An Algebraic Spline Model of Molecular Surfaces

Wenqi Zhao [a], Guoliang Xu [b] and Chandrajit Bajaj [a]

[a] *Institute for Computational Engineering Sciences*
The University of Texas at Austin

[b] *Institute of Computational Mathematics and Scientific/Engineering Computing*
Chinese Academy of Sciences

Abstract

In this paper, we describe a new method to generate a smooth algebraic spline (AS) model approximation of the molecular surface (MS), based on an initial coarse triangulation derived from the atomic coordinate information of the biomolecule, resident in the PDB (Protein data bank). Our method first constructs a triangular prism scaffold P_s covering the PDB structure, and then generates piecewise polynomial Bernstein-Bezier (BB) spline function approximation F within P_s, which are nearly C^1 everywhere. Approximation error and point sampling convergence bounds are also computed. An implicit AS model of the MS which is free of singularity, is extracted as the zero contours of F. Furthermore, we generate a polynomial parametrization of the implicit MS, which allows for an efficient point sampling on the MS, and thereby simplifies the accurate estimation of integrals needed for electrostatic solvation energy calculations.

1 Introduction

The computation of electrostatic solvation energy (also known as polarization energy) for biomolecules plays an important role in the molecular dynamics simulation [23], the analysis of stability in protein structure prediction [35], and the protein-ligand binding energy calculation [24]. The explicit model of the solvent provides the most rigorous solvation energy calculation [31]. However, due to the large amount of solvent molecules, most of the computer time is spent on the trajectories of the solvent molecules, which greatly reduces the efficiency of this method [33]. An alternative method is to represent the solvent implicitly as a dielectric continuum [34] which can be modeled by the Poisson-Boltzmann (PB) equation [10,29]. A more efficient approximation to the PB electrostatic solvation energy is the generalized Born (GB) model [36,11,27], which approximates the electrostatic solvation energy ΔG_{elec} as

$$G_{\text{pol}} = -\frac{\tau}{2} \sum_{i,j} \frac{q_i q_j}{[r_{ij}^2 + R_i R_j \exp(-\frac{r_{ij}^2}{F R_i R_j})]^{\frac{1}{2}}} \tag{1}$$

where $\tau = \frac{1}{\varepsilon_p} - \frac{1}{\varepsilon_w}$, ε_p is the solute (low) dielectric constant, ε_w is the solvent (high) dielectric constant, q_i is the atomic charge of atom i, r_{ij} is the distance between atom i and j, F is an empirical factor (4 [36], or 8 [27]), and R_i is the effective Born radius of atom i. The effective Born radius reflects how deeply an atom is buried in the molecule: the deeper it is buried, the larger the radius is and consequently the less important to the polarization. The effective Born radii can be computed by the surface integral [19]

$$R_i^{-1} = \frac{1}{4\pi} \int_{\Gamma} \frac{(\mathbf{r} - \mathbf{x}_i) \cdot \mathbf{n}(\mathbf{r})}{|\mathbf{r} - \mathbf{x}_i|^4} \, dS \tag{2}$$

where Γ is the MS of the solute, $\mathbf{x}_i$ is the center of atom i, and $\mathbf{n}(\mathbf{r})$ is the unit normal of the surface at $\mathbf{r}$. The derivation of (2) and an algorithm for the fast evaluation of (2) based on the fast Fourier transform (FFT) is discussed in detail in [9]. The integration in (2) may be more accurately approximated by a Gaussian quadrature over the 3D MS. Since the Gaussian quadrature over a regular 2D planar domain can be easily achieved, this motivates us to define Γ analytically and parameterize Γ into a regular 2D domain.

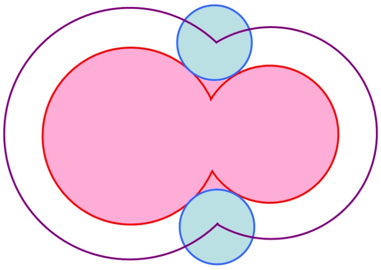

Fig. 1. The different molecular surfaces are shown for a two atom model in 2D. The VWS is shown as the red curve. The SAS is illustrated as the curve in purple. The SES is the envelope of the pink region. The solvent probe is illustrated as the blue ball.

Three molecular surface well known are shown in Figure 1 in 2D. The van der Waals surface (VWS) is the union of a set of spheres with atomic van der Waals radii. The solvent accessible surface (SAS) is the union of a set of spheres with the van der Waals radii augmented by the solvent probe radius (normally taken as $1.4\mathring{A}$) [26]. The solvent excluded surface (SES, also called molecular surface or Connolly surface) is the boundary of the union of all possible solvent probes that do not overlap with the VWS [14,32]. In practice

SES can be got by starting with SAS and removing the volume occupied by the solvent probe. Following [14], the SES consists of convex spherical patches which are parts of the VWS, concave spherical patches which lie on the probes contacting three atoms, and toroidal patches that are defined by the rolling probe touching two atoms. The VWS and SAS lead to an overestimation and underestimation of the electrostatic solvation energy, respectively [27], however SES is advanced to the other two and is most often used as the MS in the energetic calculation. The SES is proper for the stable energy calculation except for one significant drawback: cusps caused by self-intersection of the rolling probe lead to singularity in the Born radii calculation and their differentiation computation as well.

In the energetic computation, an analytical molecular surface model is needed and the singularity should be avoided. One way for generating a smooth model to approximate the SES is to define an analytical volumetric density function, such as the summation of Gaussian functions [20], Fermi-Dirac switching function [28], summation [27] or product [22] of piecewise polynomials, and construct the surface as an iso-contour of the density function. Efficient methods have been developed to construct the iso-contour of the smooth kernel functions [2,7]. However the error of the generated isosurface could be very large and result in inaccurate energy computation. Bajaj et al [5] presented a NURBS representation for the surface. Though it provides a parametric approximation to the SES, it still has singularity over the surface. Edelsbrunner [18] defines another paradigm of a smooth surface referred to as *skin* which is based on Voronoi, Delaunay, and Alpha complexes of a finite set of weighed points. The *skin* model has good geometric properties that it is free of singularity and it can be decomposed into a collection of quadratic patches. However when it is applied to the energetic computation, the skin triangulation has to be used [12,13], which only provides a linear approximation and therefor is dense. The dense triangulation causes oversampling on the surface and reduces the efficiency. Therefore it still remains a challenge to generate a model for the molecular surface which is accurate, smooth, and computable.

The main contribution of this paper is a method to model the SES as piecewise algebraic spline patches with certain continuity at the boundary of the patches. Each patch has its dual implicit and parametric representation. Therefore the surface of higher degree is parameterized over a planer domain which makes it convenient to implement numerical quadrature involved in the energetic computation. The idea of constructing a simplical hull over the triangulation and generating the piecewise smooth patches is first introduced in [15] where a series of tetrahedra are built and the quadric patches are generated. This idea is extended to cubic patches construction in [21,16], nonsingular and single sheeted cubic patches in [3]. Here the patches which are built based upon the prism scaffold surrounding the triangulation of the SES are defined implicitly and much more simple by the BB spline functions. Compared with the linear models, fewer number of triangles are needed to get the similar result. Besides, it is error bounded and in certain circumstances it is free of singularity.

The paper is organized as follows: Section 2 describes in detail the generation of the algebraic spline molecular surface (ASMS). Section 3 discusses the approximation of ASMS, its application to the energetic computation, and provides some examples.

2 Algebraic spline model

2.1 Algorithm Sketch

There are basically four steps in our ASMS construction algorithm:

- construct a triangulation mesh of the MS (Section 2.2);
- build a prism scaffold based on the triangulation (Section 2.3);
- generate piecewise algebraic patches that match with certain continuity (Section 2.3);
- parameterize the patches and implement the quadrature over the parametric domain (Section 2.5).

2.2 Generation of a triangulation mesh of Γ

So far a lot of work has been done for the triangulation of Γ that approximates the MS [1,25,13,37,6]. We could use any of these triangulation meshes as the initial of our algorithm. In the current research we choose to use the triangulation generated by a program embedded in the software TexMol [4,6]. Molecular features are well preserved in this triangulation. We further apply a procedure of triangulation decimation [8] to obtain a coarser mesh.

2.3 Generation of implicit/parametric patches

Given the triangulation mesh T, let $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$ be a triangle of T where $\mathbf{v}_i,\ \mathbf{v}_j,\ \mathbf{v}_k$ are the vertices of the triangle. Suppose the unit normals of the surface at the vertices are also known, denoted as $\mathbf{n}_l,\ (l = 1, j, k)$. Let $\mathbf{v}_l(\lambda) = \mathbf{v}_l + \lambda\mathbf{n}_l$. First we define a prism $D_{ijk} := \{\mathbf{p} :\ \mathbf{p} = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda),\ \lambda \in I_{ijk}\}$, where (b_1, b_2, b_3) are the barycentric coordinates of points in $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$, and I_{ijk} is a maximal open interval such that $0 \in I_{ijk}$ and for any $\lambda \in I_{ijk}$, $\mathbf{v}_i(\lambda)$, $\mathbf{v}_j(\lambda)$

and $\mathbf{v}_k(\lambda)$ are not collinear and $\mathbf{n}_i$, $\mathbf{n}_j$ and $\mathbf{n}_k$ point to the same side of the plane $P_{ijk}(\lambda) := \{\mathbf{p} : \ \mathbf{p} = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda)\}$ (Figure 2).

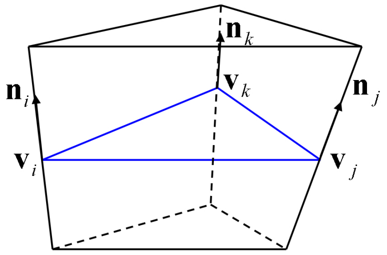

Fig. 2. A prism D_{ijk} constructed based on the triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$.

Next we define a function over the prism D_{ijk} in the Benstein-Bezier (BB) form:

$$F(b_1, b_2, b_3, \lambda) = \sum_{i+j+k=n} b_{ijk}(\lambda)B^n_{ijk}(b_1, b_2, b_3) \tag{3}$$

where $B^n_{ijk}(b_1, b_2, b_3)$ is the Bezier basis

$$B^n_{ijk}(b_1, b_2, b_3) = \frac{n!}{i!j!k!}b_1^i b_2^j b_3^k$$

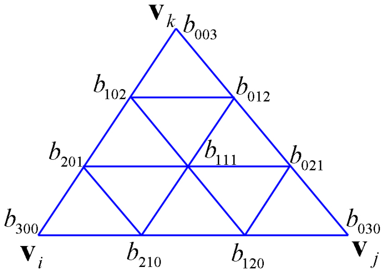

Fig. 3. The control coefficients of the cubic Bezier basis of function F

We approximate the molecular surface by the zero contour of F, denoted as S. In order to make S smooth, n, the degree of the Bezier basis, should be no less than 3. To avoid redundancy, we consider the case of $n = 3$. The control coefficients should be defined properly such that the continuity of S is satisfied. Figure 3 shows the relationship of the control coefficients and the points of the triangle when $n = 3$. Next we are going to discuss how to define these coefficients.

Since S passes through the vertices v_i, v_j, v_k, we know that

$$b_{300} = b_{030} = b_{003} = \lambda \tag{4}$$

To obtain C^1 continuity at the vertices, we let $b_{210} - b_{300} = \frac{1}{3}\nabla F(v_i) \cdot (\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda))$, where $\nabla F(v_i) = \mathbf{n}_i$. Therefore

$$b_{210} = \lambda + \frac{1}{3}\mathbf{n}_i \cdot (\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)) \tag{5}$$

b_{120}, b_{201}, b_{102}, b_{021}, b_{012} are defined similarly.

We also assume that S is C^1 continuous at the midpoints of the edges of T. We define b_{111} using using the side-vertex scheme [30]:

$$b_{111} = w_1 b_{111}^{(1)} + w_2 b_{111}^{(2)} + w_3 b_{111}^{(3)} \tag{6}$$

where

$$w_i = \frac{b_j^2 b_k^2}{b_2^2 b_3^2 + b_1^2 b_3^2 + b_1^2 b_2^2}, \qquad i = 1, 2, 3;\ i \neq j \neq k$$

Next we are going to explain how $b_{111}^{(1)}$, $b_{111}^{(2)}$ and $b_{111}^{(3)}$ are defined such that the C^1 continuity is obtained at the midpoint of the edge $\mathbf{v}_j\mathbf{v}_k$, $\mathbf{v}_i\mathbf{v}_k$ and $\mathbf{v}_i\mathbf{v}_j$. Consider the the edge $\mathbf{v}_i\mathbf{v}_j$ in the prism D_{ijk}. Recall that any point $\mathbf{p} := (x, y, z)$ in D_{ijk} can be represented by

$$(x, y, z)^T = b_1 \mathbf{v}_i(\lambda) + b_2 \mathbf{v}_j(\lambda) + b_3 \mathbf{v}_k(\lambda) \tag{7}$$

Therefore, after differentiate both sides of (7) with respect to x, y and z respectively, we get

$$I_3 = \begin{pmatrix} \frac{\partial b_1}{\partial x} & \frac{\partial b_2}{\partial x} & \frac{\partial \lambda}{\partial x} \\ \frac{\partial b_1}{\partial y} & \frac{\partial b_2}{\partial y} & \frac{\partial \lambda}{\partial y} \\ \frac{\partial b_1}{\partial z} & \frac{\partial b_2}{\partial z} & \frac{\partial \lambda}{\partial z} \end{pmatrix} \begin{pmatrix} (\mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda))^T \\ (b_1\mathbf{n}_i + b_2\mathbf{n}_j + b_3\mathbf{n}_k)^T \end{pmatrix} \tag{8}$$

where I_3 is a 3×3 unit matrix. Let

$$T = \begin{pmatrix} (\mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda))^T \\ (b_1\mathbf{n}_i + b_2\mathbf{n}_j + b_3\mathbf{n}_k)^T \end{pmatrix} \tag{9}$$

Let $A = \mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda)$, $B = \mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda)$ and $C = b_1\mathbf{n}_i + b_2\mathbf{n}_j + b_3\mathbf{n}_k$, then

$$T = \begin{pmatrix} A, & B, & C \end{pmatrix}^T$$

By (8) we have

$$\begin{pmatrix} \frac{\partial b_1}{\partial x} & \frac{\partial b_2}{\partial x} & \frac{\partial \lambda}{\partial x} \\ \frac{\partial b_1}{\partial y} & \frac{\partial b_2}{\partial y} & \frac{\partial \lambda}{\partial y} \\ \frac{\partial b_1}{\partial z} & \frac{\partial b_2}{\partial z} & \frac{\partial \lambda}{\partial z} \end{pmatrix} = T^{-1} = \frac{1}{\det(T)} (B \times C,\ C \times A,\ A \times B) \tag{10}$$

According to (3), at $(b_1, b_2, b_3) = (\frac{1}{2}, \frac{1}{2}, 0)$ (the midpoint of $\mathbf{v}_i\mathbf{v}_j$), we have

$$\begin{pmatrix} \frac{\partial F}{\partial b_1} \\ \frac{\partial F}{\partial b_2} \\ \frac{\partial F}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} (\mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{n}_i + \mathbf{n}_j)^T/2 \end{pmatrix} \left(\frac{\mathbf{n}_i + \mathbf{n}_j}{4}\right) + \begin{pmatrix} \frac{3}{2}(b_{210} - b_{111}) \\ \frac{3}{2}(b_{120} - b_{111}) \\ \frac{1}{2} \end{pmatrix}$$

By (6), at $(b_1, b_2, b_3) = (\frac{1}{2}, \frac{1}{2}, 0)$ we have

$$b_{111} = b_{111}^{(3)}$$

Therefore the gradient at $(\frac{1}{2}, \frac{1}{2}, 0)$ is

$$\begin{aligned} \nabla F &= T^{-1}(\frac{\partial F}{\partial b_1}, \frac{\partial F}{\partial b_2}, \frac{\partial F}{\partial \lambda})^T \\ &= \frac{\mathbf{n}_i + \mathbf{n}_j}{4} + \frac{1}{2\det(T)}[3(b_{210} - b_{111}^{(3)})B \times C + 3(b_{120} - b_{111}^{(3)})C \times A + A \times B] \end{aligned} \tag{11}$$

Let

$$\begin{aligned} &\mathbf{d}_1(\lambda) = \mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda) = B - A \\ &\mathbf{d}_2(b_1, b_2, b_3) = b_1\mathbf{n}_i + b_2\mathbf{n}_j + b_3\mathbf{n}_k = C \\ &\mathbf{d}_3(b_1, b_2, b_3, \lambda) = \mathbf{d}_1 \times \mathbf{d}_2 = B \times C + C \times A \end{aligned} \tag{12}$$

Define

$$\mathbf{c} = C(\frac{1}{2}, \frac{1}{2}, 0)\mathbf{d}_3(\lambda) = \mathbf{d}_3(\frac{1}{2}, \frac{1}{2}, 0, \lambda) = B \times \mathbf{c} + \mathbf{c} \times A \tag{13}$$

Let $\nabla F = \nabla F(\frac{1}{2}, \frac{1}{2}, 0)$. In order to make S be C^1 at $(\frac{1}{2}, \frac{1}{2}, 0)$, we should have $\nabla F \cdot \mathbf{d}_3(\lambda) = 0$. Therefore, by (11) and (12), we have

$$b_{111}^{(3)} = \frac{\mathbf{d}_3(\lambda)^T(3b_{210}B \times \mathbf{c} + 3b_{120}\mathbf{c} \times A + A \times B)}{3||\mathbf{d}_3(\lambda)||^2} \tag{14}$$

Similarly, we may define $b_{111}^{(1)}$ and $b_{111}^{(2)}$. For the surface evaluation, given the barycentric coordinates of a point (b_1, b_2, b_3) in triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$, we solve equation $F = 0$ for λ by Newton's method, where F is defined in (3). Then the corresponding point on S is

$$(x,\ y,\ z)^T = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda) \tag{15}$$

2.4 Smoothness

Theorem 1 *The ASMS S is C^1 at the vertices of T and the midpoints of the edges of T.*

Theorem 2 *S is C^1 everywhere if $\mathbf{n}_i \cdot (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{n}_j \cdot (\mathbf{v}_j - \mathbf{v}_i)$ for every edge $\mathbf{v}_i\mathbf{v}_j$ of T.*

Theorem 3 *S is C^1 everywhere if $\mathbf{n}_i = \mathbf{n}_j$ for every unit normal at the vertices of T.*

Proof of the Theorem 1, 2 and 3 are discussed in Appendix A.

2.5 Patch parametrization and quadrature

To compute (2), let$f = \frac{(\mathbf{r}-\mathbf{x}_i)\cdot\mathbf{n}(\mathbf{r})}{|\mathbf{r}-\mathbf{x}_i|^4}$, then

$$\int_\Gamma f(\mathbf{x})\ dS = \sum_j \int_{\Gamma_j} f(\mathbf{x})\ dS$$

where Γ_j is the zero contour of the BB function over the jth triangle. For any point $\mathbf{x} = (x, y, z)$ on Γ_j, by the inverse map of (15), one can be uniquely map $\mathbf{x}$ to a point (b_1, b_2, b_3) in the jth triangle ($[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$), where $b_3 = 1 - b_1 - b_2$. Therefore, we have

$$x = x(b_1, b_2,), \qquad y = y(b_1, b_2), \qquad z = z(b_1, b_2)$$

Thus,

$$\int_{\Gamma_i} f(\mathbf{x})\ dS = \int_{\sigma_i} f(x(b_1, b_2), y(b_1, b_2), z(b_1, b_2))\sqrt{EG - F^2}\ db_1 db_2 \tag{16}$$

where

$$\begin{aligned} E &= (\frac{\partial x}{\partial b_1})^2 + (\frac{\partial y}{\partial b_1})^2 + (\frac{\partial z}{\partial b_1})^2 \\ F &= \frac{\partial x}{\partial b_1}\frac{\partial x}{\partial b_2} + \frac{\partial y}{\partial b_1}\frac{\partial y}{\partial b_2} + \frac{\partial z}{\partial b_1}\frac{\partial z}{\partial b_2} \\ G &= (\frac{\partial x}{\partial b_2})^2 + (\frac{\partial y}{\partial b_2})^2 + (\frac{\partial z}{\partial b_2})^2 \end{aligned}$$

We apply the Gaussian quadrature to (16)

$$\int_{\sigma_i} f(b_1, b_2)\sqrt{EG - F^2}\ db_1 db_2 \approx \sum_{k=1}^{n} W_k f(b_1^k, b_2^k)\sqrt{EG - F^2}|_{b_1^k, b_2^k} \tag{17}$$

where (b_1^k, b_2^k) and W_k are the Gaussian integration nodes and weights for the triangle.

3 Approximation of the ASMS model and an application

3.1 Error bound and convergence

Now we consider the error between S and the true surface. Suppose $\mathbf{p} := (x, y, z)$ is a point on the true surface within the prism D_{ijk}. Because of (7), we know the volume coordinate (b_1, b_2, b_3, λ). The approximation point $\mathbf{p}'$ on S is $(b_1, b_2, b_3, \lambda')$, where λ' is the solution to the equation $F = 0$ with (b_1, b_2, b_3) known.

Lemma 3.1 *The error of the approximation point* $\mathbf{p}'$ *to the true point* $\mathbf{p}$ *is bounded by* $|\lambda - \lambda'|$.

Proof:

$$\begin{aligned} ||\mathbf{p} - \mathbf{p}'|| &\le b_1||\mathbf{v}_i(\lambda) - \mathbf{v}_i(\lambda')|| + b_2||\mathbf{v}_j(\lambda) - \mathbf{v}_j(\lambda')|| \\ &\quad + b_3||\mathbf{v}_k(\lambda) - \mathbf{v}_k(\lambda')|| \\ &\le |\lambda - \lambda'|(b_1||\mathbf{n}_i|| + b_2||\mathbf{n}_j|| + b_3||\mathbf{n}_k||) \\ &= |\lambda - \lambda'| \end{aligned}$$

Claim: Let h be the maximum side length of triangulation mesh T, P be the point on the true surface, P' be the corresponding point on the approximation surface, then P' converges to P at the rate of $O(h^3)$. i.e. There exists a constant C such that $||P - P'|| \le Ch^3$.

In order to show the error of S to the true surface, we test some exact surfaces $S_0 := z = f(x, y), (x, y) \in [0, 1]^2$ with f known. We first generate a triangulation mesh over the true surfaces. Based on the mesh, we construct the piecewise BB zero contour S. The error of S compared with the exact surface S_0 is defined as max $\frac{||\mathbf{p}-\mathbf{p}'||}{||\mathbf{p}||}$, where $\mathbf{p} \in S_0$, $\mathbf{p}' \in S$, and the pair of points $(\mathbf{p},\ \mathbf{p}')$ are mapped to the same point (b_1, b_2, b_3) on the triangle with different λ. With the maximum edge length of the triangles $h = 0.1$, we sample $(\mathbf{p},\ \mathbf{p}')$ on the surfaces and compute the maximum relative error (Table 1).

To show the convergence of S to S_0, we gradually refine the mesh and compute the ratio of the maximum difference of λ and λ' to h^3. For each of these functions, as h decreases, we observe that the ratio converges to a constant C (Table 1) This result verifies our claim.

Function $(x, y) \in [0,1]^2$	$\max\{\frac{\|\|\mathbf{p}-\mathbf{p}'\|\|}{\|\|\mathbf{p}\|\|}\}$	C
$z = 0$	0	0
$z = x^2 + y^2$	2.450030e-05	1.010636e-2
$z = x^3 + y^3$	1.063699e-04	2.610113e-2
$z = e^{-\frac{1}{4}[(x-0.5)^2+(y-0.5)^2]}$	5.286856e-07	6.288604e-5
$z = 1.25 + \frac{\cos(5.4y)}{6+6(3x-1)^2}$	2.555683e-04	4.58608e-2
$z = \tanh(9y - 9x)$	1.196519e-02	1.896754e-1
$z = \sqrt{1 - x^2 - y^2}$	8.614969e-05	1.744051e-1 (h^4)
$z = [(2 - \sqrt{1-y^2})^2 - x^2]^{1/2}$	1.418242e-05	1.748754e-02

Table 1
For some typical explicit surfaces (column 1), we compute the maximum relative error (the ratio of the Euclidian distance between $\mathbf{p}$ and $\mathbf{p}'$ to the norm of $\mathbf{p}$) in column 2 with $h = 0.1$, $(x, y) \in [0,1]^2$. By letting $h \downarrow 0$, we compute the limit of $\frac{|\lambda-\lambda'|}{h^3}$ and observe a constant C (column 3) which verifies the rate of convergence of S we claimed. For certain cases, the rate of convergence can be as good as $O(h^4)$.

3.2 Biomolecular energetic computation

We apply our ASMS to the GB model to compute the electrostatic solvation energies. As examples, we start from the initial triangulation derived from the PDB (Protein Data Bank) atomic structure for three protein models: 1CGI (852 atoms, Figure 5 (a)), 1HIA (693 atoms, Figure 4 (a)), and 1PPE (436 atoms, Figure 5 (c)). We generate the surface S and compute the electrostatic solvation energy based on S. For the numerical integration (17), 4-point Gaussian quadrature rule over a triangle given in [17] is applied. We repeat the generation and computation for reduced number of triangles. Figure 4 (d), Figure 5 (b) and Figure 5 (d) show the piecewise algebraic surface generated from the decimated triangulation. Table 2 compares the computed polarization energy with different number of triangles for the proteins. As we see, while a coarser triangulation is used (roughly 1/3 of the original triangulation number), we still get similar results as the initial fine triangulation. We also do the computation on the piecewise linear surface. It turns out that much fewer number of triangles are needed if the piecewise algebraic surface is used.

4 Conclusions

We have introduced a method to generate a model for the MS. The main steps of the method are the following: construct a triangular prism scaffold covering the PDB structure, and then define nearly C^1 piecewise polynomials, BB

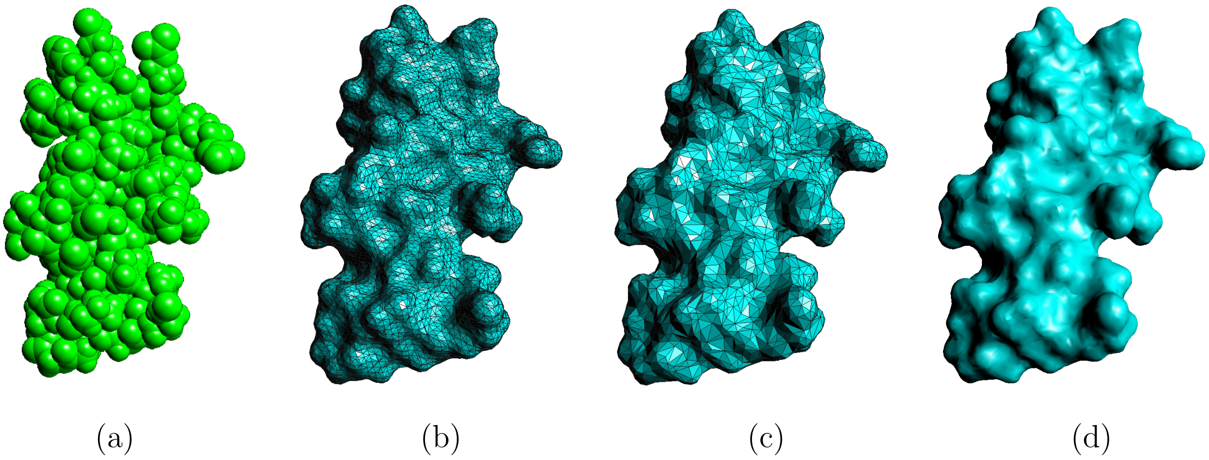

Fig. 4. Molecular models of a protein(1HIA). (a) The van der Waals surface model (693 atoms); (b) An initial triangulated solvent excluded surface (SES) model (27480 triangles); (c) The decimated triangulated SES model(7770 triangles); (d) Our algebraic spline molecular surface model (7770 patches) generated from (c).

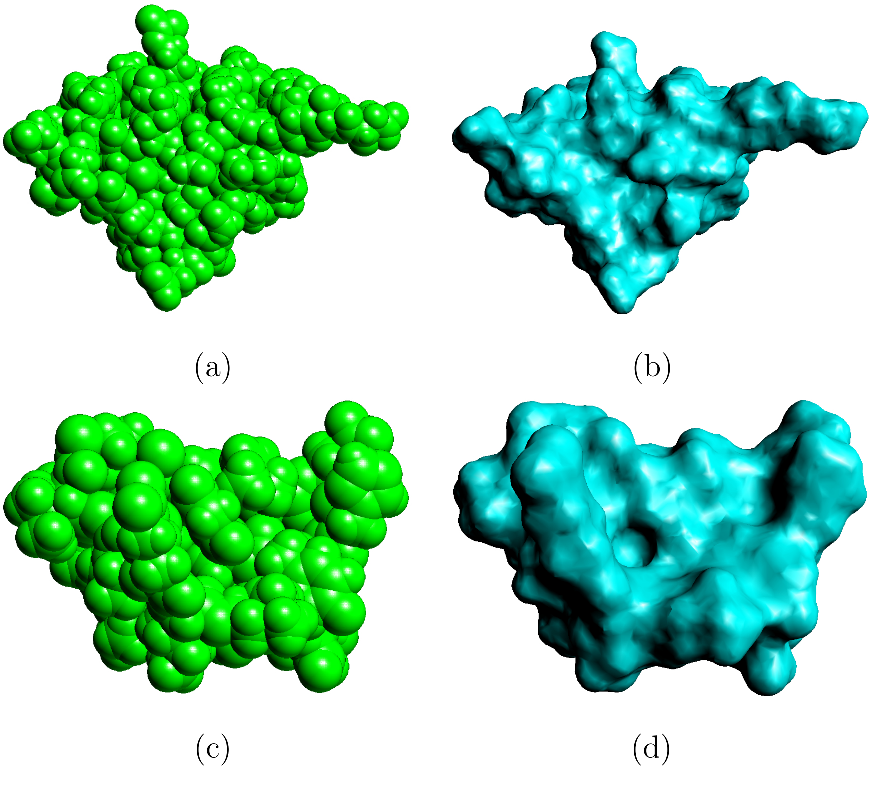

Fig. 5. (a) The atomic model of protein 1CGI (852 atoms); (b) The piecewise algebraic surface of 1CGI (8712 patches); (c) The atomic model of protein 1PPE (436 atoms); (d) The piecewise algebraic surface of 1PPE (6004 patches).

Protein	No. of	G_{pol}, kcal/mol (timing, s)	
ID	Triangles	piecewise linear	AS
1CGI	29108	-1371.741894 (39.64)	-1343.1496 (40.31)
	8712	-1399.194841 (12.94)	-1346.2230 (12.64)
	3674	-1678.444735 (7.40)	-1394.2270 (6.11)
1HIA	27480	-1361.226603 (30.23)	-1340.6384 (31.18)
	7770	-1389.017538 (9.43)	-1347.8067 (9.93)
	3510	-1571.890827 (5.21)	-1388.4665 (5.21)
1PPE	24244	-835.563905 (17.27)	-825.3252 (18.26)
	6004	-852.713039 (5.09)	-828.2158 (5.39)
	2748	-933.956234 (2.74)	-845.5085 (3.27)

Table 2
Comparison of the electrostatic solvation energy computed by the piecewise linear surface and the ASMS. With fewer number of triangles used, the ASMS gives a result better than the linear surface which needs more number of triangles. The running time contains the time needed for computing the integration nodes over the surfaces, computing the Born radii, and evaluating G_{pol}.

functions, over the collection of prisms. The model to the molecular surface is the union of the zero contours of the piecewise BB functions. This model is extremely convenient for fast electrostatic solvation energy calculation. We should mention that, while not detailed in this paper, the algorithm of Section 2.3 can, by repeated evocation, yield a hierarchical multiresolution spline model of the MS. In the future research we could extend this algebraic patch model to the computation of electrostatic solvation forces which are the main driven forces in molecular dynamics simulation. The main tasks of the force calculation also involve fast numerical integration. It is challenging because the integration domain is not a surface but a fat skin over the atoms.

Acknowledgment. We wish to thank Vinay Siddavanahalli for helping in the implementation of our ASMS within TexMol, a molecular modeling and visualization software of our center (http://cvcweb.ices.utexas.edu/software/#TexMol). A substantial part of this work in this paper was done when Guoliang Xu was visiting Chandrajit Bajaj at UT-CVC. His visit was supported by the J. T. Oden ICES visitor fellowship.

A Appendix

Proof of Theorem 1: It is obvious that S is C^1 at the vertices. For the continuity at the midpoints of edges, let us consider the edge $\mathbf{v}_i\mathbf{v}_j$ in triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$. On the edge $\mathbf{v}_i\mathbf{v}_j$, $b_3 = 0$. So we may let $b_2 = t$ and $b_1 = 1 - t$. Then matrix T can be written as

$$T(t) = \begin{pmatrix} (\mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda))^T \\ (\mathbf{n}_i + t(\mathbf{n}_j - \mathbf{n}_i))^T \end{pmatrix}$$

and

$$T^{-1} = \frac{1}{\det(T)} \left(B \times C,\ C \times A,\ A \times B\right)$$

where $A = \mathbf{v}_i(\lambda) - \mathbf{v}_k(\lambda)$, $B = \mathbf{v}_j(\lambda) - \mathbf{v}_k(\lambda)$ and $C(t) = \mathbf{n}_i + t(\mathbf{n}_j - \mathbf{n}_i)$. Therefore on the edge $\mathbf{v}_i\mathbf{v}_j$,

$$\begin{pmatrix} \frac{\partial F}{\partial b_1} \\ \frac{\partial F}{\partial b_2} \\ \frac{\partial F}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} A^T \\ B^T \\ C^T \end{pmatrix} (\mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2) + \begin{pmatrix} 3(b_{210} - b_{111}) \\ 3(b_{120} - b_{111}) \\ 1 \end{pmatrix} 2t(1-t)$$

The gradient of F on the edge $\mathbf{v}_i\mathbf{v}_j$ can be written as

$$\begin{aligned} \nabla F =& \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + T^{-1} \begin{pmatrix} 3(b_{210} - b_{111}) \\ 3(b_{120} - b_{111}) \\ 1 \end{pmatrix} 2t(1-t) \\ =& \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + \frac{2t(1-t)}{\det(T)(t)} \Big[3(B \times C(t))(b_{210} - b_{111}) \\ &+ 3(C(t) \times A)(b_{120} - b_{111}) + A \times B \Big] \end{aligned} \tag{A.1}$$

When $t = \frac{1}{2}$, $C(\frac{1}{2}) = \mathbf{c}$, therefore

$$B \times C(t) + C(t) \times A = \mathbf{d}_3(\lambda)$$

Take the function inside the square bracket of (A.1) into account. Name this function as F_1.

$$F_1 = 3(B \times \mathbf{c})b_{210} + 3(\mathbf{c} \times A)(b_{120} + A \times B - 3(B \times \mathbf{c} + \mathbf{c} \times A)b_{111} \tag{A.2}$$

Since on the edge $\mathbf{v}_i\mathbf{v}_j$, $b_{111} = b_{111}^{(3)}$, substituting (14) into (A.2), we get F_1 is 0. Therefore, at the midpoint

$$\nabla F = (\mathbf{n}_i + \mathbf{n}_j)/4 \tag{A.3}$$

So S is C^1 continuous at the midpoints of the edges. □

Proof of Theorem 2: It is obvious that S is C^1 within the triangles. By Theorem 1 we have already known that S is C^1 at the vertices and the midpoints of the edges. Here we only need to show S is C^1 at any points of the edges, let us consider the the edge $\mathbf{v}_i\mathbf{v}_j$ in the triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$.

Under the condition $\mathbf{n}_i \cdot (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{n}_j \cdot (\mathbf{v}_j - \mathbf{v}_i)$, we have $b_{120} = b_{210}$, so (A.1) is written as

$$\begin{aligned} \nabla F & = \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 \\ +\frac{2t(1-t)}{\det(T(t))} & [3(b_{210} - b_{111})(B-A) \times C + A \times B] \end{aligned} \tag{A.4}$$

Similar as (12), we define

$$\mathbf{d}_3(t,\lambda) = (B-A) \times C(t) \tag{A.5}$$

By (14), with the facts that $b_{120} = b_{210}$ and $b_{111} = b_{111}^{(3)}$ on edge $\mathbf{v}_i\mathbf{v}_j$, we have

$$b_{210} - b_{111} = -\frac{\mathbf{d}_3^T(\lambda)(A \times B)}{3||\mathbf{d}_3(\lambda)||^2} \tag{A.6}$$

where $\mathbf{d}_3(\lambda)$ is defined in (13). Plug (A) and (A.6) in (A), we get

$$\begin{aligned} \nabla F & = \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 \\ +\frac{2t(1-t)}{||\mathbf{d}_3(\lambda)||^2} & \left[\frac{||\mathbf{d}_3(\lambda)||^2 A \times B - \mathbf{d}_3(t,\lambda)\mathbf{d}_3^T(\lambda) A \times B}{\det(T(t))}\right] \end{aligned} \tag{A.7}$$

Consider the function inside the square bracket of (A) and name it as F_2. Our goal is to show that $F_2 = 0$. Since we have already known that when $t = \frac{1}{2}$, $F_2 = 0$ which prompts us to compute the derivative of F_2 with respect to t and see if the derivative is 0. We observe that both the numerator of the denominator of F_2 are linear in terms of t, so F_2 is of the form $\frac{at+b}{ct+d}$ with

$$\begin{aligned} a &= (\mathbf{n}_j - \mathbf{n}_i) \times (B-A)\mathbf{d}_3^T(\lambda) A \times B \\ b &= ||\mathbf{d}_3(\lambda)||^2 A \times B + \mathbf{n}_i \times (B-A)\mathbf{d}_3^T(\lambda) A \times B \\ c &= (\mathbf{n}_j - \mathbf{n}_i)^T (A \times B) \\ d &= \mathbf{n}_i^T (A \times B) \end{aligned}$$

In order to show $\frac{\partial F_2}{\partial t} = 0$, it is equivalent to show $N := ad - bc = 0$. As we computed,

$$\begin{aligned} N = &[\mathbf{n}_j \times (B - A)\mathbf{d}_3^T A \times B]\mathbf{n}_i^T (A \times B) \\ &- (||\mathbf{d}_3(\lambda)||^2 A \times B)(\mathbf{n}_j - \mathbf{n}_i) \times (B - A) \\ &- [\mathbf{n}_i \times (B - A)\mathbf{d}_3^T A \times B]\mathbf{n}_j^T (A \times B) \end{aligned} \tag{A.8}$$

Under the condition $\mathbf{n}_i \cdot (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{n}_j \cdot (\mathbf{v}_j - \mathbf{v}_i)$, we have $(B - A)^T \mathbf{c} = (\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda))^T \mathbf{c} = 0$, where $\mathbf{c} = C(\frac{1}{2}, \frac{1}{2}, 0)$. Therefore

$$||\mathbf{d}_3(\lambda)||^2 = ((B - A) \times \mathbf{c}) \cdot ((B - A) \times \mathbf{c}) = ||\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)||^2 ||\mathbf{c}||^2 \tag{A.9}$$

$$\begin{aligned} &\mathbf{d}_3^T(\lambda) A \times B = \mathbf{d}_3^T(\lambda) A \times (B - A) \\ =&((B - A) \times \mathbf{c}) \cdot (A \times (B - A)) = -\mathbf{c}^T A ||\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)||^2 \end{aligned} \tag{A.10}$$

Plug (A.9)and (A.10) into (A.8) and divide both sides by $||\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)||^2$, we get

$$\begin{aligned} F_3 :=&\frac{N}{||(\mathbf{v}_j - \mathbf{v}_i)(\lambda)||^2} \\ =& - \mathbf{n}_j \times (B - A)\mathbf{c}^T A \mathbf{n}_i^T (A \times B) - ||\mathbf{c}||^2 A \times B(\mathbf{n}_j - \mathbf{n}_i)^T A \times B \\ &+ (\mathbf{n}_i \times (B - A)\mathbf{c}^T A)\mathbf{n}_j^T (A \times B) \\ =&[(\mathbf{c}^T A \mathbf{n}_i - ||\mathbf{c}||^2 A) \times (B - A)]\mathbf{n}_j^T (A \times B) \\ &+ [(||\mathbf{c}||^2 A - \mathbf{c}^T A \mathbf{n}_j) \times (B - A)]\mathbf{n}_i^T (A \times B) \end{aligned} \tag{A.11}$$

If $\mathbf{n}_i = \mathbf{n}_j$, we have (A.11) being 0. Now let us assume $\mathbf{n}_i \neq \mathbf{n}_j$. We have already had $\mathbf{c} = \frac{1}{2}(\mathbf{n}_i + \mathbf{n}_j)$. Now let us define another vector $\mathbf{e} = \frac{1}{2}(\mathbf{n}_i - \mathbf{n}_j)$ and let $D = B - A$. Then $\mathbf{c}$ is orthogonal to $\mathbf{e}$ and D:

$$\mathbf{c}^T \mathbf{e} = 0, \qquad \mathbf{c}^T D = 0 \tag{A.12}$$

Furthermore

$$\mathbf{c} \times (D \times \mathbf{e}) = 0 \tag{A.13}$$

By the definition of $\mathbf{c}$ and $\mathbf{e}$,

$$\mathbf{n}_i = \mathbf{c} + \mathbf{e}, \qquad \mathbf{n}_j = \mathbf{c} - \mathbf{e} \tag{A.14}$$

Substitute (A.14) into (A.11) and replace $A \times B$ with $A \times D$, we get

$$\begin{aligned} F_3 =&[\mathbf{c}^T A(\mathbf{c} + \mathbf{e}) - ||\mathbf{c}||^2 A] \times D(\mathbf{c} - \mathbf{e})^T (A \times D) \\ &+ [||\mathbf{c}||^2 A - \mathbf{c}^T A(\mathbf{c} - \mathbf{e})] \times D(\mathbf{c} + \mathbf{e})^T (A \times D) \\ =&2\mathbf{c}^T A(\mathbf{e} \times D)\mathbf{c}^T (A \times D) - 2[\mathbf{c}^T A\mathbf{c} - ||\mathbf{c}||^2 A] \times D\mathbf{e}^T (A \times D) \end{aligned} \tag{A.15}$$

If $\mathbf{e}$ and D are linearly dependent, then $\mathbf{e} \times D = 0$ and moreover $\mathbf{e}^{(} A \times D) = 0$.

Therefore $F_3 = 0$. Otherwise, we introduce a new matrix

$$M = \begin{pmatrix} D^T \\ \mathbf{c}^T \\ \mathbf{e}^T \end{pmatrix}$$

$\mathbf{c}$, $\mathbf{e}$, and D are linearly independent, so M is nonsingular. So F_3 (which is a vector) is equal to

$$\begin{aligned}
&2M^{-1}\begin{pmatrix} D^T \\ \mathbf{c}^T \\ \mathbf{e}^T \end{pmatrix}\Big(\mathbf{c}^T A(\mathbf{e}\times D)\mathbf{c}^T(A\times D) - [\mathbf{c}^T A\mathbf{c} - ||\mathbf{c}||^2 A]\times D\mathbf{e}^T(A\times D)\Big) \\
=&-2M^{-1}\begin{pmatrix} 0 \\ (-\mathbf{c}^T A\mathbf{c}^T(\mathbf{e}\times D) - ||\mathbf{c}||^2\mathbf{e}^T(A\times D))\mathbf{c}^T(A\times D) \\ (\mathbf{c}^T A\mathbf{e}^T(\mathbf{c}\times D) - ||\mathbf{c}||^2\mathbf{e}^T(A\times D))\mathbf{e}^T(A\times D) \end{pmatrix} \\
=&-2M^{-1}\begin{pmatrix} 0 \\ (\mathbf{c}^T A\mathbf{c}^T(D\times \mathbf{e}) - ||\mathbf{c}||^2 A^T(D\times\mathbf{e}))\mathbf{c}^T(A\times D) \\ (\mathbf{c}^T A\mathbf{c}^T(D\times \mathbf{e}) - ||\mathbf{c}||^2 A^T(D\times\mathbf{e}))\mathbf{e}^T(A\times D) \end{pmatrix} \\
=&-2[\mathbf{c}^T A\mathbf{c}^T(D\times\mathbf{e}) - ||\mathbf{c}||^2 A^T(D\times\mathbf{e})]M^{-1}\begin{pmatrix} 0 \\ \mathbf{c}^T(A\times D) \\ \mathbf{e}^T(A\times D) \end{pmatrix}
\end{aligned}$$

By the Lagrange's formula,

$$\mathbf{c}^T A\mathbf{c}^T(D\times\mathbf{e}) - ||\mathbf{c}||^2 A^T(D\times\mathbf{e}) = (\mathbf{c}\times A)\cdot(\mathbf{c}\times(D\times\mathbf{e})) \tag{A.16}$$

From (A.13),(A.16) is zero and consequently $F_3 = 0$. So far we have proved F_2 is independent of t. In the proof of Theorem 1, we know that $F_2 = 0$ at $t = \frac{1}{2}$. Therefore $F_2 = 0$ for all t and ∇F on the edge $\mathbf{v}_i\mathbf{v}_j$ is

$$\nabla F = \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2$$

So S is also C^1 on the edges. □

Proof of Theorem 3: As same as the proof of Theorem 2, we only need to show that S is C^1 on the edge $\mathbf{v}_i\mathbf{v}_j$. In the proof of Theorem 1, we have already derived the gradient function on the edge $\mathbf{v}_i\mathbf{v}_j$ (A.1)

$$\begin{aligned}\nabla F =& \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + \frac{2t(1-t)}{\det(T)(t)}[3(B \times C(t))(b_{210} - b_{111}) \\ &+ 3(C(t) \times A)(b_{120} - b_{111}) + A \times B]\end{aligned}$$

Let

$$F_4 = \frac{1}{\det(T)(t)}[3(B \times C(t))(b_{210} - b_{111}) + 3(C(t) \times A)(b_{120} - b_{111}) + A \times B] \tag{A.17}$$

Following the same idea of the proof the Theorem 2, we compute $\frac{\partial F_4}{\partial t}$. The numerator of $\frac{\partial F_4}{\partial t}$ is

$$\begin{aligned}&[3(B \times C'(t))(b_{210} - b_{111}) + 3(C'(t) \times A)(b_{120} - b_{111}) + A \times B] \det(T) \\ &- \det(T)'(t)[3(B \times C(t))(b_{210} - b_{111}) + 3(C(t) \times A)(b_{120} - b_{111}) + A \times B]\end{aligned} \tag{A.18}$$

Since

$$\begin{aligned}C'(t) &= \mathbf{n}_j - \mathbf{n}_i \\ \det(T)'(t) &= (\mathbf{n}_j - \mathbf{n}_i)^T (A \times B)\end{aligned}$$

when $\mathbf{n}_i = \mathbf{n}_j$, (A.18) is 0. So F_4 is independent of t. By the proof of Theorem 1, $F_4 = 0$ at $t = \frac{1}{2}$. So $F_4 = 0$ for all t. The C^1 continuity of S is thus proved. □

References

[1] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Applied Mathematics*, 71:5–22, 1996.

[2] C. Bajaj, J. Castrillon-Candas, V. Siddavanahalli, and Z. Xu. Compressed representations of macromolecular structures and properties. *Structure*, 13:463–471, 2005.

[3] C. Bajaj, J. Chen, and G. Xu. Modeling with cubic A-patches. *ACM Transactions on Graphics*, 14:103–133, 1995.

[4] C. Bajaj, P. Djeu, V. Siddavanahalli, and A. Thane. Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. *Proc. of the Annual IEEE Visualization Conference*, pages 243–250, 2004.

[5] C. Bajaj, H. Lee, R. Merkert, and V. Pascucci. Nurbs based b-rep models from macromolecules and their properties. *In Proceedings of Fourth Symposium on Solid Modeling and Applications*, pages 217–228, 1997.

[6] C. Bajaj and V. Siddavanahalli. An adaptive grid based method for computing molecular surfaces and properties. ICES Technical Report TR-06-57, 2006.

[7] C. Bajaj and V. Siddavanahalli. Fast error-bounded surfaces and derivatives computation for volumetric particle data. ICES Technical Report TR-06-06, 2006.

[8] C. Bajaj, G. Xu, R. Holt, and A. Netravali. Hierarchical multiresolution reconstruction of shell surfaces. *Computer Aided Geometric Design*, 19:89–112, 2002.

[9] C. Bajaj and W. Zhao. Fast and accurate generalized born solvation energy computations. *Manuscript*, 2006.

[10] N. Baker, M. Holst, and F. Wang. Adaptive multilevel finite element solution of the poisson-boltzmann equation ii. refinement at solvent-accessible surfaces in biomolecular systems. *J. Comput Chem.*, 21:1343–1352, 2000.

[11] D. Bashford and D. A. Case. Generalized born models of macromolecular solvation effects. *Annu. Rev. Phys. Chem.*, 51:129–152, 2000.

[12] H. Cheng and X. Shi. Guaranteed quality triangulation of molecular skin surfaces. *IEEE Visualization*, pages 481–488, 2004.

[13] H. Cheng and X. Shi. Quality mesh generation for molecular skin surfaces using restricted union of balls. *IEEE Visualization*, pages 51–58, 2005.

[14] M. L. Connolly. Analytical molecular surface calculation. *J. Appl. Cryst*, 16:548–558, 1983.

[15] W. Dahmen. Smooth piecewise quadratic surfaces. In T. Lyche and L. Schumaker, editors, *Mathematical methods in computer aided geometric design*, pages 181–193. Academic Press, Boston, 1989.

[16] W. Dahmen and T-M. Thamm-Schaar. Cubicoids: modeling and visualization. *Computer Aided Geometric Design*, 10:89–108, 1993.

[17] D. Dunavant. High degree efficient symmetrical gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.

[18] H. Edelsbrunner. Deformable smooth surface design. *Discrete Computational Geometry*, 21:87–115, 1999.

[19] A. Ghosh, C. S. Rapp, and R. A. Friesner. Generalized born model based on a surface integral formulation. *J. Phys. Chem. B*, 102:10983–10990, 1998.

[20] J. A. Grant and B. T. Pickup. A gaussian description of molecular shape. *J. Phys. Chem.*, 99:3503–3510, 1995.

[21] B. Guo. *Modeling arbitrary smooth objects with algebraic surfaces.* PhD thesis, Cornell University, 1991.

[22] W. Im, M. S. Lee, and C. L. Brooks. Generalized born model with a simple smoothing function. *J. Comput Chem.*, 24:1691–1702, 2003.

[23] Martin Karplus and J. Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, 9:646–652, 2002.

[24] B. Kuhn and P. A. Kollman. A ligand that is predicted to bind better to avidin than biotin: insights from computational fluorine scanning. *J. Am. Chem. Soc*, 122:3909–3916, 2000.

[25] P. Laug and H. Borouchaki. Molecular surface modeling and meshing. *Engineering with Computers*, 18:199–210, 2002.

[26] B. Lee and F. M. Richards. The interpretation of protein structure: estimation of static accessiblilty. *J. Mol. Biol.*, 55:379–400, 1971.

[27] M. S. Lee, M. Feig, F. R. Salsbury, and C. L. Brooks. New analytic approximation to the standard molecular volume definition and its application to generalized born calculations. *J. Comput Chem.*, 24:1348–1356, 2003.

[28] M. S. Lee, F. R. Salsbury, and C. L. Brooks. Novel generalized born methods. *J Chemical Physics*, 116:10606–10614, 2002.

[29] J. D. Madura, J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Ilin, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. R. Scott, and J. A. McCammon. Electrostatics and diffusion of molecules in solution: simulations with the university of houston brownian dynamics program. *Computer Physics Communications*, 91:57–95, 1995.

[30] G. Nielson. The side-vertex method for interpolation in triangles. *J. Apprrox. Theory*, 25:318–336, 1979.

[31] M. Nina, D. Beglov, and B. Roux. Atomic radii for continuum electrostatics calculations based on molecular dynamics free energy simulations. *J. Phys. Chem. B*, 101:5239–5248, 1997.

[32] F. M. Richards. Areas, volumes, packing, and protein structure. *Annu. Rev. Biophys. Bioeng.*, 6:151–176, 1977.

[33] B. Roux and T. Simonson. Implicit solvent models. *Biophysical Chemistry*, 78:1–20, 1999.

[34] M. Schaefer and M. Karplus. A comprehensive analytical treatment of continuum electrostatics. *J. Phys. Chem.*, 100:1578–1599, 1996.

[35] J. Srinivasan, T. E. Cheatham, P. Cieplak, P. A. Kollman, and D. A. Case. Continuum solvent studies of the stability of dna, rna, and phosphoramidate-dna helices. *J. Am. Chem. Soc*, 120:9401–9409, 1998.

[36] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc*, 112:6127–6129, 1990.

[37] Y. Zhang, G. Xu, and C. Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design*, 23:510–530, 2006.