# An Algebraic Spline Model of Molecular Surfaces for Energetic Computations

Wenqi Zhao, Guoliang Xu, and Chandrajit Bajaj, *Member, IEEE*

W. Zhao and C. Bajaj are with the Institute for Computational Engineering and Science, University of Texas at Austin, (email: wzhao@ices.utexas.edu, bajaj@ices.utexas.edu).

G. Xu is with the Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, (email: xuguo@lsec.cc.ac.cn).

**Abstract**

In this paper, we describe a new method to generate a smooth algebraic spline (AS) 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 covering the PDB structure, and then generates a piecewise polynomial $F$ on the Bernstein-Bezier (BB) basis within the scaffold. An ASMS model of the molecular surface is extracted as the zero contours of $F$ which is nearly $C^1$ and has dual implicit and parametric representations. The dual representations allow us easily do the point sampling on the ASMS model and apply it to the accurate estimation of the integrals involved in the electrostatic solvation energy computations. Meanwhile comparing with the trivial piecewise linear surface model, fewer number of sampling points are needed for the ASMS, which effectively reduces the complexity of the energy estimation.

**Index Terms**

Polynomial splines, molecular surfaces, prismatic scaffolds, Bernstein-Bezier basis, solvation energetics, error bounds, rate of convergence.

## I. INTRODUCTION

The computation of electrostatic solvation energy (also known as polarization energy) for biomolecules plays an important role in the molecular dynamics simulation [1], the analysis of stability in protein structure prediction [2], and the protein-ligand binding energy calculation [3]. The explicit model of the solvent provides the most rigorous solvation energy calculation [4]. However, due to the large amount of solvent molecules, most of the computation time is spent on the trajectories of the solvent molecules, which severely increases the computation cost of this method [5]. An alternative method is to represent the solvent implicitly as a dielectric continuum [6], then the electrostatic potential is known by solving the Poisson-Boltzmann (PB) equations [7] [8]. A more efficient method is to approximate the PB electrostatic solvation energy by the generalized Born (GB) model [9] [10] [11], which computes the electrostatic solvation energy $\Delta G_{\text{elec}}$ 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 $\tau = \frac{1}{\varepsilon_p} - \frac{1}{\varepsilon_w}$, $\varepsilon_p$ is the solute (low) dielectric constant, $\varepsilon_w$ 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 (could be 4 [9] or 8 [11]), and $R_i$ is the effective Born radius of atom $i$. The effective Born radius reflects how deep an atom is buried in the molecule and consequently determines the importance to the polarization. The formulation of the effective Born radii is derived in [12]:

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

where $\Gamma$ is the molecular surface 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 details of the derivation of (2) and a fast evaluation algorithm based on the fast Fourier transform (FFT) for (2) is discussed in [13]. Since the numerical integrations are done on the molecular surface $\Gamma$, an accurate and analytic representation of $\Gamma$ is needed.

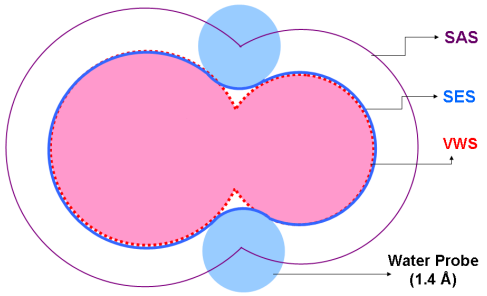


Fig. 1. Three molecular surfaces are shown for two atoms in two dimension. The boundary of the union of balls (dotted red line) with the van der Waals radii is the VWS. The SAS (solid thin line in purple) is the union of augmented van der Waals spheres with each radius enlarged by the radius of a solvent probe (light blue sphere). The SES (the solid thick line in blue) is boundary of all possible solvent probes that do not intersect with the interior of the VWS.

Three well-known molecular surfaces 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 augmented van der Waals spheres with each radius enlarged by the solvent probe radius (normally taken as 1.4 $\mathring{A}$) [14]. 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 intersect with the interior of the VWS [15] [16]. As described in [15], the SES consists of the convex spherical patches which are parts of the VWS as well, the toroidal patches and the concave spherical patches, which are generated by the probes rolling along the intersections of neighboring atoms. The VWS causes an overestimation of the electrostatic

solvation energy, while the SAS leads to an underestimation [11]. The SES is the most accurate when it is applied in the energetic calculation and therefore it is most often used to model the molecular surface. However the SES still has one significant drawback: it contains cusps when the rolling probe self-intersects, which may cause singularity in the Born radii and the force calculations.

In the energetic computation, knowing the patch complexes of the molecular surface is not enough. For convenience, an analytical representation of the molecular surface is needed and the singularity should also be avoided. One way to generate such a model is to define an analytical volumetric density function, for example, the summation of Gaussian functions [17], Fermi-Dirac switching function [18], or piecewise polynomials [11], and approximate the SES by an iso-contour of the density function. Techniques of fast extracting an iso-contour of smooth kernel functions are developed in [19] [20]. However the error of the generated isosurface could be large and result in inaccurate energy computation. A NURBS representation for the SES is presented in [21]. Although it provides a parametric approximation to the SES, it does not solve the singularity problem. Edelsbrunner [22] defines another paradigm of a smooth surface referred to as *skin* which is based on the Voronoi, Delaunay, and Alpha complexes of a finite set of weighed points. The *skin* model has good geometric properties such as it is free of singularity and it can be decomposed into a collection of quadratic patches. Triangulation schemes based on the *skin* model are provided in [23] [24]. However when applied to the energetic computation, the *skin* triangulation which in fact is a linear approximation to the SES has to be very dense to gain accuracy, which causes oversampling on the surface and hence makes the computation very slow. 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 to provide a method to model the SES as piecewise algebraic spline patches with certain continuity at the boundary of the patches. Each patch has dual implicit and parametric representations. Hence high order implicit surfaces can be parameterized onto a planer domain and therefore higher order quadrature rules of 2D such as the Gaussian quadrature rules can be easily applied to the energetic computation. Moreover, because higher order spline patches are used to approximate the SES, fewer number of triangles are needed to obtain the same accuracy in the energetic computation as the linear model. The algebraic spline patches are generated based on the prism scaffold built surrounding the

original triangular mesh of the SES and are defined implicitly by simple BB spline functions. Previous work on constructing piecewise spline patches within a simplical hull over a triangular mesh includes generating quadric patches [25], cubic patches [26] [27], and nonsingular and single sheeted cubic patches [28] in a tetrahedra scaffold. In this paper, we also show that the so generated algebraic spline patches are error bounded and free of singularity under certain conditions.

The paper is organized as follows: Section II describes the details of the algebraic spline molecular surface (ASMS) generation; Section III discusses the error of ASMS and Section IV discusses the application to the energetic computation and provides some examples.

## II. Algebraic spline model

### A. Algorithm Sketch

There are four main steps in our ASMS construction algorithm: (1) construct an initial triangular mesh of the SES; (2) build a prism scaffold surrounding the triangulation; (3) define a piecewise polynomial with certain continuity; (4) extract the 0-contour of the piecewise polynomial. We are going the explain each step in detail in the following and discuss how to make use the parametrization of the ASMS in the numerical integration.

### B. Initial triangulation of the MS

So far a lot of work has been done on the triangulation of the SES or its approximation [24] [29] [30] [31] [32]. The ASMS generation could be applied to any of these triangulations. In our current research we use the triangulation generated by a program in the software TexMol [32] [33] as the initial. In this program the SES is described as an iso-contour of a sign distance function (SDF) with the isovalue equal to the radius of the water probe. The SDF measures the distance of any point in $\mathbb{R}^3$ to the SAS where the sign indicates which side the point locates of the SAS. Here we define the SDF to be positive if the point is inside the SAS and negative if it is outside the SAS. A dual contour method is used to extract the iso-contour. The cusps created by the self-intersecting patches are detected and removed. Features of the molecular surface are well preserved in this triangulation. We then decimate the mesh by removing some of vertices from the triangulation. These vertices have the smallest normal variation, so the detailed features of the surface can still be captured after the vertices are removed [34].

### C. Implicit/parametric patches generation

Given the triangulation mesh $\mathcal{T}$, let $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$ be one of the triangles 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 $\mathbf{n}_l,\ (l = i, j, k)$. Let $\mathbf{v}_l(\lambda) = \mathbf{v}_l + \lambda\mathbf{n}_l$. First we define a prism (Figure 2) $D_{ijk} := \{\mathbf{p} :\ \mathbf{p} = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda),\ \lambda \in I_{ijk}\}$, 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 containing $0$ and for any $\lambda \in I_{ijk}$, $\mathbf{v}_i(\lambda)$, $\mathbf{v}_j(\lambda)$, $\mathbf{v}_k(\lambda)$ are not collinear and $\mathbf{n}_i$, $\mathbf{n}_j$, $\mathbf{n}_k$ point to the same side of the plane $P_{ijk}(\lambda) := \{\mathbf{p} :\ \mathbf{p} = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda)\}$. Next we define a function in the

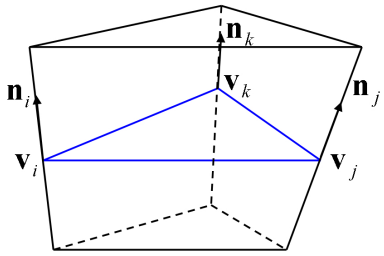


Fig. 2. A prism $D_{ijk}$ constructed based on the triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$.

Bernstein-Bezier (BB) basis over the prism $D_{ijk}$:

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

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

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

We approximate the molecular surface by the zero contour of $F$, denoted as $S$. In order to make $S$ smooth, the degree of the Bezier basis $n$ should be no less than 3. For simplicity, here we consider the case of $n = 3$. The control coefficients $b_{ijk}(\lambda)$ should be properly defined such that $S$ is continuous. In Figure 3 we show the relationship of the control coefficients and the points of the triangle when $n = 3$. Next we are going to discuss these coefficients are defined.

Since $S$ passes through the vertices $\mathbf{v}_i,\ \mathbf{v}_j,\ \mathbf{v}_k$, we define

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

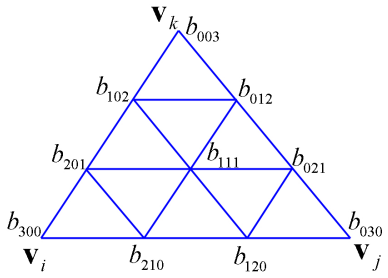


Fig. 3. The control coefficients of the cubic Bezier basis of function $F$.

Next we are going to define the coefficients on the edges of the triangle in Figure 3. To obtain $C^1$ continuity at $\mathbf{v}_i$, we require that the directional derivatives of $F$ at $\mathbf{v}_i$ in the direction of $b_2$ and $b_3$ are equal to $\nabla F \cdot (\mathbf{v}_j - \mathbf{v}_i)$ and $\nabla F \cdot (\mathbf{v}_k - \mathbf{v}_i)$, respectively. Noticing that F has the form of (3) and $(b_1, b_2, b_3) = (1, 0, 0)$ at $\mathbf{v}_i$, one can derive that $b_{210} - b_{300} = \frac{1}{3}\nabla F(v_i) \cdot (\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda))$, 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.

To obtain the $C^1$ continuity at the midpoints of the edges of $\mathcal{T}$, we define $b_{111}$ by using the side-vertex scheme [35]:

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

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 define $b_{111}^{(1)}$, $b_{111}^{(2)}$ and $b_{111}^{(3)}$. In Appendix V-A we prove that our scheme of defining this three coefficients can guarantee the $C^1$ continuity at the midpoints of the edges $\mathbf{v}_j\mathbf{v}_k$, $\mathbf{v}_i\mathbf{v}_k$ and $\mathbf{v}_i\mathbf{v}_j$. Consider the edge $\mathbf{v}_i\mathbf{v}_j$. Recall that any point $\mathbf{p} = (x, y, z)$ in $D_{ijk}$ can be represented by

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

Therefore differentiating both sides of (7) with respect to $x$, $y$ and $z$, respectively, yields

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

where $I_3$ is a $3 \times 3$ unit matrix. Denote

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

and 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 $M = (A\ B\ C)^{\mathrm{T}}$. From (8) we have

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

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

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

By (6), at $(b_1, b_2, b_3) = (\frac{1}{2}, \frac{1}{2}, 0)$ we have $b_{111} = b^{(3)}_{111}$. Therefore the gradient at $(\frac{1}{2}, \frac{1}{2}, 0)$ is

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

Define vectors

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

Let

$$\mathbf{c} = C(\frac{1}{2}, \frac{1}{2}, 0), \tag{13}$$

$$\mathbf{d}_3(\lambda) = \mathbf{d}_3(\frac{1}{2}, \frac{1}{2}, 0, \lambda) = B \times \mathbf{c} + \mathbf{c} \times A. \tag{14}$$

Let $\nabla F = \nabla F(\frac{1}{2}, \frac{1}{2}, 0)$. In order to have $C^1$ continuity at $(\frac{1}{2}, \frac{1}{2}, 0)$, we should have $\nabla F \cdot \mathbf{d}_3(\lambda) = 0$. Therefore, by (11) and (14), we have

$$b_{111}^{(3)} = \frac{\mathbf{d}_3(\lambda)^{\mathrm{T}}(3b_{210}B \times \mathbf{c} + 3b_{120}\mathbf{c} \times A + A \times B)}{3\|\mathbf{d}_3(\lambda)\|^2}. \tag{15}$$

Similarly, we may define $b_{111}^{(1)}$ and $b_{111}^{(2)}$.

Now the function $F(b_1, b_2, b_3, \lambda)$ is well defined. The next step is to extract the zero level set $S$. Given the barycentric coordinates $(b_1, b_2, b_3)$ of a point in the triangle $[\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k]$, we find the corresponding $\lambda$ by solving the equation $F(b_1, b_2, b_3, \lambda) = 0$ for $\lambda$ and this could be done by the Newton's method. Then we may get the corresponding point on $S$ as

$$(x,\ y,\ z)^{\mathrm{T}} = b_1\mathbf{v}_i(\lambda) + b_2\mathbf{v}_j(\lambda) + b_3\mathbf{v}_k(\lambda). \tag{16}$$

### *D. Smoothness*

*Theorem 2.1:* The ASMS $S$ is $C^1$ at the vertices of $\mathcal{T}$ and the midpoints of the edges of $\mathcal{T}$.

*Theorem 2.2:* $S$ is $C^1$ everywhere if every edge $\mathbf{v}_i\mathbf{v}_j$ of $\mathcal{T}$ satisfies
$\mathbf{n}_i \cdot (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{n}_j \cdot (\mathbf{v}_j - \mathbf{v}_i)$.

*Theorem 2.3:* $S$ is $C^1$ everywhere if the unit normals at the vertices of $\mathcal{T}$ are the same.

Proofs of the theorems are shown in the Appendix.

### *E. Parametrization and quadrature*

In this section, we would like to show how the ASMS is applied to the computation of (2). Since we use the ASMS to represent the molecular surface, now $\Gamma = S$. Let $f = \frac{(\mathbf{r}-\mathbf{x}_i)\cdot\mathbf{n}(\mathbf{r})}{|\mathbf{r}-\mathbf{x}_i|^4}$. We decompose the entire surface $S$ into patches $\{S_j\}$ with $S_j$ being the AMSM generated over triangle $j$, then we have

$$\int_S f(\mathbf{x})\ dS = \sum_j \int_{S_j} f(\mathbf{x})\ dS. \tag{17}$$

For any point $\mathbf{x} = (x, y, z)$ on $S_j$, by the inverse map of (16), one can uniquely map $\mathbf{x}$ to a point in triangle $j$ and get its baricentric coordinates $(b_1, b_2, b_3)$ with $b_3 = 1 - b_1 - b_2$. Therefore, $x$, $y$, $z$ can be represented in terms of $(b_1, b_2)$:

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

Replacing $(x, y, z)$ with $(b_1, b_1, b_3)$ in (17) and letting

$$g(b_1, b_2) = f(x(b_1, b_2), y(b_1, b_2), z(b_1, b_2)),$$

we get

$$\int_{S_j} f(\mathbf{x})\ dS = \int_{\sigma_j} g(b_1, b_2)\sqrt{EG - F^2}\ db_1 db_2, \tag{18}$$

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 then apply the Gaussian quadrature to (18):

$$\int_{\sigma_i} g(b_1, b_2)\sqrt{EG - F^2}\ db_1 db_2 \approx \sum_{k=1}^{n} W_k g(b_1^k, b_2^k)\sqrt{EG - F^2}|_{b_1^k, b_2^k}, \tag{19}$$

where $(b_1^k, b_2^k, b_3^k)$ and $W_k$ are the Gaussian integration nodes and weights on the triangles.

## III. Error of the ASMS model

In order to show the error of $S$ to the true surface $S_0$, we do a test on some typical surfaces (Table I) $S_0 := \{(x, y, z) : z = f(x, y), (x, y) \in [0, 1]^2\}$ which are considered as the true surfaces. We generate a triangulation mesh over the true surface with the maximum edge length $h$ being $0.1$. Based on the mesh, we construct the ASMS model $S$. The error of $S$ to $S_0$ is defined as $\max \frac{\|\mathbf{p}-\mathbf{q}\|}{\|\mathbf{q}\|}$, where $\mathbf{p} \in S$, $\mathbf{q} \in S_0$, and $\mathbf{p}$ and $\mathbf{q}$ have the same $(b_1, b_2, b_3)$ coordinates but different $\lambda$. We sample $(\mathbf{p},\ \mathbf{q})$ on the surfaces and compute the maximum relative error. For the point pair $\mathbf{p}(b_1, b_2, b_3, \lambda_p)$ and $\mathbf{q}(b_1, b_2, b_3, \lambda_q)$ defined above, we prove that their Euclidean distance is bounded by the difference of their $\lambda$ coordinates.

*Lemma 3.1:* The error of the approximation point $\mathbf{p}$ to the true point $\mathbf{q}$ is bounded by $|\lambda_p - \lambda_q|$.

TABLE I

RELATIVE ERROR AND CONVERGENCE

| Function $(x, y) \in [0, 1]^2$ | $\max\{\frac{\|\mathbf{p}-\mathbf{q}\|}{\|\mathbf{q}\|}\}$ | $C$ |
|---|---|---|
| $z = 0$ | 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 |

*Proof:*

$$\begin{aligned}\|\mathbf{p} - \mathbf{q}\| &\leq b_1\|\mathbf{v}_i(\lambda_p) - \mathbf{v}_i(\lambda_q)\| + b_2\|\mathbf{v}_j(\lambda_p) - \mathbf{v}_j(\lambda_q)\| + b_3\|\mathbf{v}_k(\lambda_p) - \mathbf{v}_k(\lambda_q)\| \\ &\leq |\lambda_p - \lambda_q|(b_1\|\mathbf{n}_i\| + b_2\|\mathbf{n}_j\| + b_3\|\mathbf{n}_k\|) \\ &= |\lambda_p - \lambda_q|\end{aligned}$$

■

To study the rate of converges of $S$ to $S_0$, we gradually refine the initial mesh. Since the error is bounded by $|\lambda_p - \lambda_q|$, we compute the ratio of the maximum difference of $\lambda_p$ and $\lambda_q$ to $h$, $h^2$, $h^3$, and so forth. As $h$ decreases, we check if the ratio converges or not, which allows us to know the highest rate of convergence of $S$ to $S_0$. For most of the test functions in Table I, we observe that $S$ converges to $S_0$ as fast as $O(h^3)$. We also observe that for the case $z = \sqrt{1 - x^2 - y^2}$, the rate of convergence reaches $O(h^4)$. We show the limit of the ratio $\frac{|\lambda-\lambda'|}{h^3}$ as $h \downarrow 0$, denoted as $C$, in Table I. Hence we draw the following claim:

**Claim:** Let $h$ be the maximum side length of triangulation mesh $\mathcal{T}$, $\mathbf{p}$ be the point on the ASMS, $\mathbf{q}$ be the corresponding point on the true surface, then $\mathbf{p}$ converges to $\mathbf{q}$ at the rate of $O(h^3)$. i.e. There exists a constant $C$ such that $\|\mathbf{p} - \mathbf{q}\| \leq Ch^3$.

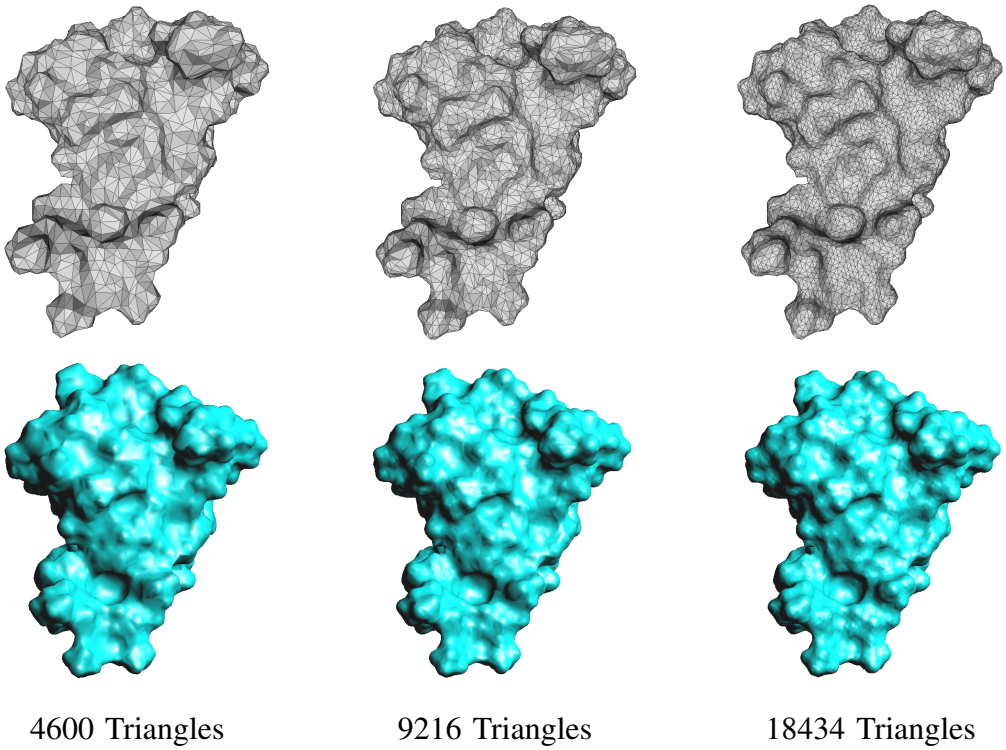


Fig. 4. The top row is the triangulation of the SES of protein 1ML0 with different number of triangles. The bottom row is the ASMS generated from the above corresponding triangulation.

We generated the ASMS for the real proteins based on different size of meshes (Figure 4) and show the error of the ASMS to the SES of three proteins: 1GCQ (843 atoms), 1ML0 (1051 atoms), and 1KKL (1276 atoms) in Table II. Here the SES is modeled as a level set of the summation of fast decaying Gaussian functions. The ASMS is generated from the triangulation of the SES at different resolution. The number of triangles of the initial meshes are listed in Table II. The error $\varepsilon_{max}$ is defined as the one-way Hausdorff distance from the ASMS to the SES: $\varepsilon_{max} = \max_{\mathbf{p}\in\text{ASMS}} \min_{\mathbf{q}\in\text{SES}} \|\mathbf{p}-\mathbf{q}\|$. As we see in the table, the errors are small and decrease rapidly as the initial triangulation becomes dense.

## IV. Application to the Biomolecular Energetic Computation

We apply the ASMS model to the GB electrostatic solvation energy computations of the example proteins 1PPE (436 atoms), 1HIA (693 atoms), 1CGI (852 atoms), 7CEI (1912 atoms), 1F15 (7704 atoms), and 1KXP (11859 atoms). The ASMS models $S$ for the proteins are generated

TABLE II

ERROR OF ASMS TO THE SES

| 1GCQ | | 1ML0 | | 1KKL | |
|---|---|---|---|---|---|
| No. of $\Delta$s | $\varepsilon_{max}$ | No. of $\Delta$s | $\varepsilon_{max}$ | No. of $\Delta$s | $\varepsilon_{max}$ |
| 16,312 | 0.266069 | 18,400 | 0.233949 | 19,968 | 0.260418 |
| 32,624 | 0.142149 | 36,864 | 0.142380 | 39,544 | 0.134689 |
| 65,456 | 0.082550 | 73,736 | 0.083895 | 79,096 | 0.085855 |

based on the initial mesh with different number of triangles (Table III). We show the ASMS of the example molecules generated from the decimated triangulations in Figure 5 and Figure 6. As a comparison, we compute the polarization energy $G_{pol}$ for both the ASMS and the piecewise linear (PL) surfaces and show the energy results and the timing in Table III. For all the computations, a 4-point Gaussian quadrature rule over a triangle [36] is used for the numerical integration in (19) when computing the Born radii. The running time contains the time cost of computing the integration nodes over the surfaces, computing the Born radii, and evaluating $G_{pol}$. If we consider the energy computed from the dense mesh as accurate, as we see from the table, the $G_{pol}$ computed from the coarse PL model has a large error, however for the coarse ASMS model, it is very close to the dense mesh result but with less time. On the other hand, to get a energy result of the same accuracy, fewer number of triangles are needed for the ASMS model than the PL model. For example, for the protein 1CGI, the $G_{pol}$ computed from the ASMS with 3674 triangles is -1394.227 kcal/mol. However to get a similar result, 8712 triangles are needed for the piecewise linear model. Therefore the ASMS model is much more efficient in the energetic computation than trivial piecewise linear models.

## V. CONCLUSIONS

We have introduced a method to generate a model for the molecular surface. Like the other molecular surface models, this ASMS model is smooth and close to the SES as long as the initial triangulation is based on the SES. In addition, it has dual implicit and parametric representations. The implicit representation enables us to flexibly vary the surface by selecting different level sets, while the parametric representation allows us easily apply the ASMS to the numerical

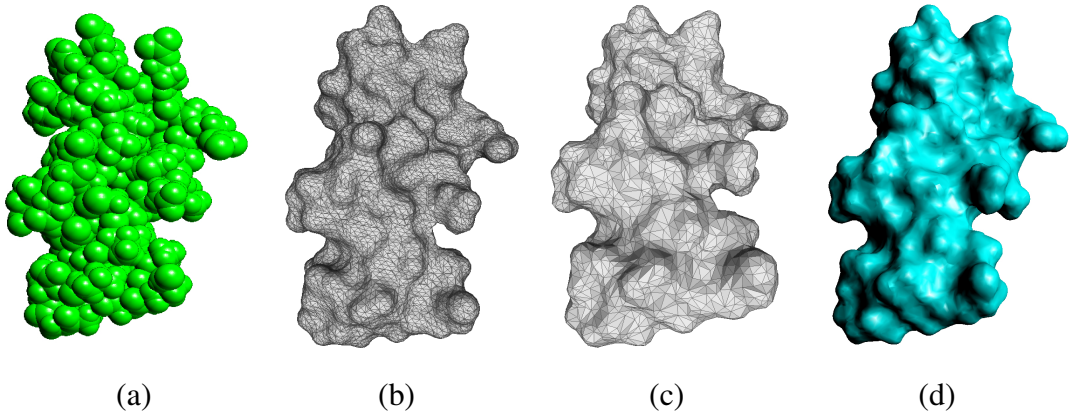


Fig. 5. Molecular models of a protein(1HIA). (a) is The atomic model. (b) is the initial dense mesh of the SES (27480 triangles). (c) is the decimated mesh of the SES model (7770 triangles). (d) is the ASMS (7770 patches) generated from (c).

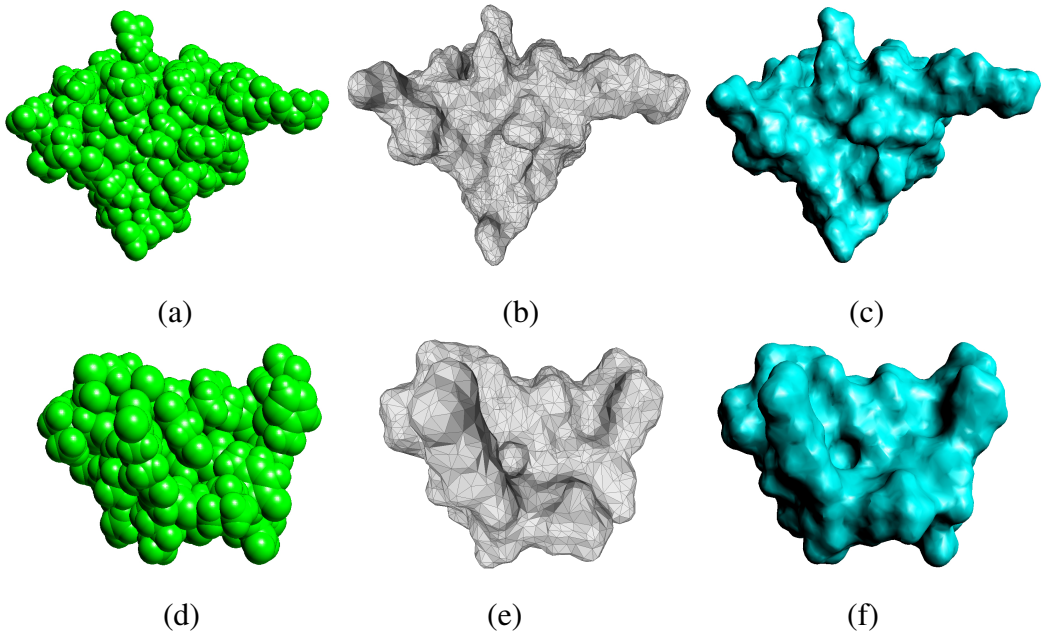


Fig. 6. The top row are the models of 1CGI and the bottom row are the models of 1PPE. (a) and (d) are the atomic structures of the proteins. (b) and (e) are the decimated triangular meshes of the proteins with 8712 triangles and 6004 triangles, respectively. (c) and (f) are the ASMS models generated from (b) and (e), respectively.

TABLE III

ELECTROSTATIC SOLVATION ENERGY AND TIMING

| Protein ID | No. of Triangles | $G_{pol}$ (kcal/mol) | | Timing (s) | |
|---|---|---|---|---|---|
| | | PL | AS | PL | AS |
| 1PPE | 24244 | -835.5639 | -825.3252 | 17.27 | 18.26 |
| | 6004 | -852.7130 | -828.2158 | 5.09 | 5.39 |
| | 2748 | -933.9562 | -845.5085 | 2.74 | 3.27 |
| 1HIA | 27480 | -1361.2266 | -1340.6384 | 30.23 | 31.18 |
| | 7770 | -1389.0175 | -1347.8067 | 9.43 | 9.93 |
| | 3510 | -1571.8908 | -1388.4665 | 5.21 | 5.21 |
| 1CGI | 29108 | -1371.7419 | -1343.1496 | 39.64 | 40.31 |
| | 8712 | -1399.1948 | -1346.2230 | 12.94 | 12.64 |
| | 3674 | -1678.4447 | -1394.2270 | 7.40 | 6.11 |
| 7CEI | 54544 | -3758.7928 | -3711.3626 | 29.04 | 29.96 |
| | 17044 | -3771.7803 | -3753.3377 | 10.03 | 9.89 |
| | 5324 | -3876.7333 | -3826.1959 | 4.11 | 4.08 |
| 1F51 | 87516 | -11656.0327 | -11411.9689 | 123.55 | 121.52 |
| | 33660 | -11691.4450 | -11622.8886 | 51.95 | 52.56 |
| | 8290 | -12527.8362 | -11721.4931 | 21.01 | 21.55 |
| 1KXP | 402812 | -13258.0206 | -13121.3053 | 975.08 | 977.96 |
| | 134272 | -13325.0423 | -13264.7272 | 340.16 | 346.31 |
| | 94352 | -14669.1209 | -14071.9965 | 246.68 | 244.33 |

computations, such as the numerical integrations involved in the finite element method or the boundary element method. Moreover, unlike the other piecewise linear models, the ASMS surface is of higher degree, therefore, to get the same accuracy, fewer number of triangles (roughly one-third of the PL model) are needed for the ASMS when it is applied to the numerical integrations. For many large system problems, for example the atomistic molecular dynamics simulations, efficient computation is the most concerning issue, hence he ASMS is very suitable to be used in this kind of problems. We should mention that, while not detailed in this paper, the algorithm of Section II-C can, by repeated evocation, yield a hierarchical multiresolution spline model of the molecular surface. In the future research we could extend this algebraic patch model to the electrostatic solvation forces calculation which is crucial in the molecular dynamics simulations.

Fast and accurate numerical integration is also one of the main tasks of the force calculation and is more challenging because the integration domain contains not only the surface but also a skin layer over each atom.

## APPENDIX

### A. *Proof of Theorem 2.1*

*Proof:* 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 $M$ can be written as

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

and

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

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 $\mathbf{v}_i\mathbf{v}_j$,

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

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

$$\begin{aligned} \nabla F =& \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + M^{-1} \begin{pmatrix} 3(b_{210} - b_{111}) \\ 3(b_{120} - b_{111}) \\ 1 \end{pmatrix} 2t(1-t) \\ =& \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + \frac{2t(1-t)}{\det(M)(t)} \big[3(B \times C(t))(b_{210} - b_{111}) \\ & + 3(C(t) \times A)(b_{120} - b_{111}) + A \times B\big]. \end{aligned} \tag{20}$$

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

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

Consider the function inside the square bracket of (20) and denote it as $F_1$. Then

$$F_1 = 3(B \times \mathbf{c})b_{210} + 3(\mathbf{c} \times A)(b_{120} + A \times B - 3(B \times \mathbf{c} + \mathbf{c} \times A)b_{111}. \tag{21}$$

Since on the edge $\mathbf{v}_i\mathbf{v}_j$, $b_{111} = b_{111}^{(3)}$, substituting (15) into (21), we get $F_1$ is 0. Therefore, at the midpoint

$$\nabla F = (\mathbf{n}_i + \mathbf{n}_j)/4. \tag{22}$$

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

### *B. Proof of Theorem 2.2*

*Proof:* It is obvious that $S$ is $C^1$ within the triangles. By Theorem 2.1 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 (20) is written as

$$\nabla F = \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 + \frac{2t(1-t)}{\det(M(t))}[3(b_{210} - b_{111})(B - A) \times C + A \times B]. \tag{23}$$

Similar as (12), we define

$$\mathbf{d}_3(t, \lambda) = (B - A) \times C(t). \tag{24}$$

By (15) together with the facts that $b_{120} = b_{210}$ and $b_{111} = b_{111}^{(3)}$ on edge $\mathbf{v}_i\mathbf{v}_j$, we have

$$b_{210} - b_{111} = -\frac{\mathbf{d}_3^{\mathrm{T}}(\lambda)(A \times B)}{3\|\mathbf{d}_3(\lambda)\|^2}, \tag{25}$$

where $\mathbf{d}_3(\lambda)$ is defined in (14). Plug (24) and (25) in (23), we get

$$\begin{aligned} \nabla F = {} & \mathbf{n}_i(1-t)^2 + \mathbf{n}_j t^2 \\ & + \frac{2t(1-t)}{\|\mathbf{d}_3(\lambda)\|^2}\left[\frac{\|\mathbf{d}_3(\lambda)\|^2 A \times B - \mathbf{d}_3(t,\lambda)\mathbf{d}_3^{\mathrm{T}}(\lambda)A \times B}{\det(M(t))}\right]. \end{aligned} \tag{26}$$

Consider the function inside the square bracket of (26) and denote it as $F_2$. Our goal is to show that $F_2 = 0$. Since we have already known that when $t = \frac{1}{2}$, $F_2 = 0$, this 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

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

In order to show $\frac{\partial F_2}{\partial t} = 0$, which is equivalent to show $N := ad - bc = 0$, we compute

$$\begin{aligned} N =& [\mathbf{n}_j \times (B - A)\mathbf{d}_3^{\mathrm{T}} A \times B]\mathbf{n}_i^{\mathrm{T}}(A \times B) \\ & -(\|\mathbf{d}_3(\lambda)\|^2 A \times B)(\mathbf{n}_j - \mathbf{n}_i) \times (B - A) \\ & -[\mathbf{n}_i \times (B - A)\mathbf{d}_3^{\mathrm{T}} A \times B]\mathbf{n}_j^{\mathrm{T}}(A \times B). \end{aligned} \tag{27}$$

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 $(B-A)^{\mathrm{T}}\mathbf{c} = (\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda))^{\mathrm{T}}\mathbf{c} = 0$, where $\mathbf{c} = C(\frac{1}{2}, \frac{1}{2}, 0)$. Therefore

$$\|\mathbf{d}_3(\lambda)\|^2 = ((B - A) \times \mathbf{c}) \cdot ((B - A) \times \mathbf{c}) = \|\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)\|^2 \|\mathbf{c}\|^2, \tag{28}$$

and

$$\begin{aligned} \mathbf{d}_3^{\mathrm{T}}(\lambda)A \times B &= \mathbf{d}_3^{\mathrm{T}}(\lambda)A \times (B - A) \\ &= ((B - A) \times \mathbf{c}) \cdot (A \times (B - A)) = -\mathbf{c}^{\mathrm{T}} A \|\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)\|^2. \end{aligned} \tag{29}$$

Plug (28) and (29) into (27) and divide both sides by $\|\mathbf{v}_j(\lambda) - \mathbf{v}_i(\lambda)\|^2$, we get

$$\begin{aligned} F_3 :=& \frac{N}{\|(\mathbf{v}_j - \mathbf{v}_i)(\lambda)\|^2} \\ =& -\mathbf{n}_j \times (B - A)\mathbf{c}^{\mathrm{T}} A \mathbf{n}_i^{\mathrm{T}}(A \times B) - \|\mathbf{c}\|^2 A \times B(\mathbf{n}_j - \mathbf{n}_i)^{\mathrm{T}} A \times B \\ & + (\mathbf{n}_i \times (B - A)\mathbf{c}^{\mathrm{T}} A)\mathbf{n}_j^{\mathrm{T}}(A \times B) \\ =& [(\mathbf{c}^{\mathrm{T}} A \mathbf{n}_i - \|\mathbf{c}\|^2 A) \times (B - A)]\mathbf{n}_j^{\mathrm{T}}(A \times B) \\ & + [(\|\mathbf{c}\|^2 A - \mathbf{c}^{\mathrm{T}} A \mathbf{n}_j) \times (B - A)]\mathbf{n}_i^{\mathrm{T}}(A \times B). \end{aligned} \tag{30}$$

If $\mathbf{n}_i = \mathbf{n}_j$, (30) is 0. Now let us assume $\mathbf{n}_i \neq \mathbf{n}_j$. Recall that $\mathbf{c} = \frac{1}{2}(\mathbf{n}_i + \mathbf{n}_j)$. we 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 $D$:

$$\mathbf{c}^{\mathrm{T}}\mathbf{e} = 0, \qquad \mathbf{c}^{\mathrm{T}} D = 0. \tag{31}$$

Furthermore

$$\mathbf{c} \times (D \times \mathbf{e}) = 0. \tag{32}$$

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

$$\mathbf{n}_i = \mathbf{c} + \mathbf{e}, \qquad \mathbf{n}_j = \mathbf{c} - \mathbf{e}. \tag{33}$$

Substitute (33) into (30) and replace $A \times B$ with $A \times D$, we get

$$\begin{aligned} F_3 =& [\mathbf{c}^{\mathrm{T}} A(\mathbf{c}+\mathbf{e}) - \|\mathbf{c}\|^2 A] \times D(\mathbf{c}-\mathbf{e})^{\mathrm{T}}(A \times D) \\ &+ [\|\mathbf{c}\|^2 A - \mathbf{c}^{\mathrm{T}} A(\mathbf{c}-\mathbf{e})] \times D(\mathbf{c}+\mathbf{e})^{\mathrm{T}}(A \times D) \\ =& 2\mathbf{c}^{\mathrm{T}} A(\mathbf{e} \times D)\mathbf{c}^{\mathrm{T}}(A \times D) \\ &- 2[\mathbf{c}^{\mathrm{T}} A\mathbf{c} - \|\mathbf{c}\|^2 A] \times D\mathbf{e}^{\mathrm{T}}(A \times D). \end{aligned} \tag{34}$$

If $\mathbf{e}$ and $D$ are linearly dependent, then $\mathbf{e} \times D = 0$, moreover $\mathbf{e}^(A \times D) = 0$, which yields $F_3 = 0$. Otherwise, we introduce a new matrix

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

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

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

By the Lagrange's formula:

$$\mathbf{c}^{\mathrm{T}}A\mathbf{c}^{\mathrm{T}}(D\times\mathbf{e})-\|\mathbf{c}\|^2A^{\mathrm{T}}(D\times\mathbf{e})=(\mathbf{c}\times A)\cdot(\mathbf{c}\times(D\times\mathbf{e})), \tag{35}$$

and (32), (35) is zero and thus $F_3=0$. So far we have proved that $F_2$ is independent of $t$. Meanwhile in the proof of Theorem 2.1, we know that $F_2=0$ at $t=\frac{1}{2}$. Hence $F_2=0$ for all $t$ and therefore on the edge $\mathbf{v}_i\mathbf{v}_j$, $\nabla F$ is

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

So $S$ is $C^1$ on the edges. ■

### C. Proof of Theorem 2.3

*Proof:* As same as the proof of Theorem 2.2, we only need to show that $S$ is $C^1$ on the edge $\mathbf{v}_i\mathbf{v}_j$. In the proof of Theorem 2.1, we have already derived the gradient function on the edge $\mathbf{v}_i\mathbf{v}_j$ (20):

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

Let

$$F_4=\frac{1}{\det(M)(t)}[3(B\times C(t))(b_{210}-b_{111})+3(C(t)\times A)(b_{120}-b_{111})+A\times B]. \tag{36}$$

Following the same idea of the proof the Theorem 2.2, we compute $\frac{\partial F_4}{\partial t}$. The numerator of $\frac{\partial F_4}{\partial t}$ is

$$\begin{aligned}&[3(B\times C'(t))(b_{210}-b_{111})+3(C'(t)\times A)(b_{120}-b_{111})\\&+A\times B]\det(M)-\det(M)'(t)[3(B\times C(t))(b_{210}-b_{111})\\&+3(C(t)\times A)(b_{120}-b_{111})+A\times B].\end{aligned} \tag{37}$$

Since

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

(37) is 0 when $\mathbf{n}_i=\mathbf{n}_j$. So $F_4$ is independent of $t$. By the proof of Theorem 2.1, $F_4=0$ at $t=\frac{1}{2}$. So $F_4=0$ for all $t$. So $S$ is $C^1$ continuous. ■

## ACKNOWLEDGMENT

This research was supported in part by NSF grant CNS-0540033 and NIH contracts R01-EB00487, R01-GM074258, R01-GM07308. We wish to thank Vinay Siddavanahalli and other members of the CVC group for developing and maintaining TexMol, our molecular modeling and visualization software tool (http://cvcweb.ices.utexas.edu/software/). A substantial part of this work in this paper was done when Guoliang Xu was visiting Chandrajit Bajaj at UT-CVC. His visit was additionally supported by the J. T. Oden ICES visitor fellowship.

## REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] C. Bajaj, V. Siddavanahalli, and W. Zhao. Fast algorithms for molecular interface triangulation and solvation energy computations. *ICES Technical Report TR-07-06*, 2007.

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

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

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

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

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

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

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

[21] 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.

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

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

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

[25] 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.

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

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

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

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

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

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

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

[33] 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.

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

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

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