

# Quality Meshing of Implicit Solvation Models of Biomolecular Structures<sup>\*</sup>

Yongjie Zhang<sup>†</sup>   Guoliang Xu<sup>§</sup>   Chandrajit Bajaj<sup>†</sup>

<sup>†</sup>*Computational Visualization Center, Department of Computer Sciences, Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA*

<sup>§</sup>*State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China*

---

## Abstract

This paper describes a comprehensive approach to construct quality meshes for implicit solvation models of biomolecular structures starting from atomic resolution data in the Protein Data Bank (PDB). First, a smooth volumetric electron density map is constructed from atomic data using weighted Gaussian isotropic kernel functions and a two-level clustering technique. This enables the selection of a smooth implicit solvation surface approximation to the Lee-Richards molecular surface. Next, a modified dual contouring method is used to extract triangular meshes for the surface, and tetrahedral meshes for the volume inside or outside the molecule within a bounding sphere/box of influence. Finally, geometric flow techniques are used to improve the surface and volume mesh quality. Several examples are presented, including generated meshes for biomolecules that have been successfully used in finite element simulations involving solvation energetics and rate binding constants.

*Key words:* quality mesh, biomolecule, implicit solvation model, finite element simulation.

---

## 1 Introduction

Finite element simulations have become an important tool in the analysis of biomolecular functional models, such as electrophoresis, electrostatics and diffusion influenced reaction rate constants [36] [37] [41]. For efficient and accurate finite element solutions, adaptive and quality meshes are a necessary first step. Quite often, people have to give up FEM because they can not generate satisfied triangular or tetrahedral meshes to represent the geometric model for large complicated biomolecules such as Ribosome (Fig. 1), or those structures whose active site occurs at the bottom of a narrow gorge (deep pocket) (Fig.13).

---

<sup>\*</sup> Visit <http://www.ices.utexas.edu/cvc/meshing/MolMesh>

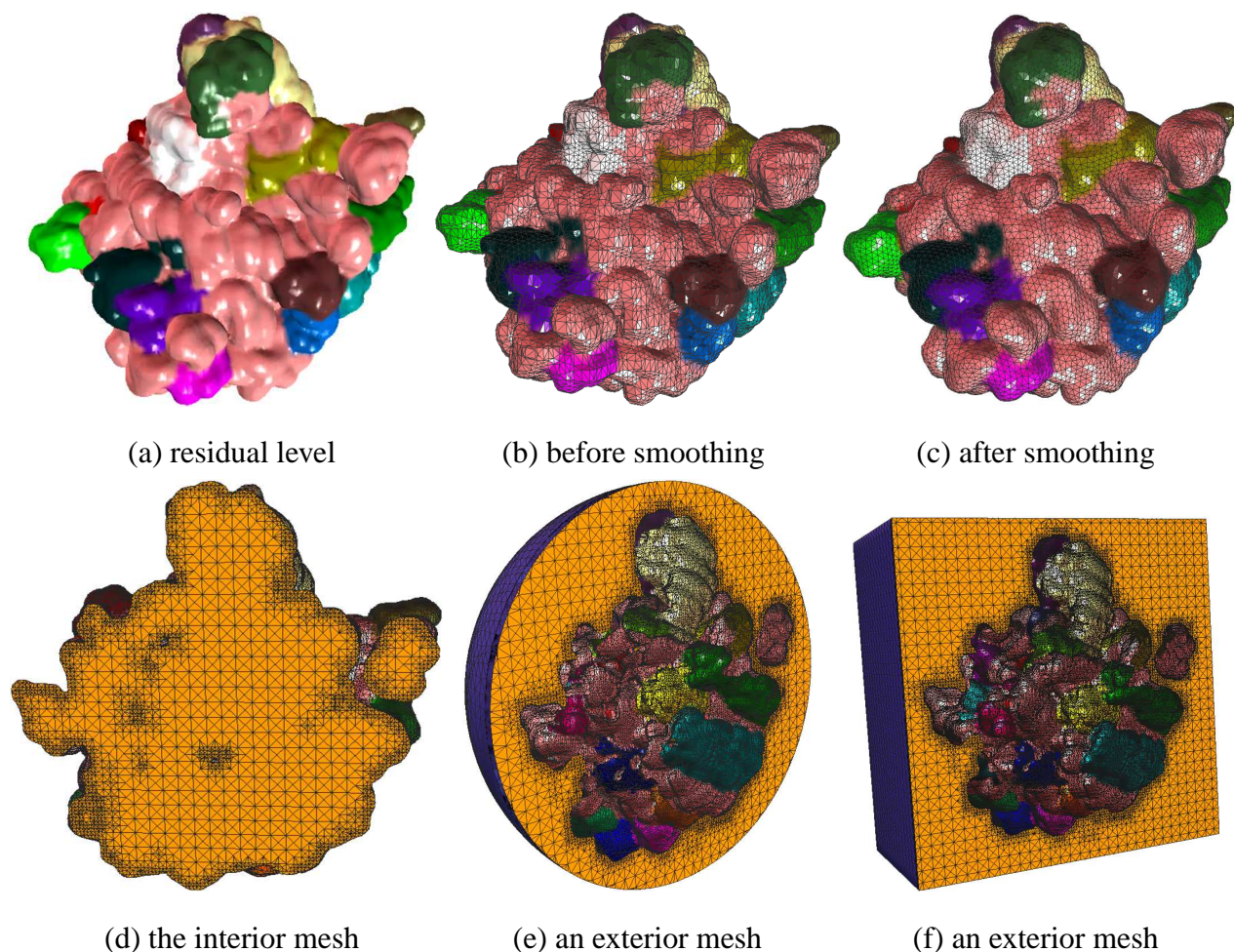


Fig. 1. Implicit solvation models of *Haloarcula Marismortui* large Ribosome 50S (1JJ2) crystal subunit. The light yellow and the pink color show 5S and 23S rRNA respectively, the remaining colors are proteins. (a): the implicit solvation model at the medium resolution level,  $p_1 = 0.0625$ ,  $p_2 = 1.0$ ; (b) and (c): triangular meshes (16700 vertices, 33400 triangles); (d): the interior mesh (230025 vertices, 1141575 tets); (e): an exterior mesh within a sphere (234902 vertices, 1162568 tets); (f): an exterior mesh within a bounding box (260858 vertices, 1315112 tets).

The protein data bank (<http://www.rcsb.org/pdb>) provides PDB format files for protein and RNA structures, with the location of principally all the major atoms (e.g., hydrogen atoms are not discernible via X-ray diffraction and therefore rarely present in the PDB). The summation of kernel functions centered at each atom can be used to construct a smooth volumetric electron density map from PDB data [4] [17]. The volumetric data is often sampled at each rectilinear grid point,  $V = \{F(i, j, k) | i, j, k \text{ are indices of } x, y, z \text{ coordinates in a rectilinear grid}\}$ , and the implicit solvation surface is approximated as a level set  $S_F(c) = \{(x, y, z) | F(i, j, k) = c\}$ , where  $c$  is a constant [17] [23]. The computation of density maps can be made very efficient with worst case complexity linear in the number of grid points and the number of atoms [2]. In this paper, we describe an approach to generate quality triangular/tetrahedral meshes for complex biomolecular structures from PDB format data, conforming to good implicit solvation surface approximations. There are three main steps in our mesh generation process:

- (1) Implicit Solvation Surface – A good approximation of the implicit solvation surface is generated from a smooth volumetric synthetic electron density map by a careful choice of the parameter of Gaussian kernel functions.
- (2) Mesh Generation – The modified dual contouring method is used to generate triangular and interior/exterior tetrahedral meshes.
- (3) Quality Improvement – Geometric flows are used to improve the quality of extracted triangular and tetrahedral meshes.

The summation of Gaussian kernel functions is used to construct the density map of a biomolecule and sampled volumetric data. A smooth implicit solvation model can be constructed to approximate the Lee-Richards molecular surface by using weighted Gaussian isotropic kernel functions and a two-level clustering techniques.

The dual contouring method [19] [42] [43] is selected for mesh generation as it tends to yield meshes with better aspect ratio. In order to generate exterior meshes, we add a sphere or box outside the biomolecular surface as an outer boundary. A variant of the dual contouring method is developed to extract interior and exterior meshes. Our tetrahedral mesh is spatially adaptive and attempts to preserve molecular surface features while minimizing the number of elements. An extension step is performed to generate the exterior mesh.

The extracted triangular and tetrahedral meshes cannot be directly used for finite element calculations, they need to be modified and improved. Since the isosurface generated from discrete volumetric data suffers from noise, geometric flows are used to smooth the generated surface meshes with feature preserved. The quality of extracted surface and volume meshes is also improved.

The main contributions of this paper include: a simple and uniform treatment for approximating implicit solvation models, a modified adaptive surface and volume mesh extraction scheme combined with geometric flows and therefore yields high quality meshes. The generated meshes of the monomeric and tetrameric mouse acetylcholinesterase (mAChE) have been successfully used in solving the steady-state Smoluchowski equation using finite element method [36] [37] [41].

The remainder of this paper is organized as the following: Section 2 reviews related previous work. Section 3 introduces how to construct implicit solvation surface from PDB format data. Section 4 explains mesh generation schemes. Section 5 discusses mesh quality improvement. Section 6 presents results and conclusion.

## 2 Previous Work

**Molecular Surface Approximation:** There are three different approximations of molecular surfaces or interfaces [31], the van der Waals surface (VWS), the solvent-accessible surface (SAS) and the solvent-excluded surface (SES) or the Lee-Richards surface [22]. The VWS is simply the boundary of the union of balls. As introduced in [22], the SAS is an inflated VWS with a probe sphere. The SES is a surface inside of which the probe never intrudes.

According to the properties of molecular structures, Laug and Borouchaki used a combined advancing-front and generalized-Delaunay approach to mesh molecular surfaces [21]. Algorithms were developed for sampling and triangulating a smooth surface with correct topology [1]. Skin surfaces, introduced by Edelsbrunner in [9], have a rich combinational structure and are suitable for modeling large molecules in biological computing. Cheng et. al [6] maintained an approximating triangulation of a deforming skin surface. Simplex subdivision schemes are used to generate tetrahedral meshes for molecular structures in solving the Poisson-Boltzmann equation [18]. Gaussian functions have been used to construct density maps [4] [17] [28], from which implicit solvation models are approximated as an isocontour [17] [23] [14]. However, it still remains a challenging problem to generate quality triangular and tetrahedral meshes for large complicated molecular structures.

**Mesh Generation:** As reviewed in [30] [38], octree-based, advancing front based and Delaunay like techniques were used for triangular and tetrahedral mesh generation. The octree technique recursively subdivides the cube containing the geometric model until the desired resolution is reached [33]. Advancing front methods start from a boundary and move a front from the boundary towards empty space within the domain [12] [25]. Delaunay refinement is to refine triangles or tetrahedra locally by inserting new nodes to maintain the Delaunay criterion (‘empty circumsphere’) [8]. Sliver Exudation [7] was used to eliminate those slivers. Shewchuk [34] solved the problem of enforcing boundary conformity by constrained Delaunay triangulation (CDT).

The predominant algorithm for isosurface extraction from volume data is Marching Cubes (MC) [26], which computes a local triangulation within each cube to approximate the isosurface by using a case table of edge intersections. MC was extended to extract tetrahedral meshes between two isosurfaces [13]. A different and systematic algorithm was proposed for interval volume tetrahedralization [29]. By combining SurfaceNets [16] and the extended Marching Cubes algorithm [20], octree based Dual Contouring [19] generates adaptive multiresolution isosurfaces with good aspect ratio and preserves sharp features. The dual contouring method has already been extended to extract adaptive and quality tetrahedral meshes from volumetric imaging data [42] [43].

**Quality Improvement:** Algorithms for mesh improvement can be classified into three categories [38] [30]: local coarsening/refinement by inserting/deleting points, local remeshing by face/edge swapping and mesh smoothing by relocating vertices.

Laplacian smoothing relocates the vertex position at the average of the nodes connecting to it [10]. Instead of relocating vertices based on a heuristic algorithm, the optimization technique measures the quality of the surrounding elements to a node and attempts to optimize it. The optimization-based smoothing yields better results while it is more expensive than Laplacian smoothing. Therefore, a combined Laplacian/Optimization-based approach was recommended [5] [11]. The Laplacian operator was discretized over triangular meshes [27], and geometric flows have been used in surface and imaging processing [32] [40]. Physically-based simulations are used to reposition nodes [24]. Anisotropic meshes are obtained from bubble equilibrium [35].

### 3 Implicit Solvation Surface from volumetric Density Maps

We extract the implicit solvation surface (molecular surface) as a level set of the volumetric synthetic electron density maps. The implicit solvation surface is chosen to be a good approximation of the Lee-Richards molecular surface [22] by choosing an appropriate parameter of the Gaussian kernel functions.

#### 3.1 Gaussian Density Map

As used for Poisson-Boltzmann electrostatics calculations in [18], a characteristic function  $f(x)$  is selected to represent an ‘inflated’ van der Waals-based accessibility

$$f(x) = \begin{cases} 1, & \text{if } \|x - x_i\| < r_i + \sigma \text{ for } i = 1, \dots, N, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where  $(x_i, r_i)$  are the centers and radii of the  $N$  atoms in the biomolecule, and  $\sigma$  is the radius of the diffusing species, here we choose  $\sigma = 2$  [37]. When  $\sigma = 0$ , the VWS is constructed. The function  $f(x)$  provides a grid-based volumetric data which can be isocontoured at the isovalue 0.5 to represent the SAS. Fig. 16(a) shows one constructed geometric model of mACH<sub>E</sub>.

Molecules are often modelled as the union of hard spheres  $S_i$  (atoms). The surface, denoted as  $M_0$ , of a molecule is therefore described as the boundary of the union of balls. To have the blurring effect at the intersection parts of atoms, the molecular surface is approximated by an isocontour [4]:

$$M := \{x \in \mathbb{R}^3 : G(x) = 1\} \quad (2)$$

with

$$G(x) = \sum_{i=1}^N e^{B_i \left( \frac{\|x - x_i\|^2}{r_i^2} - 1 \right)}, \quad (3)$$

where  $(x_i, r_i)$  are the center and radius of the  $i$ th atom in the biomolecule, and  $B_i < 0$  is called ‘decay rate’, which controls the blurring effect. Note that  $B_i$  must be negative to ensure that the density function goes to zero as  $\|x - x_i\|$  goes to infinity. In order to make the distance between  $M$  and  $M_0$  as uniformly as possible, we take

$$C = B_i / r_i^2, \quad (4)$$

where  $C < 0$  is a given constant. Now  $G(x)$  becomes

$$G(x, C) = \sum_{i=1}^N e^{C(\|x-x_i\|^2 - r_i^2)}. \quad (5)$$

The various presentation  $M(C_i) = \{x \in \mathbb{R}^3 : G(x, C_i) = 1\}$  of the molecular surface is therefore achieved by taking  $C = C_1, \dots, C_l$ .

As shown in Fig. 2, the different effects of  $C$  and constant  $B_i (= B)$  are studied in a two-sphere system, one is centering at  $(0, 0, 0)$  with the radius of 1.0, the other one is at  $(2.8, 0, 0)$  with the radius of 2.0. It could be observed that

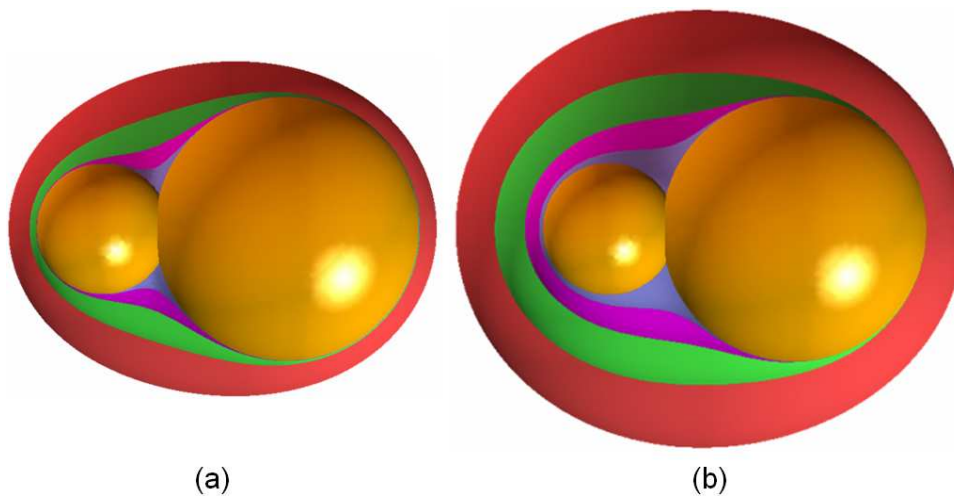


Fig. 2. Implicit Solvation models by choosing various  $C$  in (a) and  $B_i$  in (b). Yellow balls are two input atoms. The correspondence between  $C/B_i$  values and these models are shown in Table 1.

Table 1:  $C/B_i$  and Implicit Solvation Models in Fig. 2

	Red	Green	Magenta	Blue
Fig.2(a)	$C = -0.125$	$C = -0.25$	$C = -0.5$	$C = -1.0$
Fig.2(b)	$B_i = -0.125$	$B_i = -0.25$	$B_i = -0.5$	$B_i = -1.0$

- $C$  leads to more uniform inflation than  $B_i$ .
- Small balls have more inflation than big ones.
- Large error happens around the intersection region, and error reduces gradually away from it.
- Larger  $C$  and  $B_i$  lead to more inflation. For the same  $C$  and  $B_i$  value, e.g.,  $-0.125$ ,  $B_i$  tends to introduce more inflation.

Fig. 3 shows implicit solvation models of Ribosome 30S. Compared with Fig. 3(a), proteins inflate much more seriously in Fig. 3(e). rRNA in Fig. 3(c) and (f) looks similar, but proteins in Fig. 3(f) look a little more inflated than Fig. 3(b). rRNA in Fig. 3(d) and (g) looks similar too, but proteins in Fig. 3(g) are close to proteins in Fig. 3(c).

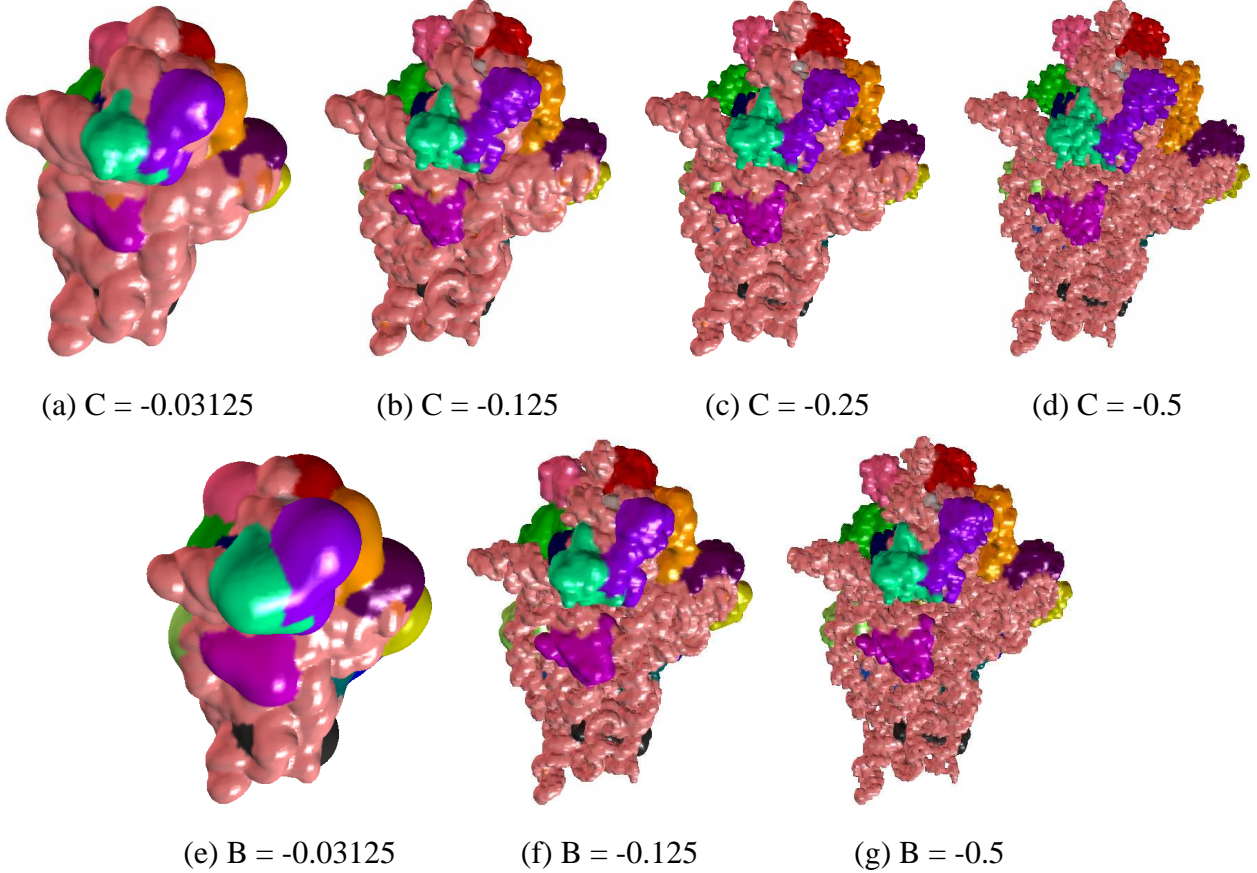


Fig. 3. Implicit solvation models of *Thermus Thermophilus* small Ribosome 30S (1J5E) crystal subunit for various Gaussian kernel parameters. The pink color shows 16S rRNA and the remaining colors are proteins.

### 3.2 Multi-Level Gaussian Density Map

In order to model different level structures, we introduce multi-level Gaussian map. First let us introduce some notations as shown in Fig. 4. Let  $N_0 = \{N_0^{(0)}, \dots, N_0^{(n)}\}$  denote the index set of all the atoms with  $N_0^{(i)} = \{i\}$ . Suppose  $N_0$  is grouped into several subsets  $N_1^{(i)}$ ,  $i = 1, 2, \dots, n_1$ , such that

$$\bigcup_{i=1}^{n_1} N_1^{(i)} = N_0, \quad N_1^{(i)} \cap_{1 \leq i \neq j \leq n_1} N_1^{(j)} = \phi. \quad (6)$$

The set  $N_1 := \{N_1^{(i)}\}_{i=1}^{n_1}$ , whose elements are also sets, may be further grouped into some subsets  $N_2^{(i)}$ ,  $i = 1, 2, \dots, n_2$ , such that

$$\bigcup_{i=1}^{n_2} N_2^{(i)} = N_1, \quad N_2^{(i)} \cap_{1 \leq i \neq j \leq n_2} N_2^{(j)} = \phi. \quad (7)$$

The collection of  $\{N_2^{(i)}\}_{i=1}^{n_2}$  is denoted by  $N_2$ . This hierarchical grouping process could be repeated several times according to the nature of the molecular complex considered. In practical, two or three iterations may be enough. By using these sets  $N_k^{(i)}$  and a given sequence  $\{p_k\}$  of  $p$  with

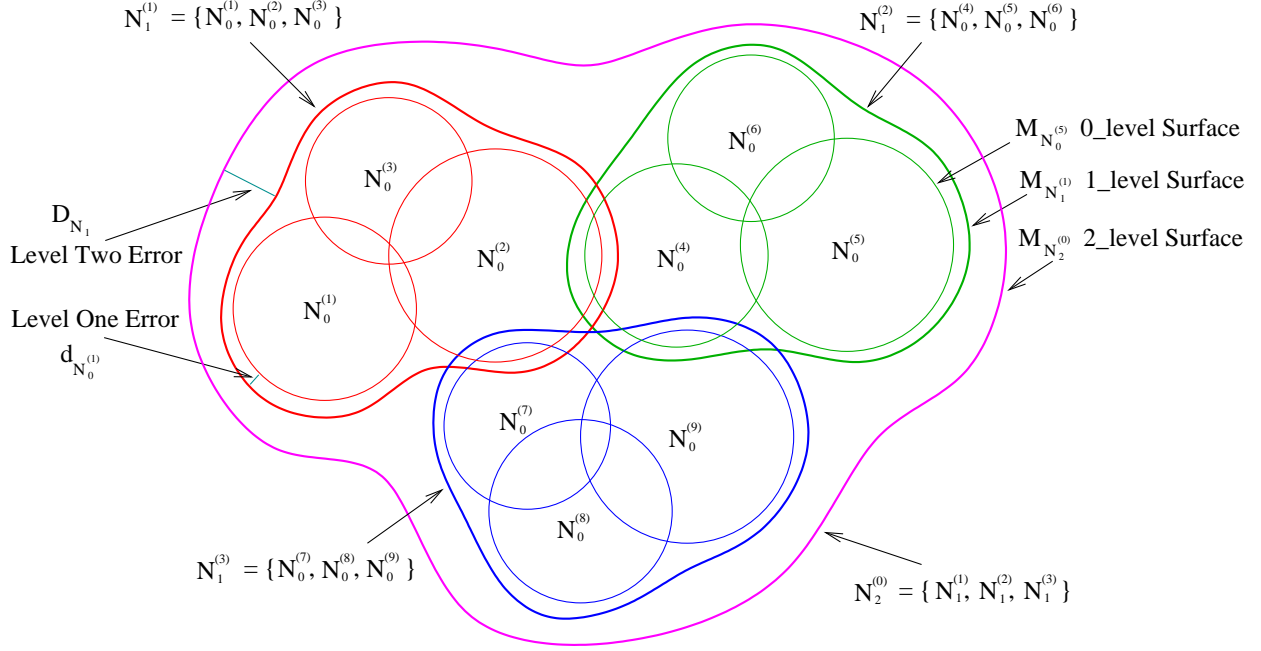


Fig. 4. The definition of multi-level surfaces.

$p_k > 0$ , the  $k$ -level Gaussian map are defined recursively as

$$G_{N_k^{(i)}}(x) = \sum_{N \in N_k^{(i)}} [G_N(x)]^{p_k}, \quad N_k^{(i)} \in N_k,$$

where 0-level Gaussian map is defined by Eqn. 5 ( $C = 1.0$ ) or

$$G_{N_0^{(i)}}(x) = K(\|x - x_i\|)/K(r_i), \quad K(x) = e^{-x^2}.$$

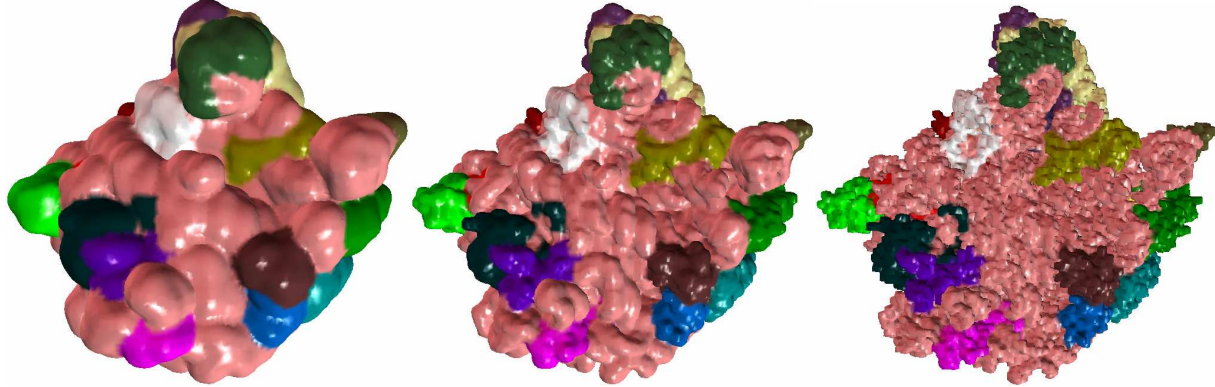
The atom group format depends on what kind of structure we want to address. For a protein, atoms may be grouped by residues, meaning that atoms in the same residue are classified into one group. Then the residues are grouped according to the secondary structure.

For each  $k$ -level Gaussian Map  $G_{N_k^{(i)}}(x)$ , a  $k$ -level surface is defined by

$$M_{N_k^{(i)}} := \{x \in \mathbb{R}^3 : G_{N_k^{(i)}}(x) = 1\}.$$

This surface encloses the surface  $M_N$  for  $N \in N_k^{(i)}$ . Hence, all these  $N_k^{(i)}$  define a hierarchical surface family. We call the surface  $M_N$  as the child of  $M_{N_k^{(i)}}$ , and  $M_{N_k^{(i)}}$  the parent of  $M_N$ . The enclosing relation of this hierarchical surface family is strict, meaning that the minimal distance from  $M_N$  to  $M_{N_k^{(i)}}$  is greater than zero for any  $N \in N_k^{(i)}$ . We further define the B-surface of  $M_N$  for





(a) low resolution

(b) residual level

(c) atomic level

Fig. 5. Implicit solvation models of Haloarcula Marismortui large Ribosome 50S (1JJ2) crystal subunit. (a)  $p_1 = 0.03125$ ; (b)  $p_1 = 0.125$ ; (c)  $p_1 = 0.5$ ,  $p_2 = 1.0$ . The light yellow and the pink color show 5S and 23S rRNA respectively, the remaining colors are proteins.

all  $N \in N_k^{(i)}$  as

$$S_{N_k^{(i)}} = \text{Bd} \left( \bigcup_{N \in N_k^{(i)}} \{x \in \mathbb{R}^3 : G_N(x) \leq 1\} \right),$$

where  $\text{Bd}()$  denotes the boundary of a region in  $\mathbb{R}^3$ . Note that  $S_{N_k^{(i)}}$  is enclosed strictly by  $M_{N_k^{(i)}}$ .

The aim to introduce multi-level Gaussian map is to address the structure of molecules at a certain level. For instance, at the residual level of a protein, we regard each residue as one unit and therefore the residue is modelled as one structure. The sub-structures of the residue (atoms), are not modelled individually. Similarly, at the secondary level, a group of residues is regraded as one unit while the residues are not regarded as individual structures. The goal of addressing certain level structure and un-addressing the higher level ones is achieved by the properly selection of the parameter  $p_k$  in the multi-level Gaussian map. Basically, larger  $p_k$  should be chosen to address  $k$ -level structure and smaller  $p_{k-1}$  is used to un-address  $(k-1)$ -level structures.

Considering three levels of structures, including the atomic, the residual and the second levels, we can construct three level Gaussian map with given  $p_1, p_2$  and  $p_3$ . To address the secondary structure, we need to choose  $p_3$  larger and  $p_2$  smaller, while  $p_1$  have less influence to the secondary structure. Therefore, we consider only two-level Gaussian map instead of three: Level one is the residual level, Level two is the low level. In this paper, we only consider two-level Gaussian map.

In implicit solvation modeling, various models are constructed by choosing different  $p_1 \in (0, \infty)$  and  $p_2 \in (0, \infty)$  in the Gaussian map. To make the constructed model correspond to a certain level,  $p_1$  and  $p_2$  need to be selected properly. For a fixed level, the structure at this level should be distinguishable, while the structure at a higher level may not be presented remarkably. For instance, at the residual level, the chain structure of residuals should be observed, while atoms may not be distinguished clearly. Fig. 5 shows constructed models of Ribosome 50S at the low resolution, residual and atomic levels.

### 3.3 Surface Point Classification and Parameterization

The adopted multi-level Gaussian map addresses certain substructures, which are visually distinguishable. To enhance the contrast, certain color maps may be used to paint the substructures. Therefore, a classification of the surface points is required. Basically, the problem is: for each point  $p \in M_{N_k^{(j)}}$  we need to classify this point by its child surfaces  $M_N, N \in N_k^{(j)}$ . Point  $p \in M_{N_k^{(j)}}$  belongs to  $M_N$  if and only if

$$G_N(p) \geq G_{\tilde{N}}(p) \quad \forall \tilde{N} \in N_k^{(j)} \quad N \neq \tilde{N}. \quad (8)$$

Such a classification is unique for point which makes the inequality above hold strictly, otherwise, it is not unique. According to this classification of surface,  $M_{N_k^{(j)}}$  is partitioned into patches, each of them belongs to one child surface. We denote the patch belonging to  $M_N$  by  $M_{N_k^{(j)}, N}$ .

The surface  $M_{N_k^{(j)}}$  could be classified according to zero level surfaces. That is, the point  $p \in M_{N_k^{(j)}}$  belongs to  $S_{N_0^{(i)}}$  if and only if

$$G_{N_0^{(i)}}(p) \geq G_{N_0^{(k)}}(p) \quad \forall i \neq k. \quad (9)$$

Let  $x_i$  and  $r_i$  be the center and radius of  $S_i$ . Then

$$P(p) := x_i + \frac{r_i(p - x_i)}{\|p - x_i\|} \quad (10)$$

is the nearest point from  $p$  to  $S_{N_0^{(i)}}$ . Equation (10) establishes a mapping from a patch of surface  $M_{N_k^{(j)}}$ , which belongs to sphere  $S_i$ , to a patch of sphere  $S_i$ . It follows from [3] that this patch of sphere could be represented exactly by a NURB patch over a  $(u, v)$  domain, with trimming curves as the boundaries of the patch. Therefore, the mapping (10) leads to a piecewise parameterization of the surface  $M_{N_k^{(j)}}$ .

### 3.4 Error Computation

In order to obtain the tight enclosing surface for the first level, we need to compute the minimal distance from  $M_N, N \in N_1^{(i)}$  to its parent surface  $M_{N_1^{(i)}}$ . On the other hand, in order to have an error controlled approximation of the second level surface, we need to compute the maximal error from  $M_N, N \in N_2^{(i)}$  to its parent surface  $M_{N_2^{(i)}}$ . Hence, we need to consider the error computation for two levels of surfaces. The error computations are based on a point projection algorithm.

Given the surface  $M_N$ , a point  $q \notin M_N$  and a unit direction  $n$ , the point projection algorithm in the following computes a nearby intersection point  $p$  of the line  $q + tn$  ( $t \in (-\infty, \infty)$ ) with the surface  $M_N$ .

**Algorithm 3.4.1** (*Point Projection*).

- (1) Compute an interval  $[a, b]$  for  $t$ , on which  $G_N(q + tn) - 1$  changes sign once. This is achieved by a linear search step in a certain range  $[A, B]$ . If  $(\nabla G_N(q))^T n [G_N(q) - 1] < 0$ , search in  $n$  direction, otherwise in  $-n$  direction. If such an interval could not be found, the project point does not exist and return a failure signal. After the interval is determined, set  $t_0 = \frac{a+b}{2}$  and  $k = 0$ .
- (2) Compute  $t_{k+1}$  by the Newton iteration method

$$t_{k+1} = t_k - \frac{G_N(q + t_k n)}{n^T \nabla G_N(q + t_k n)}. \quad (11)$$

- If  $t_{k+1} \notin (a, b)$ , replace  $t_{k+1}$  by  $\frac{a+b}{2}$ .
- (3) Replace the interval  $[a, b]$  by  $[a, t_{k+1}]$  if  $G_N(q + tn) - 1$  changes sign over  $[a, t_{k+1}]$ , and replace  $[a, b]$  by  $[t_{k+1}, b]$  otherwise.
  - (4) If  $|b - a| < \varepsilon$  ( $\varepsilon$  is a given error tolerance, we take it to be  $10^{-4}$ ), stop the iteration and  $p = q + t_{k+1}n$  is the projection point, otherwise, set  $k = k + 1$  and go back to step 2.

We specify the searching range  $[A, B]$  in step 1 of the algorithm to be  $[-4, 4]$ , since the atom diameters are around 4. Errors beyond that are not considered here. If the projection exists, then the projection point  $p$  of point  $q$  on the surface  $M_N$  in the direction  $n$  is denoted by  $P_{M_N}(q, n)$ .

### 3.4.1 Minimal Error of Level One Surface

Now we assume  $k = 1$ , then the child surfaces are atoms. Let  $N = \{j\} \in N_1^{(i)}$ , the minimal error from  $M_N = S_N$  to  $M_{N_1^{(i)}}$  is defined by

$$d_N := \min_{p \in M_{N_1^{(i)}, N}} \|p - x_j\| - r_j, \quad j \in N.$$

Let  $q = x_j + r_j \frac{p - x_j}{\|p - x_j\|}$ , then  $q$  is on the sphere  $S_N$  and  $p$  is the projection of  $q$  to the surface  $M_{N_1^{(i)}}$  in the spherical normal direction  $n(q)$ . That is,  $p = P_{M_{N_1^{(i)}}}(q, n(q))$ . Hence in order to compute  $d_N$ , we need to compute  $P_{M_{N_1^{(i)}}}(q, n(q))$  for  $q \in S_N$ .

Now we consider the computation of the minimal distance from  $M_N$  to  $M_{N_1^{(i)}}$ , where  $N \in N_1^{(i)}$ . First we assume that each atom (sphere) is uniformly sampled with  $m$  vertices. This sampling is achieved by translating a triangulated unit sphere to each of the atom center and re-scaling it to the atom size. We obtain the unit sphere triangulation from [39]. For each vertex  $q$  on the triangulated atom surface  $M_N$ ,  $P_{M_{N_1^{(i)}}}(q, n(q))$  is computed using the *point projection* algorithm, where  $n(q)$  is the spherical normal at  $q$ .

**Algorithm 3.4.2** (*Minimal Error computation*).

Set  $d_N = 4$ .

for each triangle vertex  $q \in S_N \cap S_{N_0}$  do {

compute  $P_{M_{N_1}}(q, n(q))$ , and then compute

$$d_N = \min\{d_N, \|P_{M_{N_1}}(q, n(q)) - x_j\| - r_j\}, \quad (12)$$

$$\text{if } P_{M_{N_1}}(q, n(q)) \in M_{N_1, N}.$$

}

Table 2 shows the minimal error of the level one surface for a residue and a chain from Ribosome 30S, where  $e(M)$  is defined as  $e(M) := \max_{N \in N_1^{(i)}} d_N$ . It can be observed that the error decreases as  $p$  increasing. The algorithm for computing minimal error can also be used to compute the maximal error by changing the min to max in (12). Maximal errors for Ribosome 30s are also listed in Table 2 for different  $p_1$  (see the second row).

Table 2: Minimal Error and Maximal Error of First Level Surfaces of Ribosome 30S (1J5E)

$p_1$	0.25	0.5	1.0	2.0	4.0	8.0	16.0
Min Error (atomic)	8.338e-02	2.829e-03	6.287e-06	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$
Max Error (atomic)	1.634e+00	8.656e-01	4.121e-01	2.038e-01	8.893e-02	3.940e-02	1.842e-02

### 3.4.2 Maximal Error of Level Two Surface

The maximal error from  $M_N$  to  $M_{N_2^{(i)}}$ ,  $N \in N_2^{(i)}$  is defined as

$$d_N := \max_{q \in M_N, P_{M_{N_2^{(i)}}}(q, n) \in M_{N_2^{(i)}, N}} \|q - P_{M_{N_2^{(i)}}}(q, n)\|,$$

where  $q \in M_N$ ,  $P_{M_{N_2^{(i)}}}(q, n)$  is the normal direction projection of  $q$  to the surface  $M_{N_2^{(i)}}$ . This error is computed as follows. Let  $N_1 \in N_2^{(i)}$ .

**Algorithm 3.4.3** (Maximal Error computation).

Set  $d_{N_1} = 0$ .

for each  $N \in N_1$  do {

for each triangle vertex  $q \in S_N \cap S_{N_0}$  do {

compute  $\tilde{q} := P_{M_{N_1}}(q, n(q))$ , and

compute  $P_{M_{N_2^{(i)}}}(\tilde{q}, n(\tilde{q}))$  if  $\tilde{q} \in M_{N_1, N}$

and then compute

$$d_{N_1} = \max\{d_{N_1}, \|P_{M_{N_1}}(q, n(q)) - P_{M_{N_2^{(i)}}}(\tilde{q}, n(\tilde{q}))\|\}$$

if  $P_{M_{N_2^{(i)}}}(\tilde{q}, n(\tilde{q})) \in M_{N_2^{(i)}, N_1}$ .

}

}

Again, the projection points  $\tilde{q} = P_{M_{N_1}}(q, n(q))$  and  $P_{M_{N_2^{(i)}}}(\tilde{q}, n(\tilde{q}))$  are computed by the point projection algorithm, where the searching range  $[A, B]$  is set to be  $[0, 4]$ , since we know  $M_{N_2^{(i)}}$  enclosing  $M_N$  and we are not interested in the errors that are larger than 4.

The first row of Table 3 shows the maximal errors of the second level (residual level) surfaces for ribosome 30s, where  $p_1$  is chosen to be 0.5,  $p_2 = 0.25, 0.5, 1.0, \dots, 16$ . The second row lists the maximal errors of the second level (low level) surfaces for the same  $p_1$  and  $p_2$ . The results show that the errors decrease in approximately linear rate as  $p_2$  increases.

Table 3: Maximal Error of Second Level Surfaces of Ribosome 30S (1J5E)

$p_2$	0.25	0.5	1.0	2.0	4.0	8.0	16.0
Max Error (residual)	3.923e+00	2.124e+00	6.832e-01	3.240e-01	1.550e-01	7.794e-02	3.278e-02
Max Error (low)	9.899e+00	7.695e+00	8.045e-01	2.365e-01	1.390e-01	6.113e-02	2.653e-02

### 3.5 Good Approximations of Molecular Surfaces

We have discussed that it is sufficient to consider two-level Gaussian map. To address certain structures,  $p_1$  is taken to be a small value to blur the higher level details,  $p_2$  is chosen to be larger to enhance the feature of the current level structure. As we have shown in the last section, a smaller  $p_1$  leads to a larger error for the level one surface, and a larger  $p_2$  leads to a smaller error for the second level surface. Therefore, our strategy for obtaining tight enclosing surface is to remove the level one error and ignore the error of the second level.

The main idea to obtain the tight level one enclosing surface  $M_{N_1^{(1)}}$  is to reduce the radii of the atoms, such that  $M_{N_1^{(i)}}$  touches the original atoms (see Fig. 6). Suppose  $y \in M_{N_1^{(i)}}$  is the nearest point to the  $j$ -th atom,  $j \in N_1^{(i)}$ , and the distance from  $y$  to the atom is  $d_j$ . Then we have

$$\sum_{l \in N_1^{(i)}, l \neq j} [K(\|y - x_l\|)/K(r_l)]^{p_1} + [K(\|y - x_j\|)/K(r_j)]^{p_1} = 1. \quad (13)$$

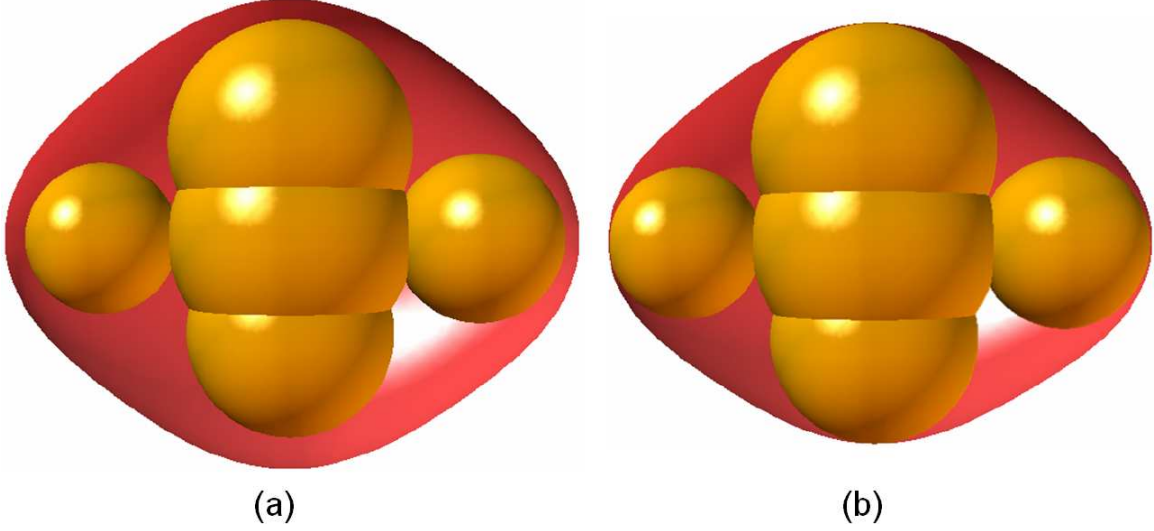


Fig. 6. The left picture shows the inflation effect by the Gaussian map. The right one shows the tight enclosing of atoms. The centers of the five atoms are  $(-2, 0, 0)$ ,  $(2, 0, 0)$ ,  $(0, -1, 0)$ ,  $(0, 1, 0)$  and  $(0, 0, 0)$ . The corresponding radii are 0.8, 0.9, 1.1, 1.3 and 1.3. The parameter  $p$  in the Gaussian map is chosen to be 0.4. The tight approximation on the right figure is obtained by shrinking the five radii into 0.55644, 0.72525, 0.60476, 1.04567 and 0.0 respectively.

where  $K(x) = e^{-x^2}$ . Now we want to adjust the radius  $r_j$  to  $\tilde{r}_j$ , such that the new nearest point  $y$  is on the  $j$ -th sphere. Since the dominate part of (13) is the second term of the left hand side, we therefore require  $\tilde{r}_j$  satisfying

$$0 \leq \tilde{r}_j \leq r_j, \quad (14)$$

$$K(r_j + d_j)/K(r_j) = K(r_j)/K(\tilde{r}_j). \quad (15)$$

From this we obtain

$$\tilde{r}_j = \begin{cases} K^{-1} \left[ \frac{K(r_j)^2}{K(r_j + d_j)} \right], & \text{if } \frac{K(r_j)^2}{K(r_j + d_j)} \in \text{Rang}(K), \\ 0, & \text{otherwise,} \end{cases}$$

where  $K^{-1}$  denotes the inverse function of  $K(x)$ ,  $\text{Rang}(K) := \{y \in \mathbb{R} : y = K(x), x \in (0, \infty)\}$ . Based on this analysis, we build the following iterative algorithm for computing  $\tilde{r}_j$ .

**Algorithm 3.5.1** (*Sphere Shrinking*).

For  $i = 1, 2, \dots, n_1$  do the following steps:

- (1) Set  $l = 0$ ,  $r_j^{(l)} = r_j$ ,  $d_j^{(l)} = \infty$ ,  $\forall j \in N_1^{(i)}$ .
- (2) Compute the minimal distance  $d_j^{(l+1)}$ ,  $\forall j \in N_1^{(i)}$  from the  $j$ -th atom to the iso-surface defined by the generalized Gaussian map  $G_{N_1^{(i)}}^{(l)}(x) = \sum_{j \in N_1^{(i)}} [K(\|x - x_j\|)/K(r_j^{(l)})]^{p_1}$ , using the Algorithm 4.2.

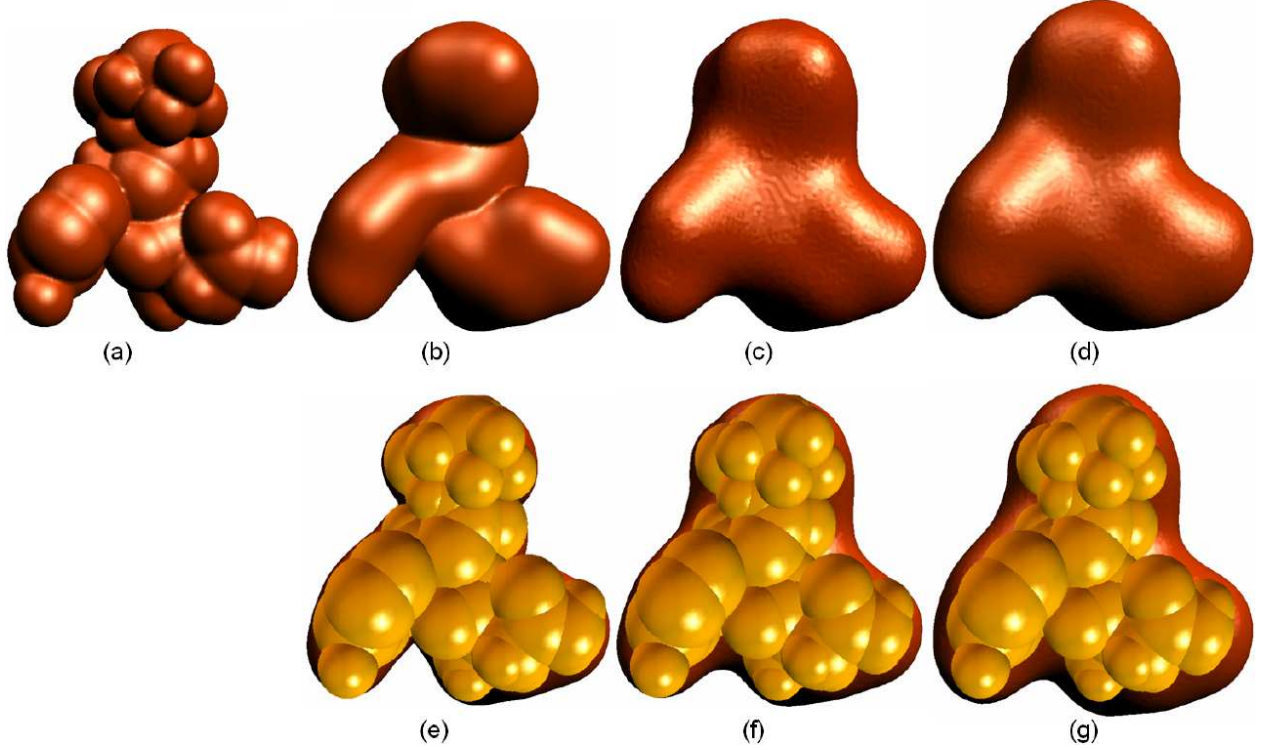


Fig. 7. Different effects of changing  $p_2$  and tight/non-tight approximations for three amino acids (ASN, THR and TYR) which consist of 49 atoms. The surface (b), (c) and (d) are the same as outer surfaces of (e), (f) and (g) respectively. The inner surface of (e), (f) and (g) is the hard sphere model of three residues. (a) shows the atomic level approximation of the hard sphere model, where  $p_1 = 5.0$ ,  $p_2 = 1.0$ ; (b), (e), (c) and (f) show the tight approximation of the residual level with  $p_1 = 0.4$ . But different  $p_2$  are used. We choose  $p_2 = 2.0$  for (b)  $p_2 = 0.5$  for (c). It could be observed that larger  $p_2$  leads to closer approximation. (d) and (g) show non-tight approximations using the same  $p_1$  and  $p_2$  as (c) and (f). Comparing with (f), even larger error is observed in (g).

(3) Compute

$$r_j^{(l+1)} = \begin{cases} K^{-1} \begin{bmatrix} K(r_j)K(r_j^{(l)}) \\ K(r_j+d_j^{(l)}) \end{bmatrix}, & \text{if } \frac{K(r_j)K(r_j^{(l)})}{K(r_j+d_j^{(l)})} \in \text{Rang}(K), \\ 0, & \text{otherwise.} \end{cases}$$

(4) If  $\max_{j \in N_1^{(l)}} |d_j^{(l)} - d_j^{(l+1)}| < \varepsilon$  (we take  $\varepsilon = 10^{-4}$ ), terminate the  $l$  loop and  $r_j^{(l+1)}$  are the required results. Otherwise, set  $l = l + 1$  and go back to step 2.

**Remark:** The condition  $\frac{K(r_j)K(r_j^{(l)})}{K(r_j+d_j^{(l)})} \in \text{Rang}(K)$  may lead to some of the atoms locating in the deep inside of the molecule are not touchable. Figure 6 shows that the circle at the origin is not touched.

The experiments show the sphere shrinking algorithm converges in a linear rate. Table 4 lists the error  $e_{max}^{(l)} = \max_{j \in N_1^{(l)}} |d_j^{(l)}|$  for 20 amino acids with  $p_1 = 0.4$ . These amino acids are taken from the protein mAChE. We notice that the shape of different copies of one amino acids will differ. Currently, we are studying theoretically the convergence of the algorithm.

Table 4: Errors  $e_{max}^{(l)}$  for 20 amino acids and  $p_1 = 0.4$

$l$	ALA	ARG	ASN	ASP	CYS	GLN	GLU	GLY	HSD	ILE
0	5.13e-01	6.97e-01	5.99e-01	6.23e-01	5.36e-01	6.26e-01	7.06e-01	4.34e-01	7.36e-01	6.00e-01
2	6.22e-02	1.37e-01	2.66e-01	6.75e-02	5.86e-02	1.16e-01	7.78e-02	5.33e-02	7.20e-02	5.62e-02
4	2.80e-03	3.79e-02	5.83e-02	1.50e-03	6.82e-04	1.76e-03	4.57e-04	1.90e-02	1.45e-02	2.73e-03
6	5.76e-04	2.30e-02	1.83e-04	4.93e-04	1.81e-04	4.51e-04	1.38e-04	8.62e-05	5.30e-03	5.60e-04
8	1.30e-04	6.95e-04	6.06e-05	1.64e-04	4.97e-05	1.74e-04	4.26e-05	6.31e-06	2.20e-03	1.25e-04
10	3.14e-05	2.18e-04	2.22e-05	5.59e-05	1.39e-05	7.84e-05	1.32e-05	7.16e-07	9.94e-04	3.11e-05

$l$	LEU	LYS	MET	PHE	PRO	SER	THR	TRP	TYR	VAL
0	8.48e-01	8.62e-01	6.08e-01	6.14e-01	7.98e-01	9.63e-01	1.06e-00	6.01e-01	6.10e-01	7.07e-01
2	6.51e-02	3.96e-01	1.13e-01	8.94e-02	2.06e-03	8.81e-02	3.06e-02	9.17e-02	6.03e-02	2.86e-02
4	5.72e-03	1.54e-03	7.78e-03	6.50e-03	3.62e-04	5.28e-04	6.63e-03	1.49e-02	4.25e-02	5.76e-03
6	1.27e-03	5.18e-04	2.25e-03	1.90e-03	9.12e-05	1.19e-04	1.68e-03	6.42e-03	1.69e-03	1.36e-03
8	3.03e-04	1.77e-04	6.77e-04	7.13e-04	2.35e-05	2.66e-05	4.67e-04	2.90e-03	6.93e-04	3.56e-04
10	7.52e-05	6.23e-05	2.09e-04	3.02e-04	6.26e-06	5.88e-06	1.36e-04	1.56e-03	3.52e-04	9.78e-05

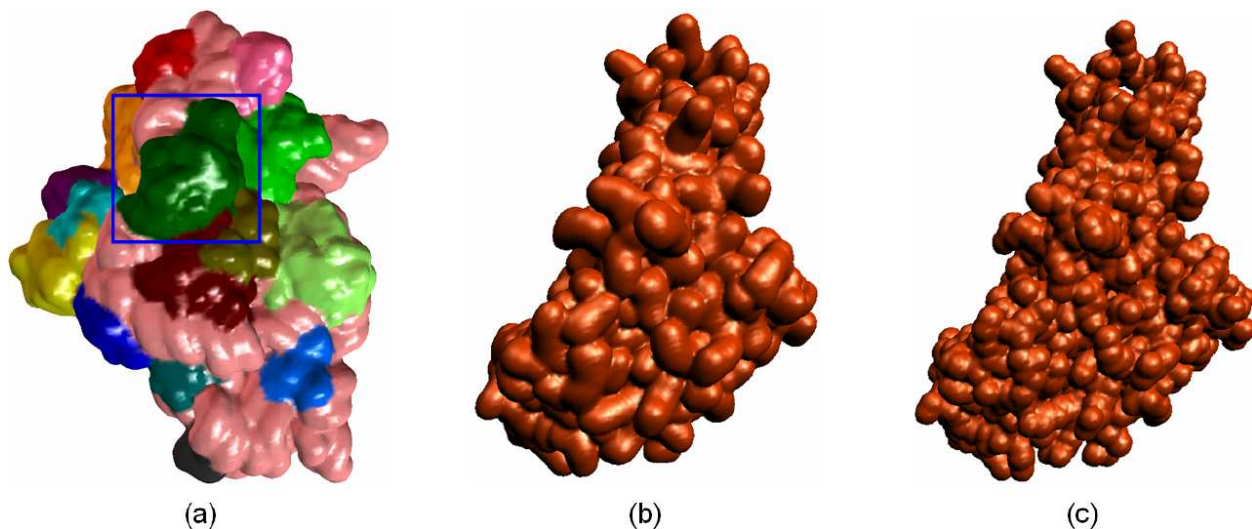


Fig. 8. Multi-resolution models of Ribosome 30S. (a) - Ribosome 30S at the low level with  $p_1 = 0.0625$ ,  $p_2 = 1.0$  in multi-level Gaussian map. Ribosome 30S contains 22 chains and each of them is painted in a different color. The pink color shows 16S rRNA and the remaining colors are proteins. The blue box shows one protein (Chain B). (b) - Chain B at the residual level with  $p_1 = 0.4$ ,  $p_2 = 5.0$ . It consists of 234 residues. (c) - Chain B at the atomic level with  $p_1 = 5.0$ ,  $p_2 = 1.0$ . It consists of 1900 atoms.

Fig. 7 shows multi-resolution models of the amino acid ASN, THR and TYR with various  $p_1$  and  $p_2$ . Fig. 7(a) shows an atomic level model, Fig. 7(a~g) are residual level models. It can be observed that when the same  $p_1$  is selected, smaller  $p_2$  leads to fatter surfaces. Compared with Fig. 7(g), Fig. 7(f) is more tight.

Fig. 8 shows multi-resolution models of Ribosome 30S. Fig. 8(a) is a low level model, the pink color shows 16S rRNA and the remaining colors are proteins. One protein (Chain B) is separated from the whole structure. The residual level model can be constructed by selecting small  $p_1$  and



large  $p_2$  as shown in Fig. 8(b), and the atomic level model is constructed by selecting large  $p_1$  and small  $p_2$  as shown in Fig. 8(c).

## 4 Mesh Generation

There are two main methods for contouring scalar fields, primal contouring [26] and dual contouring [19]. Both of them can be extended to tetrahedral mesh generation. The dual contouring method [42] [43] is often the method of choice as it tends to yield meshes with better aspect ratio.

### 4.1 Triangular Meshing

Dual contouring [19] uses an octree data structure, and analyzes those edges that have endpoints lying on different sides of the isosurface, called *sign change edges*. The mesh adaptivity is determined during a top-down octree construction. Each sign change edge is shared by either four (uniform case) or three (adaptive case) cells, and one minimizer point is calculated for each of them by minimizing a predefined Quadratic Error Function (QEF) [15]:

$$QEF[x] = \sum_i [n_i \cdot (x - p_i)]^2, \quad (16)$$

where  $p_i$ ,  $n_i$  represent the position and unit normal vectors of the intersection point respectively. For each sign change edge, a quad or triangle is constructed by connecting the minimizers. These quads and triangles provide a ‘dual’ approximation of the isosurface.

A recursive cell subdivision process was used to preserve the correct topology [43] of the isosurface. During the cell subdivision, the function value at each newly inserted grid point can be exactly calculated since we know the function (Gaussian functions, Eqn. (5)). Additionally, we can generate a more accurate triangular mesh by projecting each generated minimizer point onto the isosurface (Eqn. (2)).

### 4.2 Tetrahedral Meshing

The dual contouring method has already been extended to extract tetrahedral meshes from volumetric scalar fields [42] [43]. The cells containing the isosurface are called boundary cells, and the interior cells are those cells whose eight vertices are inside the isosurface. In the tetrahedral mesh extraction process, all the boundary cells and the interior cells need to be analyzed in the octree data structure. There are two kinds of edges in boundary cells, one is a sign change edge, the other is an interior edge. Interior cells only have interior edges. In [42] [43], interior edges and interior faces in boundary cells are dealt with in a special way, and the volume inside boundary cells is tetrahedralized. For interior cells, we only need to split them into tetrahedra.

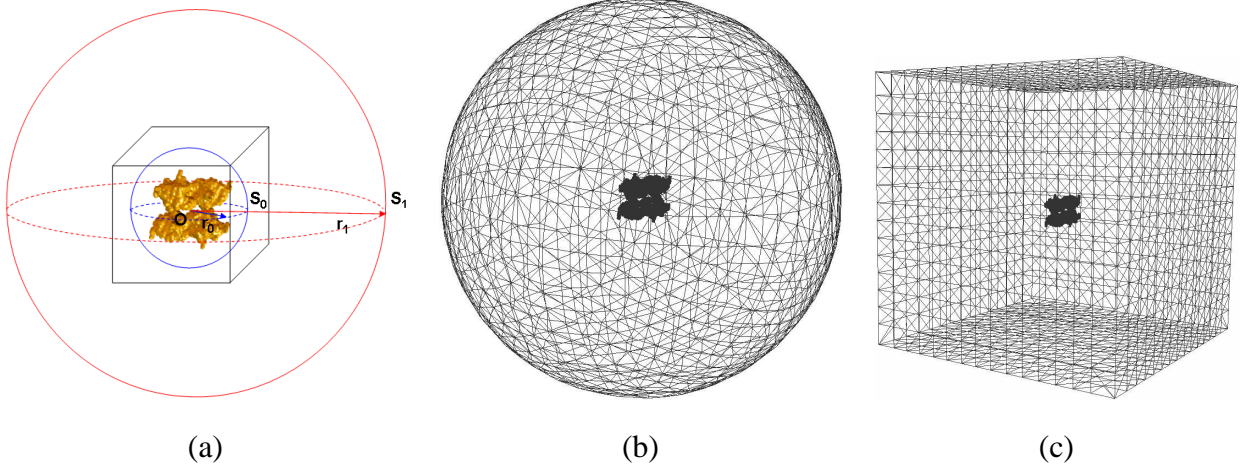


Fig. 9. The analysis domain of exterior meshes. (a) - ‘ $O$ ’ is the geometric center of the molecule, suppose the circum-sphere of the biomolecule has the radius of  $r$ . The box represents the volumetric data, and ‘ $S_0$ ’ is the maximum sphere inside the box, the radius is  $r_0$  ( $r_0 > r$ ). ‘ $S_1$ ’ is an outer sphere with the radius of  $r_1$  ( $r_1 = (20 \sim 40)r$ ). (b) - the diffusion domain is the interval volume between the biomolecular surface and the outer sphere ‘ $S_1$ ’, here we choose  $r_1 = 5r$  for visualization. (c) - the outer boundary is a cubic box.

#### 4.2.1 Adding an Outer Boundary

In the biological diffusion system, we need to analyze the field which is from infinite faraway to the molecular surface. Assume that the radius of the circum-sphere of a biomolecule is  $r$ . The computational model can be approximated by a field from an outer sphere  $S_1$  with the radius of  $(20 \sim 40)r$  to the molecular surface. Therefore the exterior mesh is defined as the tetrahedralization of the interval volume between the molecular surface and the outer sphere  $S_1$  (Fig. 9(b)). Sometimes the outer boundary is chosen to be a cubic box as shown in Fig. 9(c).

First we add a sphere  $S_0$  with the radius of  $r_0$  (where  $r_0 > r$  and  $r_0 = 2^n/2 = 2^{n-1}$ ) outside the molecular surface, and generate meshes between the molecular surface and the outer sphere  $S_0$ . Then we extend the tetrahedral meshes from the sphere  $S_0$  to the outer bounding sphere  $S_1$ . For each data point inside the molecular surface, we keep the original function value. While for each data point outside the molecular surface, we reset the function value as the smaller one of  $f(x) - \alpha$  and the shortest distance from the grid point to the sphere  $S_0$ . Eqn. (17) shows the newly constructed function  $g(x)$  which provides a grid-based volumetric data containing the biomolecular surface and an outer sphere  $S_0$ .

$$g(x) = \begin{cases} \min(\|x - x_0\| - r_0, f(x) - \alpha), & \text{if } f(x) < \alpha, \|x - x_0\| < r_0, \\ \|x - x_0\| - r_0, & \text{if } f(x) < \alpha, \|x - x_0\| \geq r_0, \\ f(x) - \alpha, & \text{if } f(x) \geq \alpha, \end{cases} \quad (17)$$

where  $x_0$  are coordinates of the molecular geometric center. The isovalue  $\alpha = 0.5$  for volumetric data generated from the characteristic function, and  $\alpha = 1.0$  for volumetric data generated from the summation of Gaussian kernels.

The biomolecular surface and the outer sphere  $S_0$  can be extracted as an isosurface at the isovalue

0,  $S_g(0) = \{x|g(x) = 0\}$ . All the grid points inside the interval volume  $I_g(0) = \{x|g(x) \leq 0\}$  have negative function values, and all the grid points outside it have positive values.

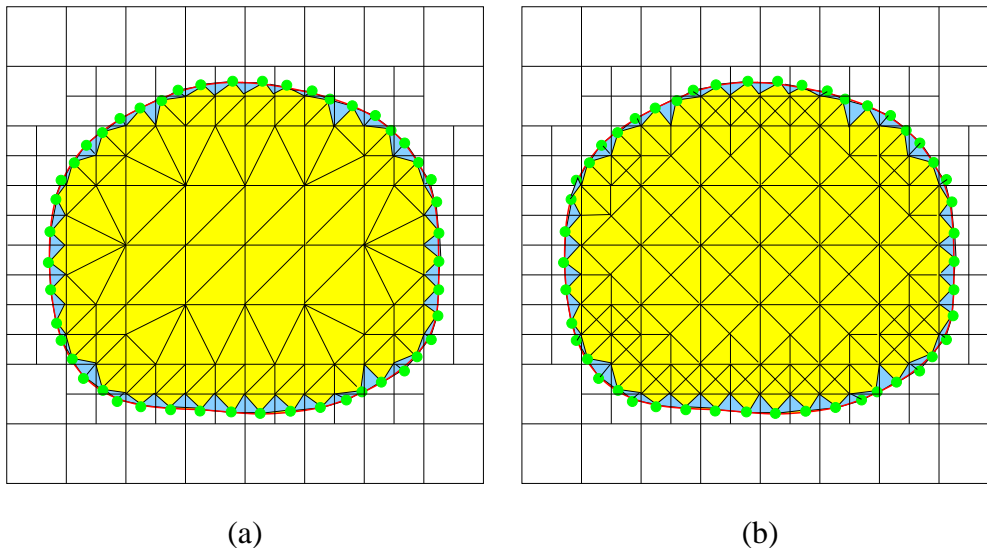


Fig. 10. 2D triangulation. (a) Old scheme, (b) New scheme. Blue and yellow triangles are generated for sign change edges and interior edges respectively. The red curve represents the molecular surface, and the green points represent minimizer points.

#### 4.2.2 Primal Mesh Extraction

Here we introduce a different scheme from the algorithm presented in [42] [43], in which we do not distinguish boundary cells and interior cells when we analyze edges. We only consider two kinds of edges - sign change edges and interior edges. For each boundary cell, we can obtain a minimizer point by minimizing its Quadratic Error Function. For each interior cell, we set the middle point of the cell as its minimizer point. Fig. 10(b) shows a simple 2D example. In 2D, there are two cells sharing each edge, and two minimizer points are obtained. For each sign change edge, the two minimizers and the interior vertex of this edge construct a triangle (blue triangles). For each interior edge, each minimizer point and this edge construct a triangle (yellow triangles). In 3D as shown in Fig. 11, there are three or four cells sharing each edge. Therefore, the three (or four) minimizers and the interior vertex of the sign change edge construct one (or two) tetrahedron, while the three (or four) minimizers and the interior edge construct two (or four) tetrahedra.

Compared with the algorithm presented in [42] [43] as shown in Fig. 10(a), Fig. 10(b) generates the same surface meshes, and tends to generate more regular interior meshes with better aspect ratio, but a little more elements for interior cells. Fig. 10(b) can be easily extended to large volume decomposition. For arbitrary large volume data, it is difficult to import all the data into memory at the same time. So we first divide the large volume data into some small subvolumes, then mesh each subvolume separately. For those sign change edges and interior edges lying on the interfaces between subvolumes, we analyze them separately. Finally, the generated meshes are merged together to obtain the desired mesh. The mesh adaptivity is controlled by the structural properties of biomolecules. The extracted tetrahedral mesh is finer around the molecular surface, and gradually

gets coarser from the molecular surface out towards the outer sphere,  $S_0$ . Furthermore, we generate the finest mesh around the active site, such as the cavity in the monomeric and tetrameric mAChE shown in Fig.16 (a~b), and a coarse mesh everywhere else.

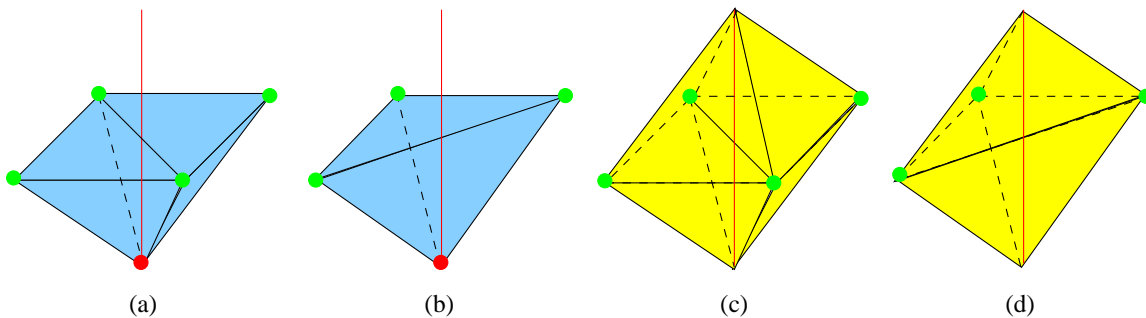


Fig. 11. Sign change edges and interior edges are analyzed in 3D tetrahedralization. (a)(b) - sign change edge (the red edge); (c)(d) - interior edge (the red edge). The green solid points represent minimizer points, and the red solid points represent the interior vertex of the sign change edge.

#### 4.2.3 Mesh Extension

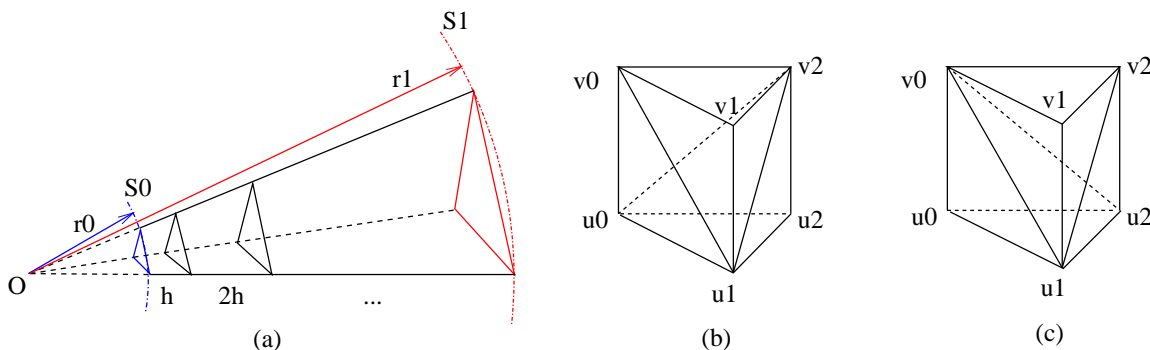


Fig. 12. (a) - one triangle in the sphere  $S_0$  (blue) is extended  $n$  steps until arriving the sphere  $S_1$  (red); (b) and (c) - a prism is decomposed into three tetrahedra in two different ways.

We have generated meshes between the biomolecular surface and the outer sphere  $S_0$ , the next step is to construct tetrahedral meshes gradually from the sphere  $S_0$  to the bounding sphere  $S_1$  (Fig. 9). The sphere  $S_0$  consists of triangles, so we extend each triangle radially as shown in Fig. 12 and a prism is obtained for each extending step. The prism can be divided into three tetrahedra. The extension step length  $h$  can be calculated by Eqn. (18). It is better for the sphere  $S_0$  to be triangulated uniformly since the step length is fixed for each extending step.

$$r_0 + h + 2h + \dots + nh = r_1 \implies h = \frac{2(r_1 - r_0)}{n(n+1)} \quad (18)$$

where  $n$  is the step number. In Figure 12, suppose  $u_0u_1u_2$  is a triangle on sphere  $S_0$ , and  $u_0, u_1, u_2$  are the unique index numbers of the three vertices, where  $u_1 < u_0$  and  $u_1 < u_2$ . For one extension step,  $u_0u_1u_2$  is extended to  $v_0v_1v_2$ , and the two triangles construct a prism, which can be decomposed into three tetrahedra. In order to avoid the diagonal conflict problem, a different

decomposition method (Fig. 12(b~c)) is chosen based on the index number of the three vertices. If  $u_0 < u_2$ , then we choose Fig. 12(b) to split the prism into three tetrahedra. If  $u_2 < u_0$ , then Fig. 12(c) is selected

Assume there are  $m$  triangles on the sphere  $S_0$ , which is extended  $n$  steps to arrive the sphere  $S_1$ .  $m$  prisms or  $3m$  tetrahedra are generated in each extending step, and a total of  $3mn$  tetrahedra are constructed in the extension process. Therefore, it is better to keep coarse and uniform triangular mesh on the sphere  $S_0$ .

## 5 Quality Improvement

In general, the molecular surface generated by iscontouring the Gaussian density function or the characteristic function is bumpy. This is because the volume data could not be sufficiently fine due to the capacity limit of the computer, and is not smooth enough, especially for the data generated from the characteristic function. The error of the isosurface from the characteristic function could be as large as half of the grid size since the characteristic function generates binary volumetric data, and could be very large relative to the atom size. Therefore, a post-processing step for the extracted isosurface is necessary. There are three tasks for the mesh quality improvement:

- (1) Denoising the surface mesh (vertex adjustment in the normal direction).
- (2) Improving the aspect ratio of the surface mesh (vertex adjustment in the tangent direction).
- (3) Improving the aspect ratio of the volumetric mesh (vertex adjustment inside the volume).

We use geometric partial differential equations (PDEs) to handle the first two problems. Geometric PDEs, such as the mean curvature flow, the surface diffusion flow and Willmore flow, have been intensively used in surface and imaging processing [40]. Here we choose the surface diffusion flow to smooth the molecular surface.

$$\frac{\partial x}{\partial t} = \Delta H(x) \vec{n}(x), \quad (19)$$

where  $H$  is the mean curvature,  $\vec{n}$  is the unit surface normal vector, and  $\Delta$  is the Laplacian-Beltrami operator. This flow is area shrinking and volume preserving. Furthermore, it preserves sphere exactly and torus approximately. Suppose a molecular surface could be ideally represented by the joining of spherical and torus surface patches [21], it is desirable to use the surface diffusion flow to evolve the isosurface. However, this flow could only improve the surface shape, not the mesh regularity. In order to improve the aspect ratio, we need to add a tangent movement in Eqn. (19). Hence the flow becomes

$$\frac{\partial x}{\partial t} = \Delta H(x) \vec{n}(x) + v(x) \vec{T}(x), \quad (20)$$

where  $v(x)$  is the velocity in the tangent direction  $\vec{T}(x)$ .

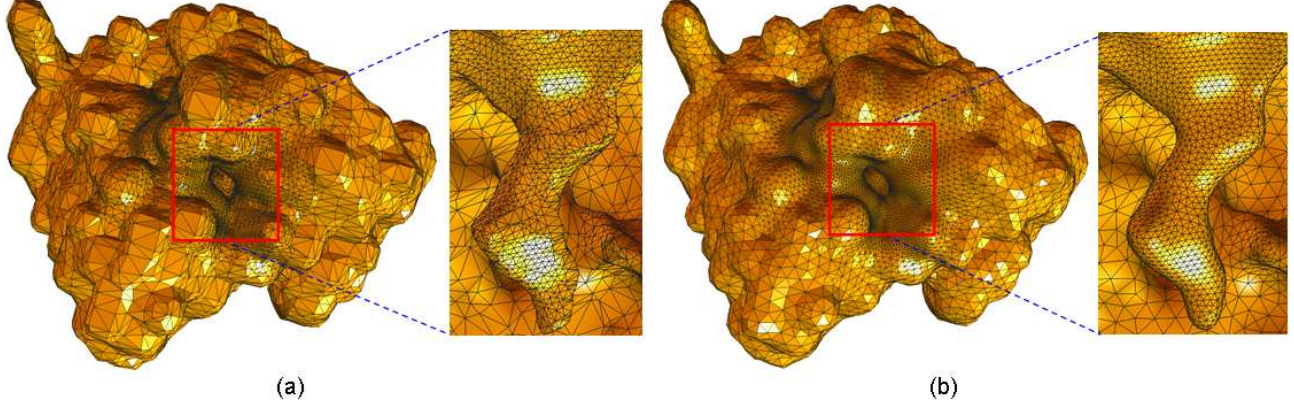


Fig. 13. Comparison of mChE (9308 vertices, 18612 triangles) before and after surface smoothing. (a) - original; (b) - after smoothing.

Eqn. (20) is solved over a triangular mesh with vertices  $\{x_i\}$  by discretizing each of its terms. In the temporal space,  $\frac{\partial x}{\partial t}$  is approximated by the Euler scheme  $\frac{x_i^{n+1} - x_i^n}{\tau}$ , where  $\tau$  is time step-length.  $x_i^n$  is the approximating solution at  $t = n\tau$ ,  $x_i^{n+1}$  is the approximating solution at  $t = (n+1)\tau$ , and  $x_i^0 = x_i$  serves as the initial value. Discretizing schemes for  $\Delta$  and  $H$  in the spatial space are given in [40], we do not go to detail here.  $v(x)\vec{T}(x)$  is approximated by

$$[m(x_i^n) - x_i^n] - \vec{n}(x_i^n)^T [m(x_i^n) - x_i^n] \vec{n}(x_i^n), \quad (21)$$

where  $m(x_i^n)$  is defined as the mass center of all the triangles around  $x_i^n$ . A mass center  $P$  of a region  $V$  is defined by finding  $p \in V$ , such that

$$\int_V \|y - p\|^2 d\sigma = \min. \quad (22)$$

$V$  could be a piece of surface or a volume in  $\mathbb{R}^3$ . For our surface mesh case,  $V$  consists of triangles around vertex  $x_i^n$ . Then from Eqn. (22), we could derive that

$$m(x_i^n) = \frac{1}{3}x_i^n + \frac{1}{3} \sum_{j \in N(i)} x_j^n (\Delta_j + \Delta_{j+1}) / A(x_i^n), \quad (23)$$

where  $N(i)$  is the index set of the one ring neighbors of  $x_i^n$ .  $\Delta_j$  is the area of the triangle  $[x_i^n x_{j-1}^n x_j^n]$ .  $A(x_i^n)$  is the total of triangle areas.

Usually, people use the geometric center [40], instead of the mass center. But we found that the mass center works better. The discretization will lead to a linear system. The approximated solution is obtained by solving it.

After the molecular surface is smoothed and regularized, the next step is to improve the volumetric mesh by relocating each interior vertex to the mass center of its surrounding tetrahedra. Let  $p_i$  be

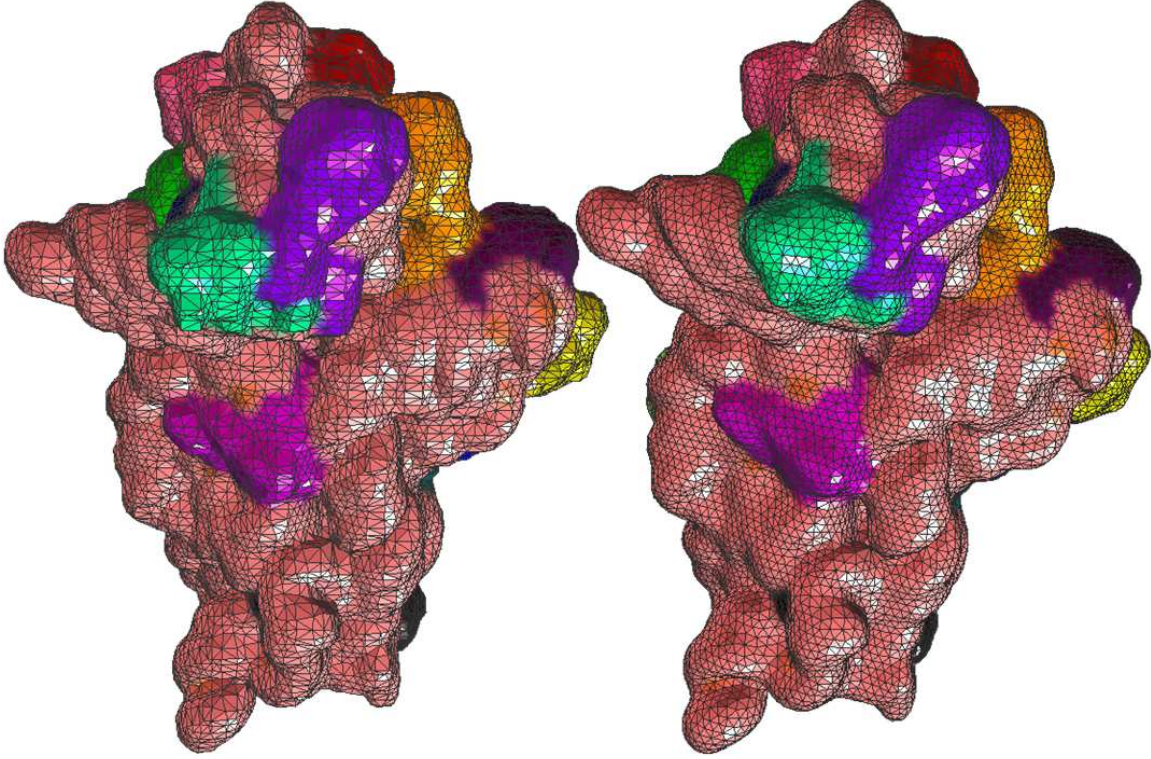


Fig. 14. Comparison of Ribosome 30S (13428 vertices, 26852 triangles) before and after surface smoothing. Left - original; Right - after smoothing.

an interior vertex,  $p_j$  be one of its neighboring vertices, then the mass center of all tetrahedra around  $p_i$  is computed by

$$m(p_i) = \frac{1}{4}p_i + \frac{1}{4V_i} \sum_j V_{ij}p_j, \quad (24)$$

where  $V_{ij}$  is the volume summation of all the tetrahedra around the edge  $[p_i p_j]$ ,  $V_i$  is the volume summation of the tetrahedra around the vertex  $p_i$ .

Here we choose the aspect ratio (twice of the ratio of incircle radius to circumcircle radius) to measure the quality of triangular meshes, and the surface diffusion flow to smooth the surface. The left picture in Fig.15 shows the improvement of the aspect ratio, and Fig.13~14 show the improvement of molecular surfaces. We can see that noises are removed and features are preserved since the surface diffusion flow preserves volume and spherical geometry. The surface error is restricted within half of the grid size for the binary data from the characteristic function, and almost zero for the data from Gaussian density map since we have projected each boundary vertex onto the isosurface.

In [43], the edge contraction and linear averaging method was used to improve the quality of tetra meshes with the edge-ratio (the longest edge length over the shortest edge length) and Joe-Liu parameter  $(2^{\frac{4}{3}} \times 3 \times (|V|)^{\frac{2}{3}} / \sum_{0 \leq i < j \leq 3} |e_{ij}|^2)$ , where  $|V|$  denotes the volume,  $e_{ij}$  represents the

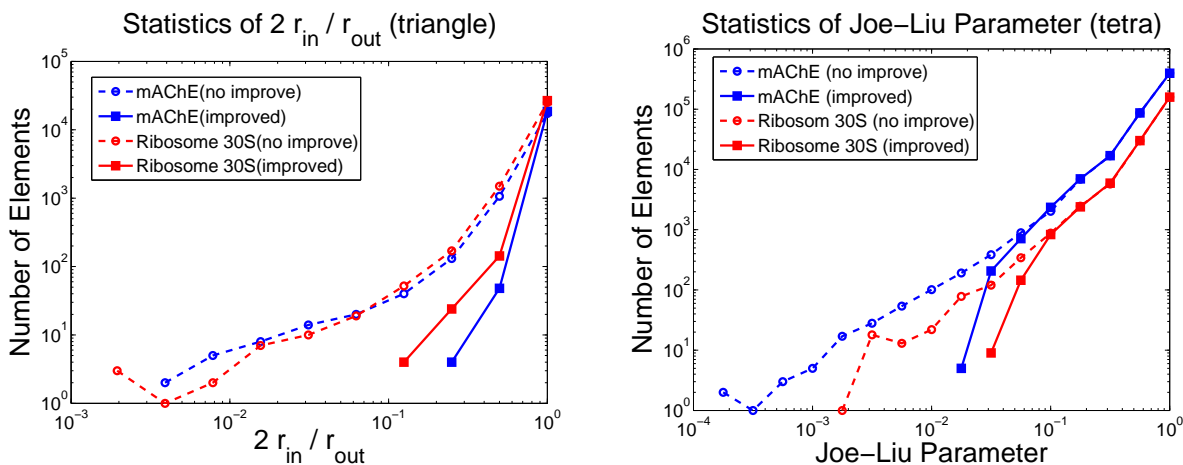


Fig. 15. The histogram of the aspect-ratio and Joe-Liu parameter.

edge connecting vertex  $v_i$  and  $v_j$ ) as metrics. The goal is to improve the worst parameters in each iteration. Here we still use the same edge contraction scheme, but relocate each interior vertex to its mass center (Eqn.(24)) since it can minimize the energy defined in Eqn.(22). From the right picture in Fig.15, we can see that the worst Joe-Liu parameter is improved significantly after quality improvement. Fig.16 and 18 show interior tetra meshes of mACHe and Ribosome 30S.

## 6 Results and Conclusion

**Monomeric mACHe:** For efficient and accurate finite element calculations, adaptive meshes are preferred. Therefore we generated finer meshes around the cavity region since the bottom of the gorge is the active site in mACHe, while coarser meshes in other regions. The extracted tetrahedral meshes of the monomer as shown in Fig. 16 have been used as the geometric model in the finite element analysis of the steady-state Smoluchowski equation (SSSE) for diffusion rate constant calculations [36] [37]. The calculated rates showed good agreement with experimental results. Our generated surface mesh is being used in calculating the electrostatic potential distribution of biomolecules.

**Tetrameric mACHe:** We also applied our approach to generate tetrahedral meshes for the tetramer, which has two different arrangement formats from four monomers according to previous crystallographic studies. Each monomer has an active site accessible though a long gorge (20 Angstrom), so there are a total of four gorges. Fig. 17 shows the two crystal structures. In the first crystal structure, two gorges are partially blocked, while the other two are completely accessible to solvent. In the second one, all the four gorges are open. Similarly, we generated fine meshes around the region of the four gorges and coarse meshes in other regions for finite element simulations [41].

**Ribosome:** Ribosomes are macromolecular complexes responsible for the translation of mRNA into protein. These complexes consist of two subunits: the large 50S and the small 30S, both of them are composed of rRNA and protein constituents. Atomic level, residual level and low resolution structures were constructed from density maps as shown in Fig. 3 and 5. The constructed



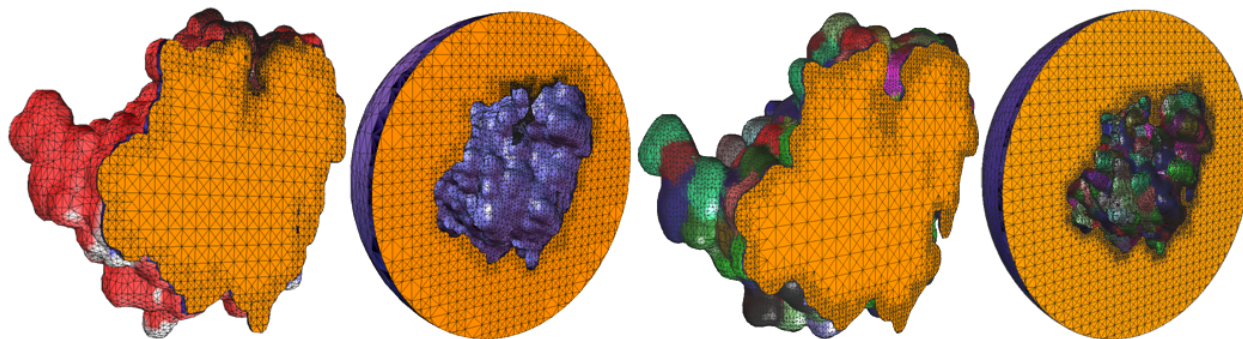


Fig. 16. Interior and exterior tetra meshes of monomeric mAChE. The left two pictures conform to the SAS with  $\sigma = 2$ , and the right two pictures conform to the surface constructed from Gaussian summation with  $p_1 = 0.25$ ,  $p_2 = 1.0$ . From left to right: (65147 vertices, 323442 tets), (121670 vertices, 656823 tets), (103680 vertices, 509597 tets) and (138967 vertices, 707284 tets). The color shows potential (leftmost) or residues (the right two).

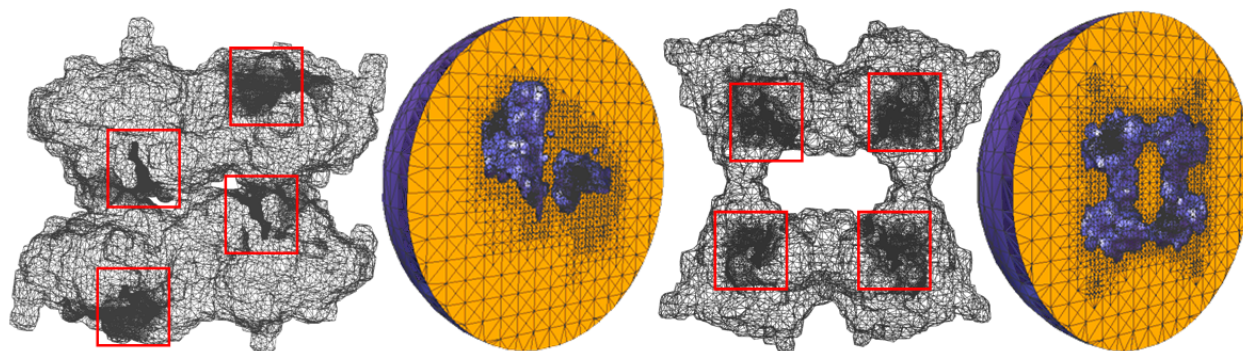


Fig. 17. Interior and exterior tetra meshes of tetrameric mAChE,  $p_1 = 0.5$ ,  $p_2 = 1.0$ . The left two pictures show the 1st crystal structure 1C2O (133078 vertices, 670950 tets), and the right two pictures show the 2nd one 1C2B, (106463 vertices, 551074 tets). Cavities are shown in red boxes.

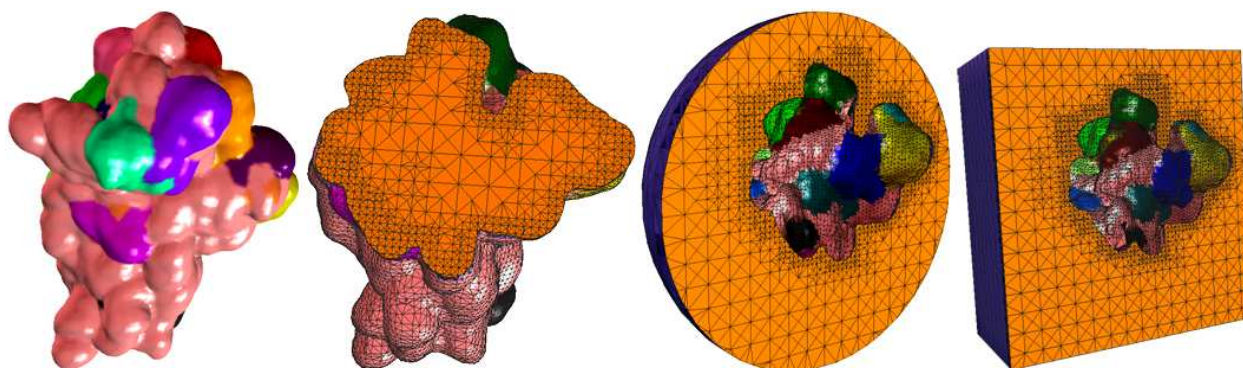


Fig. 18. Interior and exterior tetra meshes of Ribosome 30S, low resolution,  $p_1 = 0.03125$ ,  $p_2 = 1.0$ . From left to right: (33612 vertices, 163327 tets), (37613 vertices, 186496 tets) and (40255 vertices, 201724 tets). The pink color shows 16S rRNA and other colors show proteins.

various implicit solvation models help to study the machinery of protein production. Fig. 18 and Fig. 1 show interior and exterior meshes of Ribosome 30S/50S.

We have developed a quality molecular meshing approach from PDB data, including implicit solvation surface construction from electron density maps, triangular/tetrahedral mesh generation, and quality improvement with surface smoothing. Geometric features are preserved for the molecular surface. Some of our generated meshes have been used or is being used in boundary/finite element simulations.

## Acknowledgments

We wish to thank Vinay Siddavanahalli and John Wiggins for all their help with the generation of smooth volumetric electron density maps using CVC's fast summation code for biomolecular structures, and also thank Dr. Yuhua Song, Dr. Deqiang Zhang, Prof. Nathan Baker and Prof. Andrew McCammon for using and helping us calibrate our LBIE mesher for biomolecules. The work at University of Texas was supported in part by NSF-ITR grants ACI-0220037, EIA-0325550 and grants from the NIH 0P20 RR020647 and R01 GM074258. A substantial part of the work on this paper was done when Prof. Guoliang Xu was visiting Prof. Chandrajit Bajaj at UT-CVC. His work was partially supported by the aforementioned grants, the J.T. Oden ICES fellowship and in part by Natural Science Foundation of China (10371130) and National Key Basic Research Project of China (2004CB318000).

## 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, V. Siddavanahalli, and W. Zhao. Fast calculations of implicit solvation models of molecules and solvation energies. *Manuscript*, 2004.
- [3] C. Bajaj, H. Lee, R. Merkert, and V. Pascucci. NURBS based B-rep Models from Macromolecules and their Properties. In *Proceedings Fourth Symposium on Solid Modeling and Applications*, pages 217–228, 1997.
- [4] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.
- [5] S. Canann, J. Tristano, and M. Staten. An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes. In *7th International Meshing Roundtable*, pages 479–494, 1998.
- [6] H.-L. Cheng, T. Dey, H. Edelsbrunner, and J. Sullivan. Dynamic skin triangulation. *Discrete Computational Geometry*, 25:525–568, 2001.
- [7] S.-W. Cheng, T. Dey, H. Edelsbrunner, M. Facello, and S. Teng. Sliver exudation. *Proceeding Journal of ACM*, 47:883–904, 2000.

- [8] P. Chew. Guaranteed-quality delaunay meshing in 3d. In *ACM Symposium on Computational Geometry*, pages 391–393, 1997.
- [9] H. Edelsbrunner. Deformable smooth surface design. *Discrete Computational Geometry*, 21:87–115, 1999.
- [10] D. Field. Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.
- [11] L. Freitag. On combining laplacian and optimization-based mesh smoothing techniques. *AMD-Vol. 220 Trends in Unstructured Mesh Generation*, pages 37–43, 1997.
- [12] P. Frey, H. Borouchaki, and P.-L. George. Delaunay tetrahedralization using an advancing-front approach. In *5th International Meshing Roundtable*, pages 31–48, 1996.
- [13] I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima. Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):144–155, 1996.
- [14] R.R. Gabdouliline and R.C. Wade. Analytically defined surfaces to analyze molecular interaction properties. *Journal of Molecular Graphics*, 14(6):341–353, 1996.
- [15] M. Garland and P. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization*, pages 263–270, 1998.
- [16] S. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *IEEE sympos. on Volume visualization*, pages 23–30, 1998.
- [17] J. Grant and B. Pickup. A gaussian description of molecular shape. *Journal of Physical Chemistry*, 99:3503–3510, 1995.
- [18] M. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the poisson-boltzmann equation i: Algorithms and examples. *Journal of Computational Chemistry*, 21:1319–1342, 2000.
- [19] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *SIGGRAPH*, pages 339–346, 2002.
- [20] L. Kobbelt, M. Botsch, U. Schwanecke, and H. Seidel. Feature sensitive surface extraction from volume data. In *SIGGRAPH*, pages 57–66, 2001.
- [21] P. Laug and H. Borouchaki. Molecular surface modeling and meshing. *Engineering with Computers*, 18:199–210, 2002.
- [22] B. Lee and F. Richards. The interpretation of protein structures: Estimation of static accessibility. *Journal Molecular Biology*, 55:379–400, 1971.
- [23] M. Lee, M. Feig, F. Salsbury, and C. Brooks. New analytic approximation to standard molecular volume definition and its application to generalized born calculation. *Journal of Computational Chemistry*, 24:1348–1356, 2003.
- [24] R. Lohner, K. Morgan, and O. Zienkiewicz. Adaptive grid refinement for compressible euler equations. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, I. Babuska et. al. eds. Wiley, pages 281–297, 1986.
- [25] R. Lohner and P. Parikh. Three dimensional grid generation by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8:1135–1149, 1988.
- [26] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, pages 163–169, 1987.
- [27] M. Meyer, M. Desbrun, P. Schröder, and A. Burr. Discrete differential-geometry operators for triangulated 2-manifolds. *VisMath'02, Berlin*, 2002.
- [28] P. G. Mezey. *Shape in Chemistry: An Introduction to Molecular Shape and Topology*. VCH Publishers, New York, 1993.

- [29] G. Nielson and J. Sung. Interval volume tetrahedrization. In *IEEE Visualization*, pages 221–228, 1997.
- [30] S. Owen. A survey of unstructured mesh generation technology. In *7th Internat. Meshing Roundtable*, pages 26–28, 1998.
- [31] M. Sanner, A. Olson, and J. Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38:305–320, 1996.
- [32] G. Sapiro. Geometric partial differential equations and imaging analysis. *Cambridge University Press*, 2001.
- [33] M. Shephard and M. Georges. Three-dimensional mesh generation by finite octree technique. *International Journal for Numerical Methods in Engineering*, 32:709–749, 1991.
- [34] J. Shewchuk. Constrained delaunay tetrahedrizations and provably good boundary recovery. In *11th International Meshing Roundtable*, pages 193–204, 2002.
- [35] K. Shimada, A. Yamada, and T. Itoh. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. In *6th International Meshing Roundtable*, pages 375–390, 1997.
- [36] Y. Song, Y. Zhang, C. Bajaj, and N. Baker. Continuum diffusion reaction rate calculations of wild type and mutant mouse acetylcholinesterase: Adaptive finite element analysis. *Biophysical Journal*, 87(3):1558–1566, 2004.
- [37] Y. Song, Y. Zhang, T. Shen, C. Bajaj, J. McCammon, and N. Baker. Finite element solution of the steady-state smoluchowski equation for rate constant calculations. *Biophysical Journal*, 86(4):2017–2029, 2004.
- [38] S.-H. Teng and C. Wong. Unstructured mesh generation: Theory, practice and perspectives. *International Journal of Computational Geometry and Applications*, 10(3):227–266, 2000.
- [39] G. Xu. Discrete laplace-beltrami operator on sphere and optimal spherical triangulations. *Research Report, Institute of Computational Mathematics and Sciences/Engineering Computation, Chinese Academy of Sciences, No. ICM-04-11*, 2004.
- [40] G. Xu. Discrete laplace-beltrami operators and their convergence. *Computer Aided Geometry Design*, 21:767–784, 2004.
- [41] D. Zhang, J. Suen, Y. Zhang, Y. Song, Z. Radic, P. Taylor, M. Holst, C. Bajaj, N. Baker, and J. McCammon. Tetrameric mouse acetylcholinesterase: Continuum diffusion rate calculations by solving the steady-state smoluchowski equation using finite element methods. *Biophysical Journal*, 88(3):1659–1665, 2005.
- [42] Y. Zhang, C. Bajaj, and B-S Sohn. Adaptive and quality 3d meshing from imaging data. In *ACM Solid Modeling and Applications*, pages 286–291, 2003.
- [43] Y. Zhang, C. Bajaj, and B-S Sohn. 3d finite element meshing from imaging data. *Special issue of Computer Methods in Applied Mechanics and Engineering on Unstructured Mesh Generation, in press*, 2004.