Exact and Efficient Analytical Calculation of the Accessible Surface Areas and Their Gradients for Macromolecules

ROBERT FRACZKIEWICZ, WERNER BRAUN

Sealy Center for Structural Biology, University of Texas Medical Branch, Galveston, Texas 77555-1157

Received 27 May 1996; accepted 2 September 1997

ABSTRACT: A new method for exact analytical calculation of the accessible surface areas and their gradients with respect to atomic coordinates is described. The new surface routine, GETAREA, finds solvent-exposed vertices of intersecting atoms, and thereby avoids calculating buried vertices which are not needed to determine the accessible surface area by the Gauss–Bonnet theorem. The surface routine was implemented in FANTOM, a program for energy minimization and Monte Carlo simulation, and tested for accuracy and efficiency in extensive energy minimizations of Met-enkephalin, the α-amylase inhibitor tendamistat, and avian pancreatic polypeptide (APP). The CPU time for the exact calculation of the accessible surface areas and their gradients has been reduced by factors of 2.2 (Met-enkephalin) and 3.2 (tendamistat) compared with our previous approach. The efficiency of our exact method is similar to the recently described approximate methods MSEED and SASAD. The performance of several atomic solvation parameter sets was tested in searches for low energy conformations of APP among conformations near the native X-ray crystal structure and highly distorted structures. The protein solvation parameters from Ooi et al. [*Proc. Natl. Acad. Sci. USA*, **84,** 3086 (1987)] and from Wesson and Eisenberg [*Prot. Sci.*, **1,** 227 (1992)] showed a good correlation between solvation energies of the conformations and their root-mean-square deviations from the

Correspondence to: W. Braun; E-mail: werner@nmr.utmb.edu

Contract/grant sponsor: National Science Foundation; contract/grant number: DBI-9632326

Contract/grant sponsor: Department of Energy; contract/grant number: DE-FG03-96ER62267

CCC 0192-8651 / 98 / 030319-15

X-ray crystal structure of APP. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 319–333, 1998

Keywords: solvation energy; solvent accessible surface area; atomic solvation parameters; Monte Carlo simulation; FANTOM; avian pancreatic polypeptide

Introduction

The accessible surface area (ASA) of proteins[1] is central for computing the effect of protein solvation. The free energy of protein solvation[2,3] is linearly related to the ASA in a continuum approach,[4,5] and this energy term has been used with some success to recognize native or native-like folds of proteins and to distinguish these folds from nonnative compact folds.[6–8] The first methods for analytical surface area calculation, introduced by Connolly[9] and Richmond,[10] have been improved in recent years,[11–17] as reviewed in Braun.[18] Computational efficiency has been increased primarily by using a probabilistic approximation in the calculation of the surface area.[19–21] Other methods relied on fast search methods to find exposed vertices in an approximate way,[22] or proposed numerical approximations in the calculation of the gradient.[23] Calculation of free energies in solution is currently limited to small polypeptides[24,25] due to the vast amount of computer time needed. Solvation terms have been included for proteins in energy minimizations,[12,26,27] Monte Carlo simulations,[28] molecular dynamics calculations,[29–31] and protein–protein docking studies.[32] To be useful for energy minimization, Monte Carlo simulations, or molecular dynamics calculations, the accessible surface areas and their derivatives with respect to the atomic coordinates must be analytically calculated.

In this article, we describe a new method for calculating both the surface area and the gradient exactly and efficiently. We use the intersection of half-spaces (IHS), defined by the planes of two-sphere intersection, to find all solvent exposed vertices of intersecting atoms in an efficient way. Geometric inversion transforms the intersection planes into a set of points in the dual space. The convex hull of these points corresponds to a dual space image of the desired IHS. The vertices of a Gauss–Bonnet path[9,10,33] can be quickly found by intersecting edges of the IHS with the central atom sphere. Our new method avoids the calculation of a large number of buried vertices that are not needed for calculation of the accessible surface area by the Gauss–Bonnet theorem.[28,33] This approach was implemented as a new routine, GETAREA, in our energy minimization and Monte Carlo simulation package, FANTOM.[34] We show here that the new surface routine, which is a factor of two to three times faster than our previous routine,[28] is almost as efficient as the approximate methods implemented in MSEED[22] and SASAD.[23] Including a solvation energy term in the energy minimization by the new version of FANTOM[35] adds computational cost of about the same magnitude as used for energy minimization *in vacuo*.

Three protein-solvation models were tested for their capability to characterize native or near-native structures as local minima with low energy values by extensive energy minimizations of the avian pancreatic polypeptide (APP). We minimized the conformational energy including the protein solvation term of 268 APP conformations which had low *in vacuo* energy values and differed from the native structure in the range of 1–6 Å root-mean-square deviation for backbone atoms. The conformation with lowest energy had the correct topology in all three solvation parameter sets. The conformation with the lowest total energy value was considerably improved for the empirically derived parameters from Ooi et al.[5] and from Wesson and Eisenberg parameters[29] as compared with the conformation with lowest *in vacuo* energy.

Methods

PARAMETERIZATION OF MOLECULAR ACCESSIBLE SURFACE AREA

As proposed by Eisenberg and McLachlan,[4] the free energy of protein–solvent interaction can be approximately derived from the solvent-accessible surface areas (ASA) A_i of atoms i by the following linear relation:

$$E_{hyd} = \sum_{i \in \text{atoms}} \sigma_i A_i \qquad (1)$$

where σ_i is an empirical solvation parameter depending on the atom type. Atoms are treated as spheres with an *ASA radius* equal to the corresponding atomic van der Waals radius plus 1.4 Å, as defined by Richards.[1b] Only atoms at a distance less than the sum of corresponding ASA radii will mutually influence their accessible surface areas.

The atomic ASA can be exactly calculated from the global Gauss–Bonnet theorem[9, 10, 33]:

$$A_i = r_i^2\left[2\pi(2 - \chi_i) + \sum_{\lambda=1}^{p} (\Omega^i_{\lambda,\,\lambda+1} + \Phi^i_\lambda \cos\Theta^i_\lambda)\right] \tag{2}$$

where p is the number of intersecting arcs λ defining ASA (the ''Gauss–Bonnet path''), r_i is the ASA radius, χ_i stands for the Euler–Poincaré characteristic[33] of a given ASA region, $\Omega^i_{\lambda,\,\lambda+1}$ is the angle between vectors tangential to the accessible arcs; Φ^i_λ is the arc length in angular units, and Θ^i_λ denotes the polar angle of the intersection circle (see Fig. 1).

We present a modified vector parameterization for the intersection points of accessible arcs that, unlike our previous approach,[12, 28] treats all neighbor atoms equivalently. This simplifies calculation of the gradient. In the first step we calculate the intersection points from Cartesian coordinates of the central and neighbor atoms ($\mathbf{x}_k$) and their radii (r_k). A number of auxiliary vector and scalar quantities are defined below[36] and illustrated in Figure 2:

$$\mathbf{x}^i_k = \mathbf{x}_k - \mathbf{x}_i \tag{3}$$

$$d^i_k = |\mathbf{x}^i_k| \tag{4}$$

$$\boldsymbol{\mu}^i_k = \mathbf{x}^i_k / d^i_k \tag{5}$$

$$g^i_k = \frac{(d^i_k)^2 + r_i^2 - r_k^2}{2d^i_k} \tag{6}$$

$$a^i_k = \sqrt{r_i^2 - (g^i_k)^2} \tag{7}$$

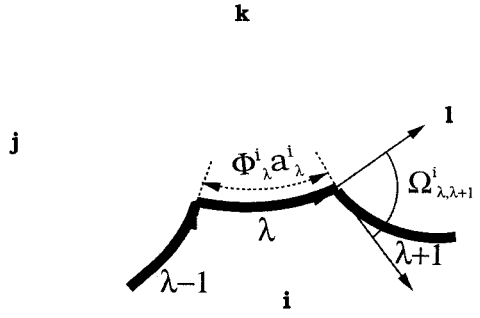

FIGURE 1. Part of the Gauss–Bonnet path enclosing the solvent-accessible surface of central atom *i*. The path is composed of a certain number of accessible arcs, λ, which are parts of the intersection circles with other atoms. $\Omega^i_{\lambda,\,\lambda+1}$ is the angle between vectors tangential to the two consecutive arcs, Φ^i_λ is the angle defining arc length, and a^i_λ denotes the radius of the intersection circle.

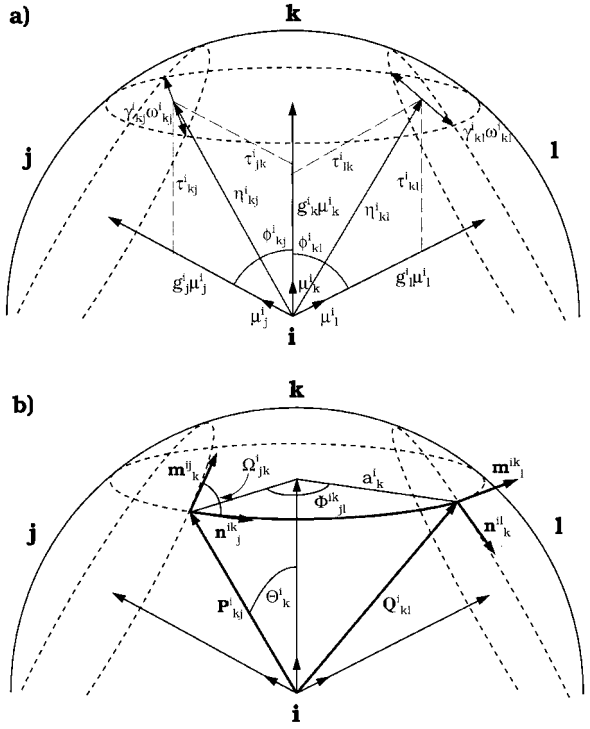

FIGURE 2. Vector parameterization of the Gauss–Bonnet path. An accessible arc is part of the intersection circle generated by central atom *i* and a neighbor atom *k*. Two other neighbor atoms, *j* and *l*, define the arc's vertices: $\mathbf{P}^i_{kj}$ and $\mathbf{Q}^i_{kl}$. These points are parameterized as vector sums: $\mathbf{P}^i_{kj} = \boldsymbol{\eta}^i_{kj} + \gamma^i_{kj}\boldsymbol{\omega}_{ikj}$ and $\mathbf{Q}^i_{kl} = \boldsymbol{\eta}^i_{kl} - \gamma^j_{kl}\boldsymbol{\omega}_{ikl}$, where $\boldsymbol{\eta}^i_{k*}$ is the midpoint of a segment generated by two crossing circles of intersection, and $\gamma^i_{k*}\boldsymbol{\omega}_{ik*}$ points to one of the segment's ends. See text for detailed description of other parameters.

$$\cos \phi_{kj}^i = \boldsymbol{\mu}_k^i \circ \boldsymbol{\mu}_j^i \tag{8}$$

$$\boldsymbol{\eta}_{kj}^i = \tau_{kj}^i \boldsymbol{\mu}_k^i + \tau_{jk}^i \boldsymbol{\mu}_j^i \tag{9}$$

$$\tau_{kj}^i = \frac{g_k^i - g_j^i \cos \phi_{kj}^i}{\sin^2 \phi_{kj}^i} \tag{10}$$

$$\boldsymbol{\omega}_{ikj} = \frac{\boldsymbol{\mu}_k^i \times \boldsymbol{\mu}_j^i}{\sin \phi_{kj}^i} \tag{11}$$

$$\gamma_{kj}^i = \sqrt{r_i^2 - g_k^i \tau_{kj}^i - g_j^i \tau_{jk}^i} \tag{12}$$

A unit vector $\boldsymbol{\mu}_k^i$ points from the central atom i toward the kth neighbor atom; g_k^i and a_k^i are the polar distance to the center of the circle of intersection (COI) with the kth atom and its radius, respectively; $\boldsymbol{\eta}_{kj}^i$ is the midpoint of the segment defined by the intersection of the kth and jth COIs; $2\gamma_{kj}^i$ is the total length of this segment. A unit vector, $\boldsymbol{\omega}_{ikj}$, is perpendicular to both $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\mu}_j^i$ and parallel to the COI intersection segment. To ensure equivalency of neighbor atoms a nonorthogonal basis set ($\boldsymbol{\mu}_k^i$, $\boldsymbol{\mu}_j^i$, and $\boldsymbol{\omega}_{ikj}$) was used.

The implementation of the global Gauss–Bonnet theorem used by Richmond[10] originally calculated the *buried surface area* of atom i which, subtracted from $4\pi r_i^2$, yields the value of ASA. For this reason, the right-handed orientation of the Gauss–Bonnet path defining the buried surface becomes left-handed with respect to the accessible surface (Fig. 1). This orientation allows us to unambiguously define the *intersection points* or *vertices*, $\mathbf{P}_{kj}^i$ and $\mathbf{Q}_{kl}^i$, of an accessible arc belonging to the kth COI intersected by spheres j and l (Fig. 2):

$$\begin{aligned} \mathbf{P}_{kj}^i &= \boldsymbol{\eta}_{kj}^i + \gamma_{kj}^i \boldsymbol{\omega}_{ikj} \\ \mathbf{Q}_{kl}^i &= \boldsymbol{\eta}_{kl}^i - \gamma_{kl}^i \boldsymbol{\omega}_{ikl} \end{aligned} \tag{13}$$

The next step yields unit vectors tangential to the accessible arc at its intersection points:

$$\begin{aligned} \mathbf{n}_j^{ik} &= \frac{\boldsymbol{\mu}_k^i \times \mathbf{P}_{kj}^i}{a_k^i} \\ \mathbf{m}_l^{ik} &= \frac{\boldsymbol{\mu}_k^i \times \mathbf{Q}_{kl}^i}{a_k^i} \end{aligned} \tag{14}$$

All quantities in eq. (2) can now be calculated from these tangential vectors:

$$\Omega_{jk}^i = -\arccos\left(\mathbf{n}_j^{ik} \circ \mathbf{m}_k^{ij}\right) \tag{15}$$

There is a minus sign in eq. (15), because, for intersecting spheres, all tangential angles are negatively oriented.[33] The angular arc length, Φ_{jl}^{ik}, can be calculated from the arcus cosine of the scalar product of two tangential vectors, $\mathbf{n}_j^{ik} \circ \mathbf{m}_l^{ik}$. This gives either the angular length of the accessible arc, or that of the complementary arc.[28] In the second case, the value of the arcus cosine must be subtracted from 2π. Both cases are handled by one equation [eq. (16)], where S_{jl}^{ik} is the sign of relative orientation of the axial vector, $\boldsymbol{\mu}_k^i$, and the tangential vectors:

$$\Phi_{jl}^{ik} = \left(1 - S_{jl}^{ik}\right)\pi + S_{jl}^{ik} \arccos\left(\mathbf{n}_j^{ik} \circ \mathbf{m}_l^{ik}\right) \tag{16}$$

$$S_{jl}^{ik} = \operatorname{sign}\left(\boldsymbol{\mu}_k^i \circ \left(\mathbf{n}_j^{ik} \times \mathbf{m}_l^{ik}\right)\right) \tag{17}$$

Finally, the polar angle of the kth COI is simply:

$$\cos \Theta_k^i = g_k^i / r_i \tag{18}$$

GRADIENT OF MOLECULAR ACCESSIBLE SURFACE AREA

We calculate the gradient of the accessible surface area in a multilevel computational scheme. All quantities calculated at an nth level depend on the results of levels 1 through $n - 1$. This scheme is highly suitable for machine processing (either scalar, or parallel) and is efficient in CPU time and memory demand. Detailed derivation of the gradient is presented below[37]; all indices correspond to Figure 2:

Level 1:

$$\frac{\partial g_k^i}{\partial \mathbf{x}_k^i} = \frac{d_k^i - g_k^i}{d_k^i} \boldsymbol{\mu}_k^i \tag{19}$$

$$\frac{\partial \boldsymbol{\mu}_k^i}{\partial \mathbf{x}_k^i} = \frac{1}{d_k^i} (\mathbf{I} - \boldsymbol{\mu}_k^i \otimes \boldsymbol{\mu}_k^i) \tag{20}$$

Level 2:

$$\frac{\partial \cos \phi_{kj}^i}{\partial \mathbf{x}_k^i} = \frac{1}{d_k^i} \left(\boldsymbol{\mu}_j^i - \cos \phi_{kj}^i \boldsymbol{\mu}_k^i\right) \tag{21}$$

$$\frac{\partial \cos \Theta_k^i}{\partial \mathbf{x}_k^i} = \frac{1}{r_i} \left(\frac{\partial g_k^i}{\partial \mathbf{x}_k^i}\right) \tag{22}$$

Level 3:

$$\frac{\partial \tau_{kj}^i}{\partial \mathbf{x}_k^i} = \frac{1}{\sin^2 \phi_{kj}^i} \times \left[\left(\frac{\partial g_k^i}{\partial \mathbf{x}_k^i} \right) + \left(g_j^i - 2\tau_{jk}^i \right) \left(\frac{\partial \cos \phi_{kj}^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (23)$$

$$\frac{\partial \tau_{jk}^i}{\partial \mathbf{x}_k^i} = \frac{1}{\sin^2 \phi_{kj}^i} \left[-\cos \phi_{kj}^i \left(\frac{\partial g_k^i}{\partial \mathbf{x}_k^i} \right) + \left(g_k^i - 2\tau_{kj}^i \right) \left(\frac{\partial \cos \phi_{kj}^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (24)$$

$$\frac{\partial \boldsymbol{\omega}_{ikj}}{\partial \mathbf{x}_k^i} = \frac{\cos \phi_{kj}^i}{\sin^2 \phi_{kj}^i} \boldsymbol{\omega}_{ikj} \otimes \left(\frac{\partial \cos \phi_{kj}^i}{\partial \mathbf{x}_k^i} \right) - \frac{1}{\sin \phi_{kj}^i} \left[\boldsymbol{\mu}_j^i \times \left(\frac{\partial \boldsymbol{\mu}_k^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (25)$$

Level 4:

$$\frac{\partial \gamma_{kj}^i}{\partial \mathbf{x}_k^i} = \frac{\tau_{kj}^i}{\gamma_{kj}^i d_k^i} \left[\tau_{jk}^i \boldsymbol{\mu}_j^i - \left(d_k^i - \tau_{kj}^i \right) \boldsymbol{\mu}_k^i \right] \quad (26)$$

$$\frac{\partial \boldsymbol{\eta}_{kj}^i}{\partial \mathbf{x}_k^i} = \boldsymbol{\mu}_k^i \otimes \left(\frac{\partial \tau_{kj}^i}{\partial \mathbf{x}_k^i} \right) + \boldsymbol{\mu}_j^i \otimes \left(\frac{\partial \tau_{jk}^i}{\partial \mathbf{x}_k^i} \right) + \tau_{kj}^i \left(\frac{\partial \boldsymbol{\mu}_k^i}{\partial \mathbf{x}_k^i} \right) \quad (27)$$

Level 5:

$$\frac{\partial \mathbf{P}_{kj}^i}{\partial \mathbf{x}_k^i} = \left(\frac{\partial \boldsymbol{\eta}_{kj}^i}{\partial \mathbf{x}_k^i} \right) + \boldsymbol{\omega}_{ikj} \otimes \left(\frac{\partial \gamma_{kj}^i}{\partial \mathbf{x}_k^i} \right) + \gamma_{kj}^i \left(\frac{\partial \boldsymbol{\omega}_{ikj}}{\partial \mathbf{x}_k^i} \right) \quad (28)$$

$$\frac{\partial \mathbf{Q}_{kl}^i}{\partial \mathbf{x}_k^i} = \left(\frac{\partial \boldsymbol{\eta}_{kl}^i}{\partial \mathbf{x}_k^i} \right) - \boldsymbol{\omega}_{ikl} \otimes \left(\frac{\partial \gamma_{kl}^i}{\partial \mathbf{x}_k^i} \right) - \gamma_{kl}^i \left(\frac{\partial \boldsymbol{\omega}_{ikl}}{\partial \mathbf{x}_k^i} \right) \quad (29)$$

Level 6:

$$\frac{\partial \mathbf{n}_j^{ik}}{\partial \mathbf{x}_k^i} = \frac{1}{a_k^i} \left[\frac{g_k^i}{a_k^i} \mathbf{n}_j^{ik} \otimes \left(\frac{\partial g_k^i}{\partial \mathbf{x}_k^i} \right) + \boldsymbol{\mu}_k^i \times \left(\frac{\partial \mathbf{P}_{kj}^i}{\partial \mathbf{x}_k^i} \right) - \mathbf{P}_{kj}^i \times \left(\frac{\partial \boldsymbol{\mu}_k^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (30)$$

$$\frac{\partial \mathbf{m}_l^{ik}}{\partial \mathbf{x}_k^i} = \frac{1}{a_k^i} \left[\frac{g_k^i}{a_k^i} \mathbf{m}_l^{ik} \otimes \left(\frac{\partial g_k^i}{\partial \mathbf{x}_k^i} \right) + \boldsymbol{\mu}_k^i \times \left(\frac{\partial \mathbf{Q}_{kl}^i}{\partial \mathbf{x}_k^i} \right) - \mathbf{Q}_{kl}^i \times \left(\frac{\partial \boldsymbol{\mu}_k^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (31)$$

$$\frac{\partial \mathbf{m}_k^{ij}}{\partial \mathbf{x}_k^i} = \frac{1}{a_j^i} \left[\boldsymbol{\mu}_j^i \times \left(\frac{\partial \mathbf{P}_{kj}^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (32)$$

$$\frac{\partial \mathbf{n}_k^{il}}{\partial \mathbf{x}_k^i} = \frac{1}{a_l^i} \left[\boldsymbol{\mu}_l^i \times \left(\frac{\partial \mathbf{Q}_{kl}^i}{\partial \mathbf{x}_k^i} \right) \right] \quad (33)$$

Level 7:

$$\frac{\partial \Omega_{jk}^i}{\partial \mathbf{x}_k^i} = \frac{1}{|\sin \Omega_{jk}^i|} \left[\mathbf{m}_k^{ij} \circ \left(\frac{\partial \mathbf{n}_j^{ik}}{\partial \mathbf{x}_k^i} \right) + \mathbf{n}_j^{ik} \circ \left(\frac{\partial \mathbf{m}_k^{ij}}{\partial \mathbf{x}_k^i} \right) \right] \quad (34)$$

Level 8:

$$\frac{\partial \Phi_{jl}^{ik}}{\partial \mathbf{x}_k^i} = -\frac{S_{jl}^{ik}}{|\sin \Phi_{jl}^{ik}|} \left[\mathbf{m}_l^{ik} \circ \left(\frac{\partial \mathbf{n}_j^{ik}}{\partial \mathbf{x}_k^i} \right) + \mathbf{n}_j^{ik} \circ \left(\frac{\partial \mathbf{m}_l^{ik}}{\partial \mathbf{x}_k^i} \right) \right] \quad (35)$$

$$\frac{\partial \Phi_{j-1,k}^{ij}}{\partial \mathbf{x}_k^i} = -\frac{S_{j-1,k}^{ij}}{|\sin \Phi_{j-1,k}^{ij}|} \left[\mathbf{n}_{j-1}^{ij} \circ \left(\frac{\partial \mathbf{m}_k^{ij}}{\partial \mathbf{x}_k^i} \right) \right] \quad (36)$$

$$\frac{\partial \Phi_{k,l+1}^{il}}{\partial \mathbf{x}_k^i} = -\frac{S_{k,l+1}^{il}}{|\sin \Phi_{k,l+1}^{il}|} \left[\mathbf{m}_{l+1}^{il} \circ \left(\frac{\partial \mathbf{n}_k^{il}}{\partial \mathbf{x}_k^i} \right) \right] \quad (37)$$

$$\frac{\partial A_i}{\partial \mathbf{x}_k} = r_i^2 \sum_{j,k,l \in K} \left[\left(\frac{\partial \Omega_{jk}^i}{\partial \mathbf{x}_k^i} \right) + \left(\frac{\partial \Omega_{kl}^i}{\partial \mathbf{x}_k^i} \right) + \Phi_{jl}^{ik} \left(\frac{\partial \cos \Theta_k^i}{\partial \mathbf{x}_k^i} \right) + \cos \Theta_k^i \left(\frac{\partial \Phi_{jl}^{ik}}{\partial \mathbf{x}_k^i} \right) + \cos \Theta_j^i \left(\frac{\partial \Phi_{j-1,k}^{ij}}{\partial \mathbf{x}_k^i} \right) + \cos \Theta_l^i \left(\frac{\partial \Phi_{k,l+1}^{il}}{\partial \mathbf{x}_k^i} \right) \right] \quad (38)$$

$$\frac{\partial A_i}{\partial \mathbf{x}_i} = -\sum_{j,k,l \in L} \left(\frac{\partial A_i}{\partial \mathbf{x}_k} \right) \quad (39)$$

Any remaining equations not included in this derivation can be obtained by index exchange and from the following symmetry properties: $\phi_{kj}^i = \phi_{jk}^i$; $\boldsymbol{\eta}_{kj}^i = \boldsymbol{\eta}_{jk}^i$; $\boldsymbol{\omega}_{ikj} = -\boldsymbol{\omega}_{ijk}$; $\gamma_{kj}^i = \gamma_{jk}^i$; $\mathbf{P}_{kj}^i = \mathbf{Q}_{jk}^i$; $\mathbf{n}_j^{ik} = -\mathbf{m}_j^{ki}$; $\Omega_{kj}^i = \Omega_{jk}^i$; $S_{jl}^{ik} = S_{lj}^{ki}$; $\Phi_{jl}^{ik} = \Phi_{lj}^{ki}$.

DETERMINATION OF GAUSS–BONNET PATH

Two methods are implemented in GETAREA to further reduce the CPU time for calculating the Gauss–Bonnet vertices. The first, the cubic lattice, is a standard method in distance geometry[38] and

molecular dynamics calculations.[39] Potential neighbor atoms are found by searching the nearest environment of each atom, not the entire molecule. The 3D space is divided by a cubic grid, the unit length of which is twice the maximal atomic radius. The search for neighbor atoms is restricted to the atom's own cell and the 26 surrounding cells.

We introduce here the second method: a new algorithm of intersecting half-spaces (IHS) for surface area and gradient calculations. Figure 3 illustrates some basic ideas of intersecting half-spaces and their relation to accessible arcs in two dimensions. A central atom sphere, S, intersects two neighboring atoms, K_1 and K_2. The surface of the central sphere, δS, is completely or partially covered by that of its neighbors. Each neighbor sphere generates half-spaces, H_1 and H_2 respectively, which are bounded by a plane containing the circle of intersection and contain those parts of the surface, δS, which are not buried by K_1 and K_2. The accessible area δS_{acc} of the surface of the central atom is give by:

$$\delta S_{\text{acc}} = \delta S \cap H_1 \cap H_2 \tag{40}$$

In the general case of N spheres, the accessible surface δS_{acc} of S can be found by the intersection of N half-spaces:

$$\delta S_{\text{acc}} = \delta S \cap \bigcap_{i=1}^{N} H_i \tag{41}$$

where IHS $\equiv \bigcap_{i=1}^{N} H_i$, the intersection of half-spaces, can be a convex polyhedron (polygon in 2D), or an unbounded convex polyhedral cone (Fig. 3). In any case, parts of δS contained within the IHS are accessible to the solvent. Faces of the IHS correspond to the neighbor atoms, but only those faces that intersect δS actually contribute to the formation of the Gauss—Bonnet path. If none of the faces contact δS, then the central atom is ''inside'' the protein and totally inaccessible to the solvent. Therefore, the IHS defines the topology of the corresponding central atom in the protein molecule.

There are standard methods to obtain an intersection of N arbitrary half-spaces in 3D[40] by computational geometry (Fig. 4). First, the distance vectors to every half-space boundary from the atom center are determined (Fig. 4a). These vectors are subsequently transformed through geometric inversion, which maps a vector with spherical polar coordinates (R, θ, ϕ) to the dual-space vector with coordinates $(\frac{1}{R}, \theta, \phi)$ (Fig. 4b). A ''half-space'' with the boundary plane placed at infinity must be included to represent cases where IHS is a polyhedral cone. This half-space transforms directly into the center of geometric inversion. In practical terms, finding the convex hull is equivalent to eliminating irrelevant neighbor atoms. In the next step, a convex hull of the inverted points (plus the center) is calculated by adapting an incremental algorithm.[41] Half-spaces that do not contribute to the boundary of IHS correspond to points in the dual space that are *internal* with respect to the convex hull (Fig. 4c). Consequently, only the ''boundary half-spaces'' are transformed into vertices of the convex hull. It can be proven[40] that the geometric inversion of distance vectors to the *faces* of the convex hull shown in Figure 4c leads to the vertices of the IHS shown in Figure 4d. The Gauss–Bonnet path can now be determined directly, as the edges of IHS cross the surface of S at exactly the vertices of accessible arcs $\mathbf{P}_{kj}^{i}$ and $\mathbf{Q}_{kl}^{i}$. However, due to rounding errors, it is better to recalculate these intersection points from eq. (13).

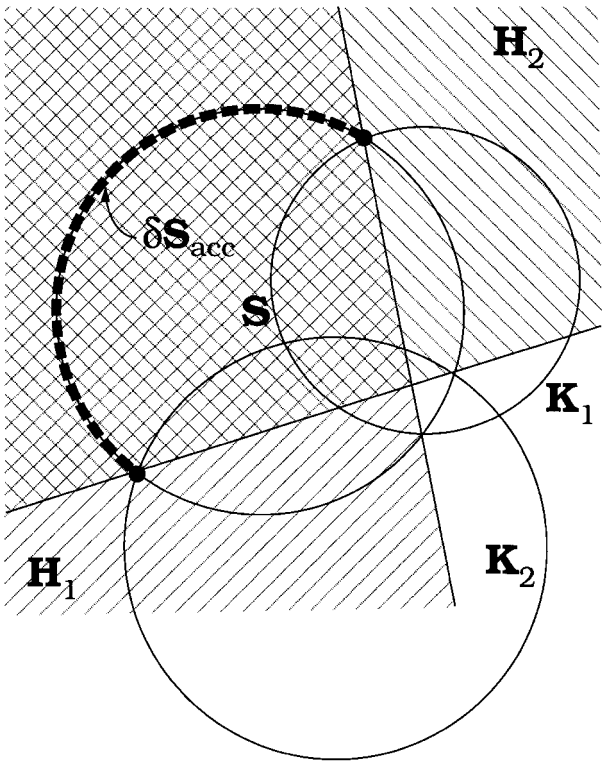

FIGURE 3. Two half-spaces (H_1 and H_2, shaded areas) defined by two spheres (K_1 and K_2) intersecting a central atom S. Intersection of $H_1 \cap H_2$ and the central sphere surface, δS, defines accessible surface δS_{acc} (thick dashed line).

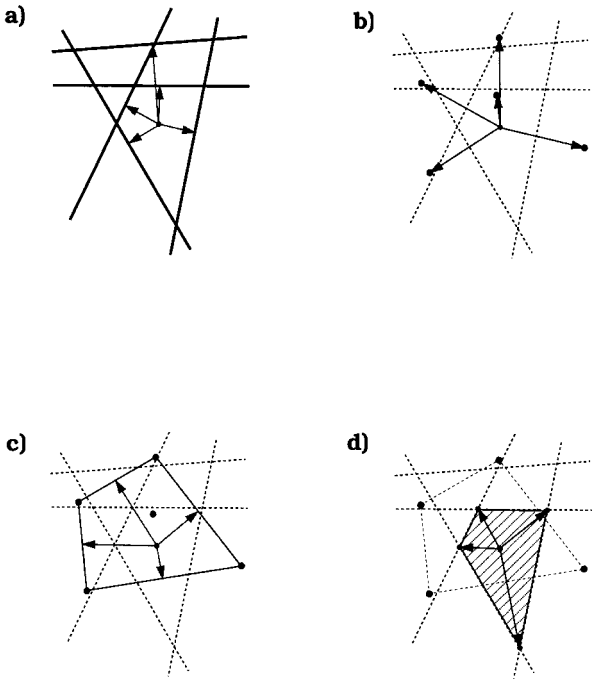

FIGURE 4. Step-by-step procedure for finding the IHS polyhedron. (a) Find distance vectors to every half-space boundary. (b) Transform distance vectors by geometric inversion. (c) Find convex hull of transformed points; find distance vectors to the faces of the convex hull. (d) Transform convex hull distance vectors by geometric inversion; the transformed points are vertices of the IHS polyhedron.

Eqs. (3)–(39) are included in the new surface routine GETAREA. GETAREA has been integrated into the program FANTOM[34] for studies on protein solvation using energy minimization and Monte Carlo simulation.

STRUCTURE GENERATION AND ENERGY MINIMIZATIONS OF AVIAN PANCREATIC POLYPEPTIDE

Atomic Solvation Parameters

As in our previous work[12,27,28] we test a simple one-parameter set, APOLAR, which calculates the protein solvent interaction as 0.025 kcal/mol · Å^2 for the nonpolar accessible surface area, as initially estimated by Chothia,[42] and two classical solvation parameter sets, OONS[5] and WWE,[29] which were statistically derived by a many-parameter fit of transfer free energies. All solvation parameters and atomic radii are summarized in Table I.

Initial Structure for MC Simulations

The crystal structure of the avian pancreatic polypeptide (APP) hormone from turkey *Meleagris gallopavo*[43] (Brookhaven Protein Data Bank[44] file 1PPT) was regularized to standard bond lengths and valence angles by the program DIAMOD,[45] using 20,000 distance constraints between 301

TABLE I.
Atomic Solvation Parameters and van der Waals Radii.

	Atomic solvation parameters (kcal / mol · Å^2)			Radius (Å)	
Atom type	OONS[a]	WWE[b]	APOLAR[c]	S & R[d]	Ooi[e]
Aliphatic C	0.008	0.012	0.025	2.00	2.00
Carbonyl or carboxyl C	0.427	0.012	0.025	1.50	1.55
Aromatic C	−0.008	0.012	0.025	1.85	1.75
Amide N	−0.132	−0.116	0.000	1.50	1.55
Amine N	−0.132	−0.186	0.000	1.50	1.55
Carbonyl or carboxyl O	−0.038	−0.116	0.000	1.40	1.40
Carbonyl or carboxyl O$^-$	−0.038	−0.175	0.000	1.40	1.40
Hydroxyl O	−0.172	−0.116	0.000	1.40	1.40
Thiol S	−0.021	−0.018	0.000	1.85	2.00
Sulfur S	−0.021	−0.018	0.025	1.85	2.00

[a] Solvation parameters from Table 1 of ref. 5.
[b] Solvation parameters from Table 3 of ref. 29.
[c] Solvation parameters according to Chothia.[42]
[d] The van der Waals radii from Table 2 of ref. 48 used for WWE and APOLAR sets.
[e] The van der Waals radii from Table 1 of ref. 5 used for OONS parameter set.

heavy atoms. The regularized structure used in our study as initial structure for the Monte Carlo simulation deviated from the X-ray crystal coordinates only by 0.26 Å and 0.48 Å RMSD for backbone and heavy atoms, respectively.

Generation of Local Minima Structures and Energy Refinement

Starting from the regularized structure, an ensemble of APP conformations was generated by the modified Monte Carlo method of Li and Scheraga[46] with an adaptive temperature schedule.[47] A total of six Monte Carlo runs (each of 1000 steps with 70 unconstrained ECEPP/2 energy minimizations per step) were performed using the Metropolis criterion, each with different temperature schedule to enhance conformational space sampling. The initial and minimal temperatures, and temperature increment were equal to: 300 K, 5 K, 600 K (run 1); 100 K, 5 K, 200 K (run 2); 500 K, 5 K, 1000 K (run 3); 1000 K, 300 K, 1000 K (run 4); 2000 K, 1000 K, 1000 K (run 5); and 15,000 K, 10,000 K, 5000 K (run 6). Energy was minimized by the Newton–Raphson algorithm with 8-Å cutoff for the nonbonding pair list, which was updated every 10 minimization steps. This minimizer reduced the length of gradient below 0.01% of its initial value in 99% of structures. The parameters for the minimization, σ, ρ, and τ, were set to 0.5, 0.3, and 0.1, respectively.[34] The Lennard–Jones potential was smoothed for nonbonding distances smaller than 2 Å to avoid numeric overflow. The dielectric constant was proportional to interatomic distances. The backbone angles ϕ and ψ were allowed to change within a range of $\pm 10°$, the side-chain angles were contained within a $\pm 90°$ range. Only one randomly chosen angle was changed per Monte Carlo step. The numbers of conformations found in each Monte Carlo run were 43, 18, 72, 66, 169, and 637, respectively. The lowest energy conformations were found in run 3. The resulting 1005 conformations were combined into a single file. The RMS deviations from the native structure of APP were characterized by a nonuniform distribution in two large clusters ranging from 1 Å to 3.5 Å and 4.6 Å to 6.2 Å, respectively. The ensemble was reduced to 268 distinct conformations by eliminating configurations that were similar or identical (maximum angular backbone and side-chain variance were less than 8° and 10°, respectively, and their energy difference was less than 300 kcal/mol). This ensemble roughly corresponds to a subset of distinct local minima on the ECEPP/2 energy hypersurface. It preserves the overall energy–RMSD distribution of the original 1005 structures and is characterized by positive correlation between backbone and heavy atoms RMSD.

The 268 conformations were subsequently minimized by a conjugate gradient algorithm with each of the ''ECEPP/2 + solvent'' functions.[12,28] The minimization iterations were stopped if the length of the gradient vector reached 1% of its initial value ($\sim$ 90% of structures) or until the conjugate gradient minimizer could not reduce the energy value after 10,000 steps. In the latter case, the relative final value of gradient was usually below 10%. All other minimization parameters were equal to those used in the minimization of ECEPP/2 energy function.

Results

TESTING CORRECTNESS AND PERFORMANCE OF GETAREA

The analytical gradient was tested for Met-enkephalin (5-residue peptide) and tendamistat (74-residue protein) against the numerically estimated first derivatives of solvation energy:

$$\frac{\Delta E_{hyd}}{\Delta \psi_i} \equiv \frac{E_{hyd}(\psi_i + \Delta\psi_i) - E_{hyd}(\psi_i - \Delta\psi_i)}{2\Delta\psi_i} \quad (42)$$

where ψ_i stands for an ith dihedral angle. The average relative differences were calculated as the RMS error:

$$D(E_{hyd}, \Delta\psi) \equiv \frac{\sqrt{\sum_{i=1}^{N}\left(\frac{\partial E_{hyd}}{\partial \psi_i} - \frac{\Delta E_{hyd}}{\Delta \psi_i}\right)^2}}{\sqrt{\sum_{i=1}^{N}\left(\frac{\Delta E_{hyd}}{\Delta \psi_i}\right)^2}} \quad (43)$$

where N is the number of dihedral angles. The tests were performed with all heavy-atom atomic solvation parameters equal to 1.0 kcal/mol · Å. The good agreement between the analytical and numerical gradients (Table II) demonstrated that our approach and implementation is correct.

As a second test, we compared the results of energy minimizations for the two peptides with the new routine GETAREA in FANTOM 4.0 to the previously used surface routine PARAREA in FANTOM 3.5 using the same solvation parameters

TABLE II.
Comparison of Numerical and Analytical Gradients of Solvation Energies for Met-Enkephalin and Tendamistat.

Molecule	$\Delta\psi_i$[a]	$D(E_{hyd}, \Delta\psi_i)$[b]
Met-enkephalin	0.0001	1.77×10^{-6}
	0.01	7.14×10^{-6}
	1.0	6.84×10^{-2}
Tendamistat	0.0001	5.11×10^{-8}
	0.01	9.62×10^{-5}
	1.0	7.83×10^{-1}

[a] Dihedral angle increments in degrees.
[b] Relative average difference between analytical and numerical gradients as defined in eq. (43).

as before. The results from both methods were identical; the final structures of Met-enkephalin minimized with FANTOM 3.5 and 4.0 (two distinct starting conformations, final gradient values below 0.05 kcal/mol · Å) differed by 0.0015 kcal/mol and 0.0025 kcal/mol, and 0.00 Å RMSD and 0.01 Å RMSD, respectively. After 500 conjugate gradient steps of energy minimization of tendamistat with both versions of FANTOM the resulting structures deviated from each other by 2.1 kcal/mol and 0.07 Å RMSD.

Convergence of energy minimizations in FANTOM 4.0 with the new surface routine to small final gradient values was tested in two energy minimizations of Met-enkephalin for all three-parameter sets in Table I. Starting from two different structures with gradient values of a few hundred kilocalories per mole per angstrom, the gradient values were reduced in all six runs below 0.01 kcal/mol · Å in less than 250 conjugate gradient iterations.

The improvement in CPU time of our new routine was tested with the peptide Met-enkephalin and the protein tendamistat (Table III). The average CPU time per one routine invocation was a factor of 2.2 and 3.2 less in GETAREA compared with PARAREA, where the difference in performance can be attributed to the larger average number of atomic neighbors in tendamistat. The improvement of *overall* FANTOM performance, by a factor of about 1.9, was observed for both molecules with GETAREA. In comparison to energy minimization *in vacuo*, inclusion of solvent in FANTOM 4.0 requires only 1.66 and 0.71 times more CPU time for Met-enkephalin and tendamistat, respectively. Analogous factors for FANTOM 3.5 were equal to 3.51 and 2.29.

We then compared the performance of GETAREA to that of two fast, approximate programs for calculating molecular surface area and its gradient: MSEED[22] and SASAD[23] (Table IV). Accessible surface areas and their gradients were calculated for a set of five proteins (4PTI, 6LYZ, 2PTN, 1RHD, and 1MCP[44]) used previously to demonstrate the performance of SASAD.[23] Atomic radii and solvation parameters were the same for every input protein. Program SASAD was run at two different accuracy levels distinguished by different initial COI point densities: 12 and 24. Four levels of density doubling[23] were used in each case re-

TABLE III.
CPU Times Required for Surface Routines GETAREA and PARAREA.

	Met-enkephalin		Tendamistat	
	PARAREA[a]	GETAREA[a]	PARAREA[a]	GETAREA[a]
Total CPU time (s)[b]	33.4	17.8	823	437
CPU time spent in surface routine (s)[c]	26.0	11.1	573	182
Number of routine invocations	1302	1215	1349	1382
CPU time per invocation (s)	0.020	0.009	0.425	0.132
Ratio (P / G)	2.22		3.22	

[a] All tests performed on a Silicon Graphics Indigo 2 workstation with a 195-MHz MIPS R10000 microprocessor. The SGI MIPSPro F77 compiler was used in each case with the same floating point precision of REAL*8 and the -64 -O2 optimization level.
[b] Measurements were taken during 500 max steps of conjugate gradient (ECEPP/2 + E_{hyd}) minimization.
[c] CPU time was obtained by reference to the system function DTIME (SGI MIPSPro F77).

TABLE IV.
Comparison of GETAREA to SASAD and MSEED in Accuracy and Efficiency.

Test	Method	Molecule[a] 4PTI	6LYZ	2PTN	1RHD	1MCP
Area[b]	SASAD(4,12)	0.21 (1.80)	0.16 (1.58)	0.15 (1.39)	0.16 (1.70)	0.15 (2.37)
	MSEED	0.35 (43.3)	0.17 (43.2)	0.08 (43.1)	0.10 (43.6)	0.04 (43.7)
Gradient[c]	SASAD(4,12)[d]	3.6 (100)	3.4 (152)	5.3 (788)	3.2 (154)	6.1 (7470)
	SASAD(4,24)[e]	1.8 (100)	2.2 (431)	2.5 (287)	1.6 (113)	3.1 (3724)
	MSEED	4.3 (251)	5.2 (693)	6.5 (342)	4.4 (461)	2.7 (247)
CPU time[f]	GETAREA	0.37	0.86	1.49	2.10	3.14
	SASAD(4,12)	0.21	0.47	0.78	1.13	1.68
	SASAD(4,24)	0.27	0.61	1.04	1.47	2.19
	MSEED	0.25	0.48	0.79	1.30	1.80
Number of atoms		454	1001	1629	2325	3401

[a]Brookhaven protein structure code.[44]
[b]Average absolute deviation of surface area per atom in sq. angstroms calculated with eq. (44). Numbers in parentheses represent the largest absolute deviation.
[c]Average relative deviation of Cartesian gradients per atom in percent calculated with eq. (45). Numbers in parentheses represent the largest relative deviation.
[d]SASAD(4,12): 12 initial COI points with four levels of density doubling.
[e]SASAD(4,24): 24 initial COI points with four levels of density doubling.
[f]CPU time in seconds. All tests were performed on a Silicon Graphics Indigo 2 workstation with a 195-MHz MIPS R10000 microprocessor. CPU times used to read input files from disk were excluded from each test. The SGI MIPSPro F77 compiler was used in each case with the same floating point precision of REAL*8 and the -64 -O3 optimization level.

sulting in 192 and 384 final COI points referred to as SASAD(4,12) and SASAD(4,24), respectively. The efficiency of GETAREA is within a factor of two that of the efficiency of the fast approximate methods.

The relative performance depends on compiler options. Figure 5 presents the typical test results for BPTI calculated with all optimization options available in 32-bit and 64-bit compilation modes. Table IV lists CPU times for all proteins at the fastest -64 -O3 level of optimization. In particular, if one assumes that unoptimized runs illustrate the efficiency of the FORTRAN code itself (-O0 in Fig. 5), then the GETAREA algorithm becomes comparable in performance to that of SASAD and MSEED.

The accuracy of surface area calculations was estimated by an average absolute deviation per atom (Table IV):

$$D(A) \equiv \frac{1}{N}\sum_{i=1}^{N} |A_i^G - A_i^a| \qquad (44)$$

where N denotes number of atoms and A_i is the ASA. Superscripts ''G'' and ''a'' denote results obtained with GETAREA and an approximate method, respectively. The accuracy of Cartesian gradients of solvation energy was calculated as an average relative deviation per atom:

$$D(\partial E_{hyd}) \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\frac{\partial E_{hyd}^G}{\partial \mathbf{x}_i} - \frac{\partial E_{hyd}^a}{\partial \mathbf{x}_i}\right|}{\left|\frac{\partial E_{hyd}^G}{\partial \mathbf{x}_i}\right|} \qquad (45)$$

where $\mathbf{x}_i$ is the position of an ith atom. Table IV shows that surface areas calculated by SASAD and MSEED consistently deviate from exact values by 0.04–0.35 Å^2/atom for all test proteins, whereby the source of their errors differs. MSEED[22] systematically ignores surface area of cavities and fully exposed circles of intersection, while taking into account only the outermost molecular surface defined by a single walk of a test probe along partially exposed COIs. The largest deviation of 43 Å^2 is due to neglecting the buried surface areas of the end atoms of arginine side chains. SASAD calculates solvent-accessible surface area numerically, by the Shrake–Rupley method with 482 test points on a template sphere.[48] The results of gradient calculations by SASAD and MSEED consistently deviate from exact values by 3–6% and 2–3%, respectively. Typical gradient error distribution histograms are shown in Figure 6. While the error of SASAD's numerical algorithm has a random

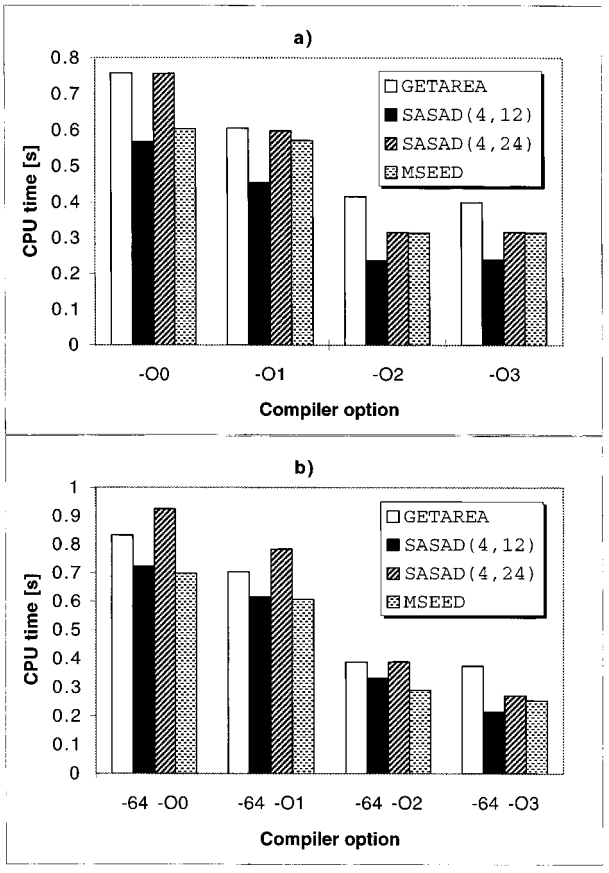

FIGURE 5. Comparison of the performance of surface routines depending on the FORTRAN compiler's level of optimization, increasing from -O0 (no optimization) to -O3 (aggressive optimization). Tests were performed on Silicon Graphics Indigo 2 workstation equipped with a 195-MHz MIPS R10000 microprocessor and MIPSPro FORTRAN 77 compiler, version 7.0. The same floating-point precision of REAL*8 was used in each case. Disk I/O processing times were excluded. Each program calculated surface area and its gradient of BPTI (Brookhaven code 4PTI[44]). Program SASAD was tested at two levels of accuracy: (1) SASAD(4,12), 12 initial COI points; and (2) SASAD(4,24), 24 initial COI points. Four levels of density doubling were used in both cases. (a) Codes compiled in the old 32-bit Application Binary Interface (MIPS2 ABI) mode. (b) Codes compiled in the 64-bit MIPS4 ABI mode.

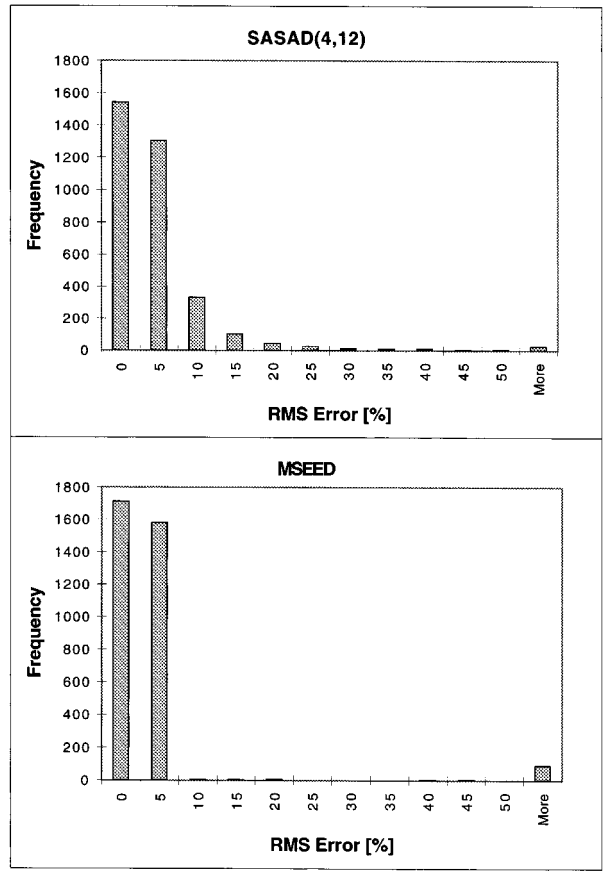

FIGURE 6. Typical histograms of RMS error distribution of the individual atomic surface gradients. First bin counts RMS errors *exactly* equal to zero, each subsequent bin represents an interval of 5%. Data shown here were calculated for 1MCP by SASAD(4,12) (upper graph) and MSEED (lower graph).

nature, the vast majority of MSEED atomic gradients are either exact (first bin) or accurate within 10^{-3}% (second bin) and only few tens of surface atoms have gradients deviating by 100% or more.

In summary, GETAREA is an exact analytical ASA routine with similar CPU time requirements as fast approximate algorithms. It surpasses them, however, when the accuracy and internal consistency of numerical calculations are considered—important factors for practical applications such as molecular energy minimization. In our experience, even very small gradient perturbations can prevent a minimizer from reaching a local energy minimum. Moreover, during minimization, evaluations of both the accessible surface area and its gradient are necessary. Therefore, it is highly imperative to perform consistent area and gradient calculations within *the same* algorithm.

CONFORMATIONAL DEPENDENCE ON ATOMIC SOLVATION PARAMETERS: STUDIES OF AVIAN PANCREATIC POLYPEPTIDE

It was previously shown that including the atomic solvation energy term in energy refinement or folding of proteins can drive perturbed

or unfolded 3D structures toward the native structure.[12, 28, 34, 49] In this study we investigate the distribution of local minima with low energies with and without solvent for the small 36-residue protein, APP.[43] The X-ray crystal structure of APP is compact and stable, although it has no disulfide bridges. An antiparallel polyproline-like helix of residues 2 to 8 is packed against an α-helix formed by residues 14 to 31. Spectroscopic data indicate that hydrophobic interactions between these two segments stabilize the X-ray conformation also in aqueous solution.[43, 50] In the X-ray crystal lattice a zinc ion is coordinated to Gly1, Asn23, and His34 of three different symmetry-related protein molecules.

We have determined the effect of solvation energy terms on shifting local minima of the ECEPP/2 energy function.[51] Energy values *in vacuo* and in solution with all three solvation parameter sets, APOLAR, OONS, and WWE (Table I), were calculated for all 268 local energy minima conformations described earlier, and correlated with the backbone root-mean-square deviation (RMSD) from the crystal structure (Fig. 7). The conformation with the lowest *in vacuo* energy value of −414 kcal/mol deviates by 2.9 Å from the crystal structure. All three solvent energy terms clearly favor the native, indicated by the diamond in the figure, or native-like structures with low energy values. For the OONS and WWE parameter set, the native structure has the lowest solvent energy value and, for the APOLAR parameter set, similar low solvent energies were found for the native and native-like structures deviating by about 2 Å RMSD. The difference in solvation energy between the native structure and the second lowest energy value was most pronounced in the WWE parameter set. Continuum solvation models can, therefore, reasonably differentiate between native and nonnative structures of APP, and show approximately a ''funnel-like'' relationship between the energy and the deviation from the native structure.[52] These results coincide with the observations reported by Vila for BPTI.[49]

The addition of the solvent term to the ECEPP/2 energy term should therefore improve the energy–RMSD relation. However, ECEPP/2 local minima conformations are generally not local minima conformations for the total energy including solvation terms. Therefore, we have minimized the total energy for the three parameter sets. The distribution of the local minima is shown in Figure 8. There are a minimal effect on the distribution of APOLAR local minima conformations as compared with the minima *in vacuo*. The conformation with the lowest total energy deviates about 2.9 Å from the crystal structure, similar to the conformation with the lowest energy *in vacuo*. The distributions significantly improved for the WWE and the OONS parameter sets, as compared with the dis-

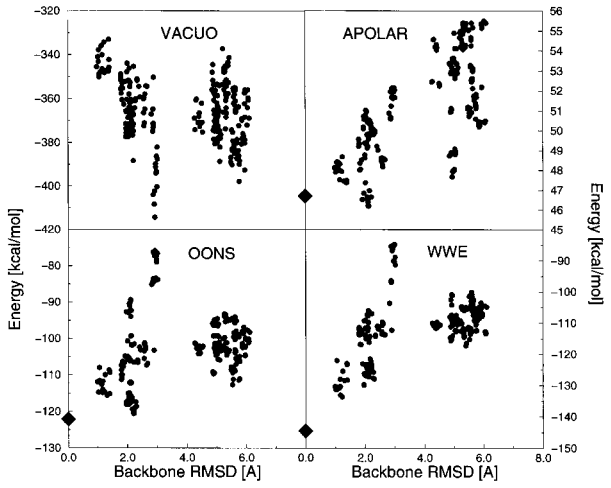

FIGURE 7. Solvation energy vs. backbone RMSD plot calculated using parameters APOLAR, OONS, and WWE from Table I for 268 *in vacuo* structures. The diamond corresponds to the native structure. *In vacuo* energy plot is included for comparison.

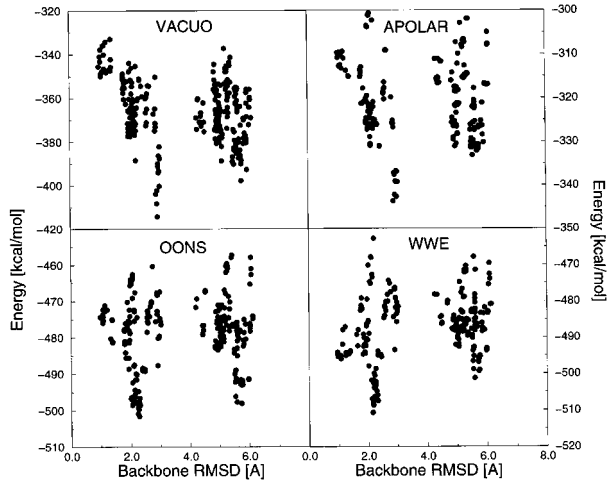

FIGURE 8. Total energy vs. backbone RMSD plots for 268 structures after minimizing ECEPP/2 plus the solvation energy term using parameters from Table I.

tribution *in vacuo*. The conformation with lowest energies shifted within about a 2-Å deviation from the native structure.

The range of APOLAR solvent energy values of about 10 kcal/mol observed in the initial conformations is not sufficient to substantially change the energy surface. Previous work for the proteins BPTI, tendamistat, and the pheromone Er-10[27,28] indicated that increasing the weight of APOLAR solvent energy terms might drive nonnative structures toward the native structure. This procedure applied to APP lead to severe distortions of the α-helix, as we did not apply secondary structure restraints (data not shown).

The four conformations with lowest energies after minimization *in vacuo* and for the three solvent terms, APOLAR, OONS, and WWE, are shown in Figure 9a–d. In all four conformations, the topology of the native structure is reproduced, a remarkable result, as the initial structures deviate from the native structure by up to 6 Å backbone RMSD. The major differences to the X-ray crystal structure occur near the N- and C-termini. Similar results for APP have been obtained from electrostatically driven Monte Carlo simulations (EDMC) with ECEPP/2 potential by Liwo et al.[53] Large deviations from the native structure were also found by molecular dynamics simulations with the OPLS/Amber force field with a continuum solvation term.[54] The relatively large RMSD may result in part from the presence of zinc ion in the crystal structure. The two lowest energy structures of the OONS and the WWE energy minima are remarkably similar to the native structures in the polyproline and the α-helical region. In contrast to the *in vacuo* and the APOLAR low energy conformations, they show an electrostatic attraction between the N- and C-termini.

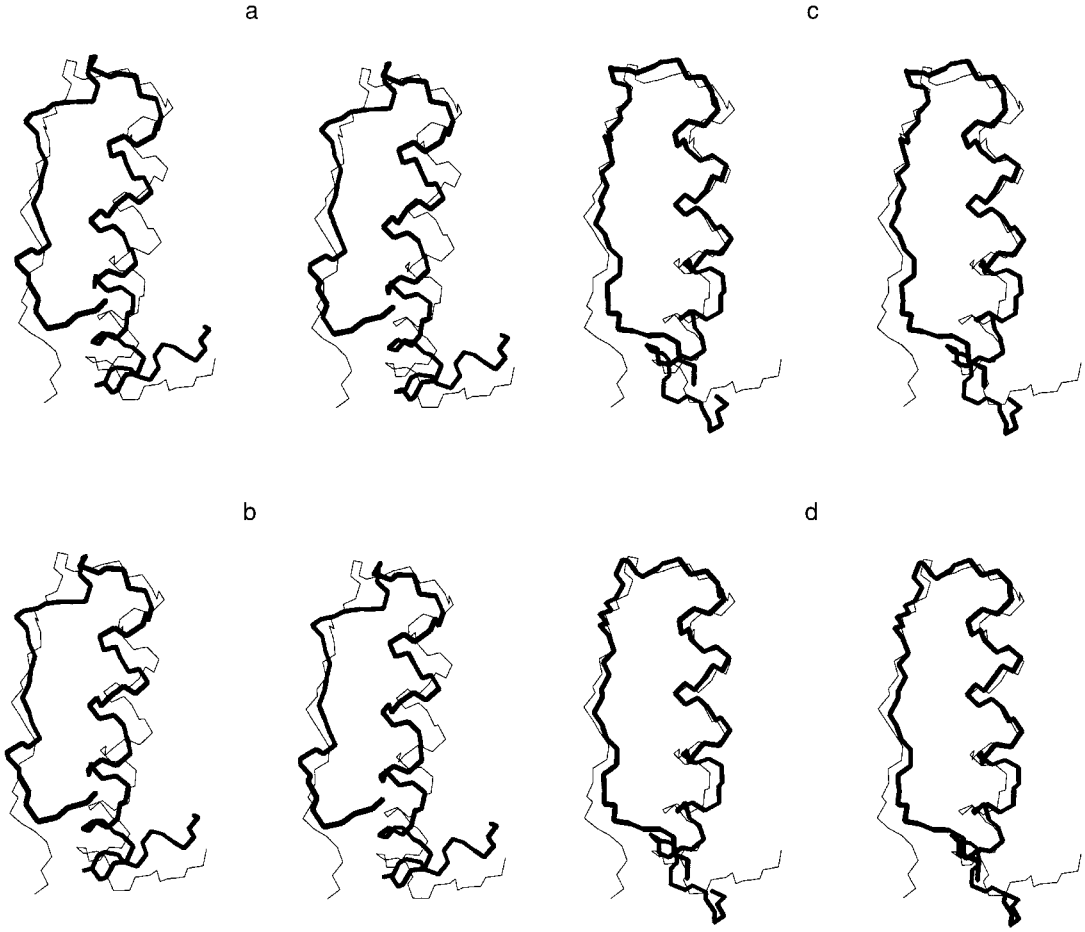

FIGURE 9. Lowest energy structures (thick lines) superposed on the regularized crystal structure of APP (thin lines). Backbone atoms of all residues were used in the superposition. The same orientation of the crystal structure was preserved in each case. The following energy functions were used (see Table I): (a) ECEPP/2; (b) APOLAR; (c) OONS; and (d) WWE. Pictures were generated with the program MOLMOL.[55]

Conclusions

The new version of FANTOM makes energy minimization, including solvation energy terms, almost as fast as energy minimization *in vacuo*. The additional computational cost for the calculation with solvent amounts to 70–160% of the CPU time for the *in vacuo* calculations. Several continuum solvation models were tested for their capability to characterize native or near-native structures as local minima with low energy values, and differentiate them from nonnative folds. In energy minimizations of 268 conformations deviating from the native structure by as much as 6 Å RMSD for backbone atoms, the conformation with the lowest energy had the correct topology in calculations with all three solvation parameter sets. The main differences were found at the protein termini. All three solvation energy terms showed a good energy–RMSD correlation in this ensemble of conformations. The native structure had low solvation energy values in all three cases. It had a significantly lower solvation energy among all sampled conformations; that is, 11 kcal/mol lower than the second lowest value in the Wesson and Eisenberg parameter set.[29] The conformation with lowest total energy value was considerably improved for the empirically derived parameters from Ooi et al.[5] and from Wesson and Eisenberg.[29] We are now extending our study to several other proteins. The new version of FANTOM can be requested from the corresponding author.

Acknowledgments

We thank Dr. C. H. Schein for critical reading of the manuscript.

References

1. (a) B. Lee and F. M. Richards, *J. Mol. Biol.*, **55,** 379 (1971); (b) F. M. Richards, *Annu. Rev. Biophys. Bioeng.*, **6,** 151 (1977).
2. B. Honig and A. Nichols, *Science*, **268,** 1144 (1995).
3. P. E. Smith and B. M. Pettitt, *J. Phys. Chem.*, **98,** 9700 (1994).
4. D. Eisenberg and A. D. McLachlan, *Nature*, **316,** 199 (1986).
5. T. Ooi, M. Oobatake, G. Némethy, and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **84,** 3086 (1987).
6. L. Chiche, L. M. Gregoret, F. E. Cohen, and P. A. Kollman, *Proc. Natl. Acad. Sci. USA*, **87,** 3240 (1990).
7. V. P. Collura, P. J. Greaney, and B. Robson, *Prot. Eng.*, **7,** 221 (1994).
8. Y. Wang, H. Zhang, and R. Scott, *Prot. Sci.*, **4,** 1402 (1995).
9. M. L. Connolly, *J. Appl. Cryst.*, **16,** 548 (1983).
10. T. J. Richmond, *J. Mol. Biol.*, **178,** 63 (1984).
11. E. Silla, F. Villar, O. Nilsson, J. L. Pascual-Ahuir, and O. Tapia, *J. Mol. Graph.*, **8,** 168 (1990).
12. B. von Freyberg and W. Braun, *J. Comput. Chem.*, **14,** 510 (1993).
13. F. Eisenhaber and P. Argos, *J. Comput. Chem.*, **14,** 1272 (1993).
14. M. L. Connolly, *J. Mol. Graph.*, **11,** 139 (1993).
15. M. Petitjean, *J. Comput. Chem.*, **15,** 507 (1994).
16. F. Eisenhaber, P. Lijnzaad, P. Argos, C. Sander, and M. Scharf, *J. Comput. Chem.*, **16,** 273 (1995).
17. M. Cossi, B. Mennucci, and R. Cammi, *J. Comput. Chem.*, **17,** 57 (1996).
18. W. Braun, In *Computer Simulation of Biomolecular Systems, Vol.* 3, W. van Gunsteren, P. Weiner, and T. Wilkinson, Eds., ESCOM, Leiden, 1996.
19. S. J. Wodak and J. Janin, *Proc. Natl. Acad. Sci. USA*, **77,** 1736 (1980).
20. W. Hasel, T. F. Hendrickson, and W. C. Still, *Tetrahed. Comput. Methodol.*, **1,** 103 (1988).
21. W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc.*, **112,** 6127 (1990).
22. G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga, *J. Comput. Chem.*, **13,** 1 (1992).
23. (a) S. Sridharan, A. Nicholls, and K. A. Sharp, *J. Comput. Chem.*, **16,** 1038 (1995); (b) S. Sridharan, A. Nicholls, and B. Honig, *Biophys. J.*, **61,** A174 (1992).
24. E. Meirovitch and H. Meirovitch, *Biopolymers*, **38,** 69 (1996).
25. O. Collet and S. Premilat, *Int. J. Peptide Prot. Res.*, **47,** 239 (1996).
26. R. L. Williams J. Vila, G. Perrot, and H. A. Scheraga, *Prot. Struct. Funct. Genet.*, **14,** 110 (1992).
27. B. von Freyberg, T. J. Richmond, and W. Braun, *J. Mol. Biol.*, **233,** 275 (1993).
28. Ch. Mumenthaler and W. Braun, *J. Mol. Model.*, **1,** 1 (1995).
29. L. Wesson and D. Eisenberg, *Prot. Sci.*, **1,** 227 (1992).
30. C. A. Schiffer, J. W. Caldwell, P. A. Kollman, and R. M. Stroud, *Mol. Sim.*, **10,** 121 (1993).
31. F. Fraternali and W. F. van Gunsteren, *J. Mol. Biol.*, **256,** 939 (1996).
32. M. D. Cummings, T. N. Hart, and R. J. Read, *Prot. Sci.*, **4,** 2087 (1995).
33. M. P. doCarmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
34. T. Schaumann, W. Braun, and K. Wüthrich, *Biopolymers*, **29,** 679 (1990).
35. Information on how to obtain the new version of FANTOM can be found on the web page http://www.scsb.utmb.edu/fantom/fm_home.html or by e-mail request to werner@nmr.utmb.edu.
36. Vectors are printed in bold. "$\circ$" and "$\times$" denote scalar and cross-products of two vectors, respectively. "$|\cdot|$" is a vector norm.

37. ''⊗'' denotes tensor product. The ''scalar product'' and ''cross-product'' of a vector **a** and a matrix **B**, $\mathbf{a} \times \mathbf{B}$ and $\mathbf{a} \circ \mathbf{B}$, result in new matrices whose columns are scalar and cross-products of **a** and corresponding column vectors of **B**, respectively. Symbol $\frac{\partial}{\partial \mathbf{a}}$, where **a** is a column vector, is equivalent to a row vector $(\frac{\partial}{\partial a_1}, \frac{\partial}{\partial a_2}, \frac{\partial}{\partial a_3})$. **I** has the meaning of a 3×3 identity matrix. K denotes set of accessible arcs belonging to the kth COI; L is the set of all accessible arcs.
38. W. Braun and N. Go, *J. Mol. Biol.*, **186,** 611 (1985).
39. G. S. Grest, B. Dünweg, and K. Kremer, *Comp. Phys. Commun.*, **55,** 269 (1989).
40. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
41. J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, New York, 1993.
42. C. Chothia, *Nature*, **248,** 338 (1974).
43. T. L. Blundell, J. E. Pitts, I. J. Tickle, S. P. Wood, and C. W. Wu, *Proc. Natl. Acad. Sci. USA*, **78,** 4175 (1981).
44. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer Jr., M. D. Brice, J. R. Rogers, O. Kennard, T. Schimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).
45. (a) P. Güntert, W. Braun, and K. Wüthrich, *J. Mol. Biol.*, **217,** 517 (1991); (b) G. Hänggi and W. Braun, *FEBS Lett.*, **344,** 147 (1994).
46. Z. Li and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **84,** 6611 (1987).
47. B. von Freyberg and W. Braun, *J. Comput. Chem.*, **12,** 1065 (1991).
48. A. Shrake and J. A. Rupley, *J. Mol. Biol.*, **79,** 351 (1973).
49. J. Vila, R. L. Williams, M. Vásquez, and H. A. Scheraga, *Prot. Struct. Funct. Genet.*, **10,** 199 (1991).
50. (a) P. J. Chang, M. E. Noelken, and J. R. Kimmel, *Biochemistry*, **19,** 1844 (1980); (b) M. E. Noelken, P. J. Chang, and J. R. Kimmel, *Biochemistry*, **19,** 1838 (1980); (c) I. Glover, I. Haneef, J. E. Pitts, S. P. Wood, D. Moss, I. J. Tickle, and T. L. Blundell, *Biopolymers*, **22,** 293 (1983).
51. (a) F. A. Momany, R. F. McGuire, A. W. Burgess, and H. A. Scheraga, *J. Phys. Chem.*, **79,** 2361 (1975); (b) G. Némethy, M. S. Pottle, and H. A. Scheraga, *J. Phys. Chem.*, **87,** 1883 (1983).
52. B. Park and M. Levitt, *J. Mol. Biol.*, **258,** 367 (1996).
53. A. Liwo, M. R. Pincus, R. J. Wawak, S. Rackovsky, and H. A. Scheraga, *Proc. Sci.*, **2,** 1715 (1993).
54. H. Zhang, C. F. Wong, T. Thacher, and H. Rabitz, *Prot. Struct. Funct. Genet.*, **23,** 218 (1995).
55. R. Koradi, M. Billeter, and K. Wüthrich, *J. Mol. Graph.*, **14,** 51 (1996).