.*, **18,** 499 (1985).
14. F.M. Richards, *Ann. Rev. Biophys. Bioeng.*, **6,** 151 (1977).
15. B. Lee, and F.M. Richards, *J. Mol. Biol.*, **55,** 379 (1971).
16. G. Némethy, M.S. Pottle, and H.A. Scheraga, *J. Phys. Chem.*, **87,** 1883 (1983).
17. M.J. Sippl, G. Némethy, and H.A. Scheraga, *J. Phys. Chem.*, **88,** 6231 (1984).
18. K.A. Palmer and H.A. Scheraga, *J. Comp. Chem.* **12,** 505 (1991).
19. W.C. Still, A. Tempczyk, R.C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc.*, **112,** 6127 (1990).
20. C. Zheng, C.F. Wong, and J.A. McCammon, *Biopolymers*, **29,** 1877 (1990).
21. S.J. Wodak, and J. Janin, *Proc. Natl. Acad. Sci.*, **77,** 1736 (1980).
22. W. Hasel, T.F. Hendrickson, and W.C. Still, *Tetrahedron Comput. Methodol.*, **1,** 103 (1988).
23. G. Némethy, Z.I. Hodes, and H.A. Scheraga, *Proc. Natl. Acad. Sci.*, **75,** 5760 (1978).
24. T.J. Richmond, *J. Mol. Biol.*, **178,** 63 (1984).
25. M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-hall, Englewood Cliffs, New Jersey, 1990, pp. 67, 264–283.
26. Z. Li and H.A. Scheraga, *Proc. Natl. Acad. Sci.*, **84,** 6611 (1987).
27. F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer, M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).
28. K.A. Palmer and H.A. Scheraga, *J. Comp. Chem.* (in press).
29. D.R. Ripoll, L. Piela, M. Vásquez, and H.A. Scheraga, *Proteins: Structure, Function, and Genetics*, **10,** 188 (1991).
30. H.R. Karfunkel and V. Eyraud, *J. Comp. Chem.*, **10,** 628 (1989).
31. J.S. Richardson, *Adv. Prot. Chem.*, **34,** 167 (1981).
32. M.K. Gilson and B. Honig, *Proteins: Structure, Function, and Genetics*, **4,** 7 (1988).
33. M. Saito and H. Nakamura, *J. Comp. Chem.*, **11,** 76 (1990).