Published in final edited form as:

Comput Aided Geom Des. 2011 June 1; 28(5): 307–320. doi:10.1016/j.cagd.2011.04.001.

Adaptive Skin Meshes Coarsening for Biomolecular Simulation

Xinwei Shi¹ and Patrice Koehl²

¹Genome Center, University of California, Davis, xshi@ucdavis.edu

²Department of Computer Science, and Genome Center, University of California, Davis, pkoehl@ucdavis.edu

Abstract

In this paper, we present efficient algorithms for generating hierarchical molecular skin meshes with decreasing size and guaranteed quality. Our algorithms generate a sequence of coarse meshes for both the surfaces and the bounded volumes. Each coarser surface mesh is adaptive to the surface curvature and maintains the topology of the skin surface with guaranteed mesh quality. The corresponding tetrahedral mesh is conforming to the interface surface mesh and contains high quality tetrahedral that decompose both the interior of the molecule and the surrounding region (enclosed in a sphere). Our hierarchical tetrahedral meshes have a number of advantages that will facilitate fast and accurate multigrid PDE solvers. Firstly, the quality of both the surface triangulations and tetrahedral meshes is guaranteed. Secondly, the interface in the tetrahedral mesh is an accurate approximation of the molecular boundary. In particular, all the boundary points lie on the skin surface. Thirdly, our meshes are Delaunay meshes. Finally, the meshes are adaptive to the geometry.

1 Introduction

Years of research in biology have established that all cellular functions are deeply connected to the shape and dynamics of their molecular actors, mostly bio-molecules. As a response, geometric modeling has emerged as an essential partner in structural biology and computational biology studies. For example, the computer aided geometric design community has been involved in various aspects of molecular modeling including rational drug design, surface representations of biomolecules as well as for the development of geometry-based energy functions that measure the stability of macromolecules. [3, 35, 36, 34]. In this paper, we address a specific geometric problem recurrent in many applications in computational biology, namely the problem of computing quality hierarchical meshes of skin surfaces and their bounded volumes. We note however that the methodology we propose should facilitate other applications in computer graphics such as shape deformation and physically based animations [24, 10].

Many simulations in computational biology involve solving partial differential equations (PDE) like the Poisson-Boltzmann equation (PBE) using numerical methods such as finite element methods (FEM) [22]. An essential step of these numerical solvers is to generate a well-shaped mesh M that approximate the problem domain Ω , including the boundary S , the interior I and surrounding regions O of a molecule. A subset T of M approximates the

© 2011 Elsevier B.V. All rights reserved.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

boundary S , namely, the surface mesh, separates M into two parts that approximate I and O respectively. Then the PDE is discretized into a system of linear or non-linear equations using the mesh M . Among the various numerical techniques for solving this linear system, the multigrid methods have become one of the most effective and successful techniques [23]. Multigrid methods solve a linear system by first constructing a hierarchy of meshes $M_0, \dots, M_k$, where $M_0 = M$ is the finest mesh that discretizes the problem domain Ω and M_i , $0 < i < k$, is a geometric coarsening of M_{i-1} . The solution of the linear system is then obtained and improved using two transformations based on the hierarchy of meshes, namely, the interpolation technique from mesh M_i to M_{i-1} and the restriction from M_{i-1} to M_i . The mesh quality of M_i and the accuracy at which T_i approximates the boundary S significantly influence both the accuracy of the solution and the efficiency of the multigrid methods [30].

In this paper, we present algorithms for constructing a hierarchy of high quality coarse meshes for biomolecular simulation using molecular skin meshes. In the following of this section, we first review the previous work about biomolecular mesh generation and mesh coarsening. Then we summarize our main contributions.

1.1 Related Work

High quality mesh generation for biomolecules has been a long standing problem due to the complex geometry of the molecular boundary S . There are three competing definitions for the boundary which correspond to the van der Waals surface, the accessible surface and the molecular surface [13]. The van der Waals surface is defined as the boundary of union of balls representing all the atoms as balls centered at the atomic positions with their van der Waals radius. The solvent accessible is the surface generated by the center of the probe sphere rolling on the van der Waals surface. The molecular surface is the lower envelop of the rolling probe sphere. The main difficulties of triangulating these surfaces with high quality relate to the smoothness or rather lack of smoothness of the surface models. The van der Waals surface and the solvent accessible surface are not smooth at the intersection of the spheres. The molecular surface is smooth in most cases but it contains self-intersections that results in cusps on the surface. To overcome these singularity problems, Edelsbrunner [15] introduced the molecular skin surface based on the framework of the Delaunay triangulation and Voronoi diagram of a set of weighted points. The skin model is smooth, free of self-intersection, and can be triangulated with good quality. Recently, Chavent et al [27] developed MetaMol to visualize the molecular skin surfaces with ray casting techniques. Kruithof and Vegter [25] developed an algorithm for meshing skin surfaces using a tetrahedral complex to decompose the skin into a collection of topological disks. The method depends on an assumption that all the anchor points lie inside the mixed cell, which however is generally not true for molecular models. Lu et al [26] used the existing softwares to set up a tool chain that automates the generation of tetrahedral meshes for biomolecules. Cheng and Shi [7, 8] developed efficient algorithms for triangulating the skin surface and the volumes bounded by the surface with guaranteed quality, which can be easily extended to generate the fine mesh M_0 with an accurate subset T_0 that approximates the boundary S of the molecule. Our algorithms in this paper take as input the skin surface mesh T_0 generated by the algorithm in [7] and generates a hierarchy of high quality tetrahedral meshes $M_1, \dots, M_k$ with the corresponding hierarchy of subsets $T_1, \dots, T_k$ each approximating the boundary S .

Mesh coarsening has received much attention in computational geometry and numerical analysis research. To facilitate efficient and accurate multigrid solvers, mesh quality and conformity to the problem domain of each coarse mesh M_i are two important issues in mesh coarsening algorithms. Unstructured mesh coarsening can be classified into node-nested and non-nested coarsening. A coarse mesh M_i is called node-nested if the set of vertices of M_i is a subset of that of M_{i-1} ; otherwise, it is called non-nested. Non-nested coarsening has the

advantage of introducing new vertices to guarantee the mesh quality and conformity that are degraded by vertices removal. Various approaches to non-nested coarsening differ in the methods used for choosing the vertices to remove and for placing new vertices. Miller et al [28] presented an algorithm for generating both node-nested and non-nested coarse triangular meshes in two dimensions with quality guarantees based on a spacing function specified by the fine mesh. Mesh vertices are removed according to the coarsening of the space function. A number of three dimensional mesh coarsening techniques have been proposed; in most of these techniques however, the mesh qualities are usually not guaranteed [29, 19]. A closely related topic is mesh simplification, which has been much discussed in computer graphics literature motivated by the demands of efficient rendering, storage and transmission of large three dimensional polygonal models (for a recent overview, see Botsch et al [5]). Our algorithm does not deviate significantly from these techniques: we still use basic operations such as edge contraction, edge flipping and vertex insertion; the main contribution of our approach however is that we provide guaranteed mesh quality and adaptivity.

In this paper, we will investigate non-nested mesh coarsening with guarantees on both coarse mesh quality and conformity. Our choice of mesh vertices removal and placement is driven by the geometry of the molecular boundary.

1.2 Our Contribution

Our algorithms construct a hierarchy $M_1, \dots, M_k$ of high quality tetrahedral meshes for a macromolecule using the skin surface and its fine surface mesh T_0 generated by the algorithm in [7]. We firstly generate a hierarchy of coarse surface meshes $T_1, \dots, T_k$ by coarsening the fine surface mesh T_0 with guaranteed topology and mesh quality. By taking advantages of the restricted Delaunay property and of the high mesh quality of the input surface mesh T_0 , our surface mesh coarsening procedure guarantees that the coarse mesh T_i is still a subset of the Delaunay triangulation of the surface mesh vertices. As a result, we build the corresponding tetrahedral mesh M_i from T_i by computing the Delaunay triangulation of the surface mesh vertices and identifying the subset T_i thereafter. Finally, the tetrahedral mesh quality is improved using prioritized Delaunay refinement. We note that alternatively it would be possible to build first the fine tetrahedral mesh M_0 and then to coarsen M_0 directly to generate coarse meshes $M_1 \dots M_k$; this approach however runs into the difficulties to preserve a subset T_i that approximates the boundary S .

As the size and quality of the surface mesh T_i determines those of the corresponding tetrahedral mesh M_i , we established two curvature adaptive conditions for the surface mesh using two parameters C_i and Q_i to control the resolution and quality of T_i respectively. We extended the mesh density control conditions proposed by Cheng et al [6] to bound the mesh size and quality of surface mesh T_i and derived new ranges for choosing pair of feasible values of C_i and Q_i , which is introduced in Section 3. We then enforce the conditions parameterized by C_1 and Q_1 on the surface mesh T_0 to generate T_1 . Thereafter, we generate T_i starting from T_{i-1} for all i in $[2, k]$. To enforce the conditions, we use three operations, namely, edge contraction, edge flipping, and vertex insertion. The edge contraction reduces the mesh size, the edge flipping maintains the restricted Delaunay property of the coarse mesh and the vertex insertion improves the mesh quality. Once the coarse surface meshes have been derived, we build quality tetrahedral meshes M_i for each level of the surface mesh T_i . With judiciously chosen C_i and Q_i , we guarantee that T_i is a subset of the Delaunay triangulation of the set vertices in T_i . As a result, the first step for building M_i is to compute the Delaunay triangulation of the surface vertices of T_i and identifying T_i that decompose the Delaunay triangulation into interior and exterior regions. Then, we apply the prioritized Delaunay refinement to the interior and exterior tetrahedral mesh respectively for achieving quality tetrahedral meshes M_i .

Our main contributions are:

- Formulation and implementation of the adaptive skin mesh coarsening algorithm;
- Refinement of the adaptive mesh conditions and enforcement operations for generating quality coarse surface meshes;
- Proof of constraint conditions for C_i and Q_i such that the vertex set of a coarse mesh T_i is an ε_i -sampling of the skin surfaces with their restricted Delaunay triangulation homeomorphic to the skin surface.

Our hierarchical tetrahedral meshes have a number of advantages that will facilitate fast and accurate multigrid based numerical solvers. Firstly, the quality of both the surface triangulations and tetrahedral meshes is guaranteed. Secondly, the interface in the tetrahedral mesh is an accurate approximation of the molecular boundary. In particular, all the boundary points are on the surface. Thirdly, our meshes are Delaunay meshes. Finally, the meshes are adaptive to the geometry. Our implementation verifies the high efficiency of the algorithms.

Outline—Section 2 describes previous works in the field that are relevant to the algorithms we present. Section 3 describes our surface mesh coarsening algorithm; it is followed in section 4 by the methods for generating conforming tetrahedral meshes for each level of coarse surface mesh. Section 5 illustrates our implementation through applications to meshing protein surfaces and their corresponding interior. We conclude our paper in section 6.

2 Previous Works

We review some previous results for the skin surface from [15] and its triangulation from [6] and [7]. A skin surface F_B is specified by a set of weighted points

$$B = \{b_i = (z_i, w_i) \in \mathbb{R}^3 \times \mathbb{R} \mid i=1 \cdots n\},$$

where b_i is a weighted point with z_i as its position and w_i as its weight. The skin surface F_B is a closed C^1 -continuous surface in $\mathbb{R}^3$. It consists of one or more disjoint components and each one is free of self-intersections and intersections with other components. Intuitively, the skin surface is geometrically similar to the boundary of the union of balls but with a smooth appearance by blending the spheres with quadratic patches, which lends itself to a smooth surface model for molecules. For a molecule, the weighted points b_i are the atoms whose coordinates z_i are derived either from experiments (X-ray crystallography or NMR), or from modeling. The weight w_i of atom b_i is equal to $\sqrt{2}$ times the sum of its van der Waals radius (i.e. the radius of atom in a hard-sphere model) and the radius of a probe sphere, R_p . If R_p is set to 0, the corresponding skin surface matches the van der Waals surface of the molecule, while if R_p is set to 1.4 Å (i.e. the radius of a water molecule), the skin surface mimics the accessible surface of the molecule.

In the following, we will review the skin definition and its curvature properties followed by the previous results about skin surface triangulations.

2.1 Skin Surfaces

We can consider a sphere $b_i \in B$ as the zero-set of the *weighted distance function* $\pi_i(x) = \|x - z_i\|^2 - w_i$. Following the addition and the scalar multiplication operations of the weighted distance functions $\pi_i(x)$, the *convex hull* of $X \subseteq B$ can be defined as

$$\text{conv}(X) = \left\{ \sum_{b_i \in X} \lambda_i b_i \mid \sum_i \lambda_i = 1, \forall \lambda_i \geq 0 \right\}.$$

The convex hull of X consists of an infinite family of spheres. For any sphere $b_j \in \text{conv}(X)$, we can write the expression of its center and radius as following,

$$\begin{aligned} z_j &= \sum_i \lambda_i z_i, \\ w_j &= \sum_i \lambda_i w_i + \left\| \sum_i \lambda_i z_i \right\|^2 - \sum_i \lambda_i \|z_i\|^2. \end{aligned}$$

That is, the center of b_j is in the convex hull of the centers of the spheres in X . The skin surface F_B specified by B is defined as the envelope of the infinite family of spheres in the convex hull of B after shrinking, namely,

$$F_B = \text{env}(\sqrt{\text{conv}(B)}),$$

in which the shrinking operation of a sphere b_i is defined as $\sqrt{b_i} = (z_i, w_i/2)$, and for a set of spheres X , $\sqrt{X} = \{\sqrt{b_i} \mid b_i \in B\}$. The envelope is the boundary of the union of balls in the shrinking convex hull of B . Even though the skin surface is the boundary of a family of infinite many spheres, it can be decomposed into a finite collection of quadratic patches, which enables efficient computation of the surface.

Skin Patches—A skin surface is decomposed into a collection of sphere patches and hyperboloid patches by the *mixed complex*, which is the collection of polyhedron obtained by shrinking the Minkowski sum of the Delaunay triangulation of B and its dual Voronoi complex. Each polyhedron is called a *mixed cell* and is defined as following,

$$\mu_x = (v_x \oplus \delta_x)/2,$$

in which v_x is the Voronoi cell defined by the independent subset X and δ_x is its dual Delaunay simplex. The skin patches join each other smoothly at the intersection of two mixed cells and form a closed smooth surface.

Curvature variation—The maximum principle curvature of skin surfaces is continuous and varies slowly on the surface. We denote the maximum curvature at $x \in F_B$ as $\kappa(x)$. The reciprocal $1/\kappa(x)$, that is, the radius of the maximum curvature, is called the *local length scale* at x , denoted as $\varrho(x)$. The variation of the local length scale of two points $x, y \in F_B$ can be controlled through a Lipschitz condition,

$$|\varrho(x) - \varrho(y)| \leq \|x - y\|.$$

We will use the local length scale to control the density of the surface meshes T_i .

2.2 Triangulation of skin surfaces

The triangulation of a skin surface F_B is a 2-manifold that approximates the surface accurately and maintain its topology. For a skin surface with arbitrary topology, we seek to find a set of sample points $P \subset F_B$ and a subset of the Delaunay triangulation of P that forms its triangulation. In particular, we have found that the restricted Delaunay triangulation of an ε_i -sampling of the skin surface is homeomorphic to the surface for a feasible ε value. We refer to the results from [6] and [7].

ε -sampling—A finite subset $P \subset F_B$ is an ε -sampling of F_B if for every point $x \in F_B$ there is a point $p \in P$ such that $\|x - p\| \leq \varepsilon \varrho(x)$. We call a ε -sampling P_i an ε_i -sampling if the ε has a constant value ε_i .

Restricted Delaunay triangulation—The *restricted Delaunay triangulation* R_P of $P \subset F_B$ is the subset of the triangles t of Delaunay triangulation of P , D_T , such that the dual Voronoi edge of t intersects the skin surface F_B ,

$$R_T = \{t \mid t \in D_T, F_B \cap v_t \neq \emptyset\}.$$

Homeomorphism theorem—The restricted Delaunay triangulation R_P of an ε -sampling P of F_B is homeomorphic to F_B for $0 < \varepsilon < \varepsilon_0 = 0.279$.

The theorem implies that we can build a skin surface mesh with guaranteed topology by generating an ε_0 -sampling and computing its restricted Delaunay triangulation. However, bad shaped triangles may exist in such a mesh. Cheng and Shi [7] used the restricted union of balls to generate an ε_0 -sampling with a lower bound on the distance between any two sample points. The restricted Delaunay triangulation of such a sampling is a high quality surface mesh with guaranteed topology.

Quality Skin Surface Meshes—A surface mesh T_0 generated by the algorithm [7] is homeomorphic to the input skin surface F_B . Each triangle abc in T_0 is a restricted Delaunay triangle with bounded circumradius and edge lengths, namely,

$$\begin{aligned} R_{abc} &\leq \frac{\varepsilon_0}{1-\varepsilon_0} \varrho_{abc}, \\ \|xy\| &\geq \frac{\varepsilon_0}{1+\varepsilon_0} \varrho_{xy}, xy \in \{ab, bc, ac\}, \end{aligned}$$

in which R_{abc} is the circumradius of a triangle abc and $\|xy\|$ is the distance between x and y . The local length scales ϱ_{abc} and ϱ_{xy} of a triangle abc and edge xy are defined as $\varrho_{abc} = \min\{\varrho(a), \varrho(b), \varrho(c)\}$ and $\varrho_{xy} = \max\{\varrho(x), \varrho(y)\}$ respectively.

We use the mesh T_0 as our starting mesh for generating a hierarchy of coarse surface meshes. The properties of the mesh T_0 are used to guarantee the topology and qualities of the coarse mesh. Firstly, the homeomorphism between the mesh T_0 and the skin surface F_B guarantees the topological correctness of the coarse meshes as long as we do not change the topology of the mesh during the coarsening process. Secondly, the restricted Delaunay property of T_0 provides for the possibility of guaranteeing that the coarse mesh maintains the restricted Delaunay property as well. Thirdly, the adaptivity of T_0 to the surface curvature and its high quality pave the way for adaptive coarsening.

Next, we describe the surface mesh coarsening algorithms and analyze its correctness and guarantees.

3 Skin Surface Mesh Coarsening

In this section, we first introduce the two conditions parametrized by the constants C_i and Q_i that control the mesh resolution and quality. Then, we give an overview of our surface mesh coarsening algorithms followed by a detail description of three key operations used in the algorithm. Finally, we discuss the termination of the algorithm and the constraints for choosing the value of C_i and Q_i .

3.1 Adaptive Mesh Conditions

In order to minimize the approximation error and use the least number of triangles in a coarse mesh, we adapt the triangle size to the surface curvature. In particular, we control the sizes of the triangle in the meshes to be proportional to the radius of the maximum curvature, i.e. the local length scale of the skin surface, which satisfies the Lipschitz condition. We adapt the two conditions proposed by Cheng et al [6] with different pair of values to bound the edge length and triangle size respectively in each coarse mesh T_i , namely,

$$\begin{aligned} [L_i]R_{ab} &> \frac{C_i}{Q_i}Q_{ab}, \text{ for every edge } ab, \\ [U_i]R_{abc} &< C_iQ_iQ_{abc}, \text{ for every triangle } abc. \end{aligned}$$

In Condition $[L_i]$, R_{ab} refers to the size of an edge $ab \in T_i$ which is equal to $\|ab\|/2$. Similarly, the size of a triangle $abc \in T_i$ is defined as R_{abc} , i.e. the circumradius of the triangle abc . The constant C_i controls how closely the mesh approximates the surface; we expect to increase C_i to generate coarser meshes. The constant Q_i controls the quality of the mesh and we choose values for Q_i such that the minimum angle in the triangle mesh is close to 20° . In order to guarantee that the coarse mesh T_i is the restricted Delaunay triangulation of its vertex set, C_i cannot be too large. A feasible bound will be provided in the end of this section. Another constraint for our system is the coarsening factor: our aim is to generate three levels of coarse meshes, such that the coarsest mesh has 1/10-th the number of vertices of the finest mesh. See Figure 1 for an example of such a third level coarse mesh. The values of C_i and Q_i clearly define the coarsening factor. We have judiciously chosen and implemented 3 pairs of feasible values of C_i and Q_i that results in three levels of coarse meshes with coarsening factors of 0.25, 0.125 and 0.1. In figure 1, we have used $C_3 = 0.42$ and $Q_3 = 1.414$ to get a coarse mesh with 10 times less vertices than the original skin surface mesh. The choices of values for C_i and Q_i will be justified at the end of this section.

After we set up the adaptive conditions for each coarse mesh, we enforce the conditions L_i and U_i over all the triangles in T_{i-1} to generate T_i . We use three operations, namely, edge contraction, edge flipping and vertex insertion for the enforcement. Using edge contraction, we delete the short edges that violate the condition $[L_i]$. Such a contraction however may introduce new badly shaped triangles that violate the condition $[U_i]$ and also breaks the restricted Delaunay property. We use vertex insertion operations to break the big triangles violating the condition $[U_i]$ and use edge flipping operations to restore the restricted Delaunay property locally. In the next section, we first give an overview of the algorithm. Then each of the three operations is described in detail.

3.2 Coarsening Algorithm

The input of the algorithm is a pair of parameters C_i and Q_i , the skin surface F_B and a fine surface mesh T_{i-1} . The output of the algorithm is a coarse surface mesh T_i satisfying the two conditions $[L_i]$ and $[U_i]$. The algorithm is divided into two stages, namely, initialization stage and coarsening stage. In the initialization stage, we check every triangle in the input

triangulation against the two conditions $[L_i]$ and $[U_i]$. We put all the edges violating the condition $[L_i]$ to a queue called eq and all the triangles violating the condition $[U_i]$ to a stack called ts . We note that ts is usually empty at the initialization stage since the value of $C_i Q_i$ is usually greater than that of $C_{i-1} Q_{i-1}$.

Next, we successively get short edges from eq and perform edge contractions until the eq is empty. After each edge contraction, we check the restricted Delaunay property for the newly added triangles and put the non-restricted Delaunay edges to a Stack fs for performing edge flips. We also check all the newly added edges and triangles against the conditions $[L_i]$ and $[U_i]$ and update the Queue eq and the Stack ts accordingly. The pseudo code of the coarsening algorithm CoarsenSkinMesh is illustrated below.

Algorithm 1

CoarsenSkinMesh(C_i, Q_i, F_B, T_{i-1})

1:	while eq is not empty do
2:	$ab = \text{deQueue}(eq)$;
3:	$\text{edgeContraction}(ab)$;
4:	Update fs ;
5:	$\text{flipEdges}(fs)$;
6:	Update eq and ts ;
7:	$\text{vertInsertion}(ts)$;
8:	Update fs ;
9:	$\text{flipEdges}(fs)$;
10:	end while

Edge Contraction—We successively contract the short edges until all the edges satisfy the condition $[L_i]$. To contract an edge ab , we remove the vertex with larger local length scale and all the triangles attached to that vertex. For sake of simplicity, we refer to that vertex as b . This results in a hole that is subsequently filled by a set of new triangles. In order to preserve the topology, we check the link condition before we do the edge contraction [14]. That is, an edge is contractible only if the link of edge ab is the same as the intersection of the links of its vertices a and b . The link of a vertex p in a surface mesh is defined as the collection of vertices and edges on the boundary of all the triangles attached to p . The link of an edge ab refers to the vertices of the triangles sharing edge ab other than a and b . Figure 2 illustrates a contractible edge ab on the left and a non-contractible edge ab on the right.

When an edge ab satisfies the link condition, we remove its vertices b and fill the created hole with a set of new triangles. As the input surface mesh is a two manifold, the boundary of the hole is always a three dimensional closed polygon without self intersections. We fill the hole by adding diagonal edges that maintain the manifold property.

One option for creating these new triangles is simply to snap the edges ending at vertex b to a ; this may however lead to the formation of folded triangles. Alternatively, we note that the lengths of all the edges connecting to vertex b are bounded from above by our adaptive condition $[U_i]$. This implies that all the vertices on the boundary of the three dimensional polygon are almost coplanar with vertex b since the normal and curvature of skin surface are continuous and vary slowly on the surface. As a result, we can project the polygon to the tangent plane at the point b . If the projected polygon is convex, that is, the internal angle at

each vertex is not greater than π , we can iteratively connect the diagonals of the polygon to fill the hole. If however the polygon is concave, we use the ear-clipping algorithms proposed in [21]. The vertices of the polygon is first stored along a counter clockwise orientation following the edge order of the three dimensional polygon. We then label each projected vertex as either convex or reflex. A vertex is called convex if its internal angle is less than π , and reflex otherwise. Then, we check every three consecutive vertices p^-pp^+ of the polygon whether they form an ear according to the following two conditions,

- the point p is a convex vertex;
- there are no other reflex vertex that are contained inside or that lay on the boundary of the triangle p^-pp^+ .

We iteratively form new triangles with three ear vertices and update the ears information as well. The procedure stops when only three vertices are left. We add the triangle formed by these three vertices which completes the triangulation of the concave polygon. As the newly added triangles may still violate the condition $[L_i]$ and $[U_i]$, we update the queue eq and the stack ts . Furthermore, the newly added triangles may not be restricted Delaunay any more; we remediate this problem by performing edge flipping operations.

Edge Flipping—Edge flipping restores the restricted Delaunay property of the surface mesh, which may be locally broken after an edge contraction or a vertex insertion. We flip a non-restricted Delaunay edge ac by replacing the two triangles acb and acd with the two triangles bda and bdc . Figure 3 illustrates an example of the edge flipping. In left panel of Figure 3, since the intersection point x of the two lines following the Voronoi edges of triangles acb and acd is outside the skin surface, the Voronoi edges will not intersect the skin surfaces. After replacing the edge ac by the edge bd , the intersection point x of the line corresponding to the new Voronoi edges is pushed inside the skin surface, which makes the triangles bda and bdc restricted Delaunay triangles. A symmetric situation for edge flipping exists when the skin surface is inverted.

Vertex Insertion—We add a new vertex on the surface to break a triangle abc that violates the condition $[U_i]$. We firstly locate the new vertex to be inserted on the surface. Let the circumcenter of the triangle abc be z and L be the line of points at equal distance from a , b and c . The line L intersects the skin surface at two or more points. We choose the intersection point x ; that is closest to z as the new vertex to be inserted. Then we search the triangle pqr in the mesh that intersects the line L ; pqr is either the triangle abc or one of the triangles that share an edge with abc , since the distance between z and x is necessarily small relative to the circumradius of the triangle abc . We replace the triangle pqr with three new triangles pqx , prx and qrx . Finally, we push the link edges of x into the stack fs and perform edge flipping operation until fs is empty. As a result, the violating triangle abc is destroyed either by the operation of creating 3 triangles by inserting x in abc , or by the following edge flipping operations.

3.3 Termination and Quality Guarantees

The algorithm 1 terminates and outputs a coarse surface mesh T_i satisfying the conditions $[L_i]$ and $[U_i]$, which guarantees that the minimum angle of any triangle in T_i is at least $\arcsin$

$\frac{1}{Q_i^2}$. As the coarsening procedure preserves the topology of the input surface mesh, the coarse T_i is homeomorphic to the mesh T_{i-1} . Together with the fact that the input surface T_0 is homeomorphic to the input skin surface, we can claim that each coarse surface mesh T_i is homeomorphic to the input skin surface. Moreover, we restore the restricted Delaunay

property of the surface mesh immediately after it is broken by an edge contraction or vertex insertion. As a result, every triangle in the coarse mesh T_i is a restricted Delaunay triangle.

In the rest of this section, we will first prove that the algorithm terminates. Then, we derive two lemmas that bound the values of C_i and Q_i such that there is no other restricted Delaunay triangles other than T_i in the Delaunay triangulation of all the vertices of T_i . We apply this result to build the initial tetrahedral mesh for the inside and outside of the molecule, which will be discussed in the Section 4.

Lemma 1. *The algorithm 1 terminates.*

Proof. We first prove that the sequence of edge flipping operations after each edge contraction or vertex insertion terminates. As edge flips are performed only if the two triangles are not restricted Delaunay, a flipped edge will not be flipped again. Moreover, an edge contraction or a vertex insertion creates a finite number of new non-restricted Delaunay triangles. Thus, the edge flips terminates.

Next, we prove that we cannot have infinite loops of edge contraction and vertex insertion. Firstly, the number of edge contraction is finite as at some point a putative edge contraction will violate the condition $[U_i]$. Secondly, all the new edges created by the vertex insertion operation are longer than $C_i Q_i \varrho_{abc}$ and therefore satisfy the condition $[L_i]$ and will not be contracted. As a result, the termination of the algorithm follows.

Next, we show that our coarsening algorithm decreases the mesh density while maintaining an ε_i -sampling where ε_i is specified by the value of C_i and Q_i .

Lemma 2. *The set of vertices P_i of the surface mesh T_i is an ε_i -sample of the skin surface provided that ε_i satisfies the following condition,*

$$\frac{\varepsilon_i^2}{(1 - \varepsilon_i)^2} - \frac{\varepsilon_i^4}{4} < C_i^2 Q_i^2.$$

Proof. It is obvious that the set of vertices of T_{i-1} is an ε_i -sampling since ε_i is always greater than ε_{i-1} . In order to find an upper bound for ε_i , we track the changes of ε_i starting from ε_{i-1} as we perform the coarsening operation. After an edge contraction, ε_i value gets larger; the increase however is limited, as vertex insertion ultimately increases the density and therefore decreases ε_i . At some points during the construction of T_i , there will be a point $x \in F_B$ such that the sphere of center x and radius $X = \varepsilon_i \varrho(x)$ is empty. To find the minimum value for ε_i , we assume that this sphere passes through a vertex a of a triangle abc . We then increase the radius of the sphere while keeping its center on the surface until the sphere passes through the two other vertices b and c . We denote the center of the new sphere as y and its radius is

Y . According to the curvature variation property, we get $Y = \varepsilon_i \varrho(x) \geq \frac{\varepsilon_i}{1 + \varepsilon_i} \varrho_{abc}$.

Next, We denote the circumcenter of triangle abc as z . According to the condition $[U_i]$, we

have $\|za\| = R_{abc} < C_i Q_i \varrho_{abc}$. Moreover, the distance between z and y is smaller than $\frac{\varepsilon_i^2}{2} \varrho_{abc}$.

Thus we have $Y^2 = \|yz\|^2 + \|za\|^2 \leq \frac{\varepsilon_i^4}{4} \varrho_{abc}^2 + C_i^2 Q_i^2 \varrho_{abc}^2$. Together with the lower bound of Y , we have the condition in Lemma 2.

The Lemma 2 tells us that a feasible pair of values for C_i and Q_i will enable the algorithm 1 to output a mesh T_i whose vertex set is an ε_i -sampling. We want to provide bounds for C_i

and Q_i that will ensure that the surface mesh T_i is the exact restricted Delaunay triangulation of the ε_i -sampling formed by its set of vertices. That is, there are no other restricted Delaunay triangles in the Delaunay triangulation of the ε_i -sampling. As T_i is homeomorphic to the skin surface and consists of restricted Delaunay triangles, we can validate this by proving that the restricted Delaunay triangulation of the ε_i -sampling is homeomorphic to the skin surface. This condition requires that the restricted Delaunay triangulation satisfies the closed ball property [17]. We follow the proofs in [6] and apply the conditions $[L_i]$ and $[U_i]$ to improve the bounds on ε_i . It turns out that we only need to prove a new bound for the angle between the normal of a restricted Delaunay triangle and the normal at its dual restricted Voronoi vertex, which is described in the following lemma.

Lemma 3. *Given an ε_i -sampling P_i of the skin surface F_B , if each restricted Delaunay triangle $abc \in R_{P_i}$ satisfies the conditions $[L_i]$ and $[U_i]$, then the angle between the normal n_{abc} of the triangle abc and the surface normal n_x at the restricted dual Voronoi vertex x of*

$$abc \text{ is at most } \arcsin \frac{2Q_i^3 \varepsilon_i^2}{C_i(1+\varepsilon_i)}.$$

Proof. As shown in Figure 4, the point x is the dual Voronoi vertex of the triangle abc . The solid circle represents a sphere centered at x with radius r passing through the vertices a , b and c . The two dotted circles represent the two sandwich spheres tangent to x with radius $q(x)$ centered at z_1 and z_2 . The solid sphere intersects the two sandwich spheres at two parallel disks, which are represented by the two dashed lines. Point o is the intersection of $z_1 z_2$ with the upper disk and px is a diameter of the upper sandwich sphere. Let edge bc be the longest edge and at be the shortest height. We draw a line from t that is parallel to $z_1 z_2$; it intersects the lower disk at s . As the line segment ts is parallel to the normal n_x at x and is perpendicular to the two parallel planes containing the two disks, the angle α between the

normal n_{abc} and the surface normal n_x is equal to $\angle tas$. That is, we have $\sin \alpha = \frac{\|ts\|}{\|at\|}$.

Next, we bound the lengths of ts and at to derive a bound for the angle α . As the triangle pbx

is similar to the triangle box , we have $\frac{\|px\|}{\|bx\|} = \frac{\|ox\|}{\|bx\|}$. That is, $\|ox\| = \frac{\|bx\|^2}{\|px\|} = \frac{r^2}{2q(x)}$. As a result,

$$\|ts\| = 2\|ox\| = \frac{r^2}{q(x)}.$$

Since the line segment at is the shortest height, using Heron's formula for

the triangle abc , we have $\frac{1}{2}\|bc\|\|at\| = \frac{\|bc\|\|ab\|\|ac\|}{4R_{abc}}$. As the edge length is bounded by $[L_i]$, we have

$$\|at\| = \frac{\|at\|\|ac\|}{2R_{abc}} \geq \frac{\frac{C_i^2}{Q_i^2} Q_{ab} Q_{ac}}{2C_i Q_i Q_{abc}} \geq \frac{C_i \max(Q_a, Q_b, Q_c)}{2Q_i^3},$$

As P_i is an ε_i -sampling, we have $r \leq \varepsilon_i q(x)$. Thus,

$$\sin \alpha \leq \frac{2Q_i^3 r^2}{C_i \max(Q_a, Q_b, Q_c) q(x)} \leq \frac{2Q_i^3 \varepsilon_i^2 q(x)}{C_i \max(Q_a, Q_b, Q_c)} \leq \frac{2Q_i^3 \varepsilon_i^2}{C_i(1+\varepsilon_i)}.$$

The Lemmas 2 and 3 define the constraints for choosing pairs of feasible values for C_i and Q_i such that the coarse surface T_i is the restricted Delaunay triangulation of its set of vertices. For example, we chose $C_1 = 0.245, Q_1 = 1.632$; $C_2 = 0.340, Q_2 = 1.707$ and $C_3 =$

0.424, $Q_3 = 1.414$ in our implementation to generate three levels of coarse meshes that decrease the surface mesh size with a factor of 4, 8 and 10 respectively. We can further increase the value C_i to generate coarser surface mesh T_i however the restricted Delaunay triangulation of its vertices set would be a superset of T_i . Next, we describe our tetrahedral mesh generation based on the coarse mesh T_i under the condition that T_i is the exact restricted Delaunay triangulation of its vertices set.

4 Tetrahedral Skin Mesh Generation

We describe our algorithm for generating tetrahedral meshes for each coarse surface mesh T_i in this section. We first review the existing tetrahedral mesh generation techniques and give an overview of our tetrahedral mesh generation algorithm. Then we present the extraction of the surface mesh T_i from the Delaunay triangulation of the set of vertices of the mesh T_i , and the classification of the tetrahedra in the Delaunay triangulation into interior and exterior. We introduce the prioritized Delaunay refinement to improve tetrahedral quality at last.

Constrained Delaunay triangulations (CDT) together with Delaunay refinement [11, 31, 12] are well-established techniques for generating tetrahedral meshes from surface meshes. For example, CGAL [1] provides a package for generating quality tetrahedral meshes starting from a surface mesh using Delaunay refinement. Si et al [33, 32] also developed a software called TetGen to construct high quality tetrahedral meshes for a piecewise linear complex. However, there are a few factors that encourage us to develop our own tetrahedral mesh generation algorithm and implementation. First, our skin surface meshes are consisted of a large set of vertices lying on the common spheres and robust geometric predicates, e.g., point-in-sphere test, using symbolic perturbation [16] are necessary during the construction of the Delaunay triangulation of the surface mesh vertices. While both TetGen and CGAL have robust implementations of the predicates, it is different from the implementation in our Delaunay code for generating skin surface meshes. As a result, our surface meshes are often not be considered as Delaunay by both CGAL and TetGen. Second, our Delaunay triangulation code runs more efficiently than both the CGAL and Tetgen implementations. Finally, we observed that our prioritized Delaunay refinement algorithm (see below) inserts less Steiner points for achieving same quality guarantees.

We developed our tetrahedral mesh generation algorithm on the base of the restricted Delaunay triangulation and prioritized Delaunay refinement [8]. We note that the restricted Delaunay triangulation differs from the constrained Delaunay triangulation as follows: the restricted Delaunay triangulation is the subset of the Delaunay triangulation of a set of sample points on a smooth surface that their dual Voronoi edges intersect the surface; the constraint Delaunay triangulation is a Delaunay-like tetrahedralization that is constrained to respect the boundary of an input PLC.

Overview of the Algorithm

Given a coarse surface mesh T_i generated by the algorithm 1 and the skin surface, we construct the corresponding tetrahedral mesh using three steps, namely, construction of the initial Delaunay triangulation, extraction of the interface, and Delaunay refinement. See Figure 5 for an illustration of the three steps.

Our algorithm builds an initial Delaunay triangulation of the mesh vertices of T_i using the incremental algorithm [18] at the first stage. As the coarse surface mesh has the restricted Delaunay property, we extract such a subset from the Delaunay triangulation and use it to classify the tetrahedra in the Delaunay triangulation as being either interior or exterior. Then, we insert the centers of all the atoms as well as a set of vertices defining the bounding spheres into the Delaunay triangulation. The interior and exterior Delaunay tetrahedra are

updated as well. Finally, we apply the prioritized Delaunay refinement to the interior and exterior tetrahedra separately in order to generate quality tetrahedral meshes. We present the last two stages of the algorithm in detail next.

Interface Extraction and Tetrahedra Classification

We extract T_i as the interface between interior and exterior tetrahedra from the Delaunay triangulation by labeling all the restricted Delaunay triangles. We do not have to check the restricted Delaunay property for every triangle t in the Delaunay triangulation since the circumradius of t is bounded by the condition $[U_i]$, which accelerates the extraction procedure.

Then, we use the extracted interface to classify the tetrahedra of the initial Delaunay triangulation into exterior and interior to the skin surface. In particular, all the triangles in the interface but not on the convex hull of the vertices are shared by two tetrahedra, one inside the skin surface and the other outside. We can distinguish them by comparing the vector formed by the circumcenters of the two tetrahedra to the surface normal at the intersection point of the line segment between these circumcenters with the surface. The interface triangles on the convex hull are shared by only one interior tetrahedron. As a result, all the tetrahedra attached to the surface have been labeled as either interior or exterior. We then take these labeled tetrahedra as seeds and perform a breadth first search to label all the other tetrahedra. Before we perform the Delaunay refinement to improve the mesh quality, we insert the atom centers that specify the skin surface as well as vertices on the bounding sphere into the Delaunay triangulation. Since these extra points are outside of the equatorial sphere of the triangles in the interface, the interface triangles are kept in the new Delaunay triangulation. Moreover, all the new tetrahedra attached to the atom centers are interior and these attached to a vertex on the bounding sphere are exterior. See the center figure in Figure 5 for an example.

Tetrahedral Quality Improvement

We improve the tetrahedral quality for the interior and exterior meshes separately by inserting the circumcenter of the badly shaped tetrahedra iteratively. The size of the final tetrahedral meshes will be determined by the resolution of the coarse surface meshes. In order to decrease the size of the tetrahedral mesh and keep the interface surface mesh stable in the tetrahedral mesh, our algorithm inserts the circumcenters of badly shaped tetrahedra with a priority parameterized by the value of the distance function defined by the surface [8]. That is, we insert the circumcenters furthest to the surface first and the circumcenters closest to the surface at last. As a result, the algorithm can achieve an upper bound on radius-edge ratio of the tetrahedral mesh before any points close or on the skin surface need to be inserted. All the slivers are then removed in a post processing procedure described in [9]. The right panel in Figure 6 illustrates an example of the tetrahedral mesh after performing the refinement.

5 Results

We implemented the algorithms on a Linux platform using C++ starting from the skin surface meshing software built by the authors. We generate 3 levels of coarse meshes such that the coarsest mesh decreases the number of triangles by a factor of 10. Figure 7 illustrates the input fine mesh and the three levels of output coarse surface and the interior tetrahedral meshes for the protein with PDB Id 8PTI. Table 1 lists the statistics of the four levels of surface and tetrahedral meshes. These data shows that our algorithm reduce both the number of vertices and triangles from the finest to the coarsest surface mesh by a factor of around 10. For the tetrahedral mesh, we find that the reducing factor varies for different

proteins and that they are slightly smaller than the factor for surface meshes. Moreover, all the triangles and tetrahedra in the meshes have high quality. That is, all the triangles in each level of the surface mesh have a minimum angle of at least 20° . Among them, more than 50% triangles have the minimum angle around 60° . All the tetrahedra have a radius-edge ratio of at most 2 and they are well graded according to their distance to the surface.

The surface mesh coarsening method we present here has a number of advantages compared with the existing mesh simplification algorithms. Firstly, the coarse meshes generated by our method have guaranteed quality. Second, these meshes are adaptive to the surface curvature. Finally, the method guarantees that the coarse meshes are Delaunay meshes and that they maintain the surface topology. These advantages are not only desirable for geometric modeling using polygonal meshes, they also provide better input to a tetrahedral mesh generation program. Figure 8 illustrates this property on the classical example of a Bunny. The mesh of its skin surface that contains 37,603 vertices and 75,202 faces was subjected to two independent coarsening scheme, one following the method described here and the second using the Qslim program developed by Garland [20]. The resulting coarsened meshes are shown in the center and left panels of figure 8; they both correspond to a decimation of the original mesh and include 3,962 vertices and 7,920 faces. These two meshes were subsequently used to generate Delaunay tetrahedral meshes. Starting with the surface mesh generated with our method, we obtained a tetrahedral mesh that includes 4,488 vertices and 13,931 tetrahedra; in comparison, the tetrahedral mesh generated from the surface mesh computed by Qsim using TetGen contains 8,026 vertices and 30,147 tetrahedra, for the same quality criteria, e.g., no tetrahedra have radius-edge ratio greater than 2. The main reason for this difference is that the Delaunay refinement of the tetrahedral mesh generated from the Qsim surface mesh requires more Steiner points.

Given the same input surface mesh, our method for generating quality tetrahedral meshes often achieves a smaller size mesh than the mesh generated with TetGen, with the same guarantees on the mesh quality. For example, the tetrahedral mesh inside the protein shown in figure 6 generated by our method contains 165,766 tetrahedra. In comparison, Tetgen generated a mesh with 219,459 tetrahedra to achieve the same quality guarantee, starting from the same surface mesh.

We note that it is computational costly to check if a new edge created after edge contraction or vertex insertion is restricted Delaunay or not since we need to compute the Voronoi vertex of the triangles that are deleted. In our implementation, we flip an edge if it improves the minimum angle of the two triangles; this procedure generates an *almost restricted Delaunay triangulation*, i.e. more than 95% of the triangles in the coarse mesh are restricted Delaunay and the left ones can be transformed into restricted Delaunay by edge flipping operations. Such an approximation improves the efficiency of our algorithm dramatically. For example, it takes only 39 seconds to reduce a surface mesh of 141,262 triangles to 15,252 triangles with guaranteed quality and preserved topology on a Mac PC with a 2.66GHz Dual Core Intel Xeon processor. The computing time needed for generating the coarse surface meshes versus the number of triangles in the fine surface mesh increases almost linearly as the input mesh size grows. It is noteworthy that there are no obvious relationships between the computing time and the number of the atoms because the number of triangles in the input surface mesh is a function of the surface area and the curvature of the surface, with no direct relationship to the number of atoms.

6 Conclusion

This paper presents efficient algorithms for constructing a hierarchy of surface and tetrahedral meshes with decreasing resolution for a given molecular skin surface. The mesh

resolution and quality of the coarse meshes are controlled by a pair of parameters C_i and Q_i . The coarse surface meshes are adaptive to the surface curvature and have guaranteed mesh quality and topology. The tetrahedral meshes are built for each level of coarse surface mesh using the restricted Delaunay triangulation and Delaunay refinement techniques. The tetrahedral meshes are conforming to the interface mesh which separates the tetrahedral meshes into interior and exterior. All of the tetrahedra have guaranteed quality.

An immediate future work is to apply our hierarchical tetrahedral mesh to multigrid solvers of PDE, and in particular of the PBE. We believe that using the hierarchy of meshes described here will improve the convergence of the solver as well as the accuracy of the solution since the mesh conformity and the element quality are essential factors for the accuracy and convergence of FEM. It will be interesting to investigate the application of our methods to other smooth surfaces such as implicit surfaces. The essential component in such an extension would be the curvature adaptivity conditions, which is determined by the nature of the surface.

References

1. Cgal, Computational Geometry Algorithms Library. <http://www.cgal.org>.
2. Amenta N, Choi S, Kolluri RK. The power crust, union of balls, and the medial axis transform. *International Journal of Computational Geometry and its Applications*. 2001; vol. 19:127–153.
3. Bajaj CL, Pascucci V, Shamir A, Holt RJ, Netravali AN. Dynamic maintenance and visualization of molecular surfaces. *Discrete Applied Mathematics*. 2003; 127(1):23–51.
4. Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov J, Bourne P. The protein data bank. *Nucl. Acids. Res.* 2000; 28:235–242. [PubMed: 10592235]
5. Botsch, M.; Pauly, M.; Kobbelt, L.; Alliez, P.; Lévy, B.; Bischoff, S.; Rössl, C. ACM SIGGRAPH 2007 courses. New York, NY, USA: SIGGRAPH '07, ACM; 2007. Geometric modeling based on polygonal meshes.
6. Cheng H, Dey TK, Edelsbrunner H, Sullivan J. Dynamic Skin Triangulation. *Discrete Comput. Geom.* 2001; 25:525–568.
7. Cheng H, Shi X. Quality mesh generation for molecular skin surfaces using restricted union of balls. *Proceedings of IEEE Visualization*. 2005:399–405.
8. Cheng H-L, Shi X. Quality tetrahedral mesh generation for macromolecules. *ISAAC*. 2006:203–212.
9. Cheng, S-W.; Dey, TK.; Edelsbrunner, H.; Facello, MA.; Teng, S-H. Sliver exudation. *Proceedings of the fifteenth annual symposium on Computational geometry*; ACM Press; 1999. p. 1-13.
10. Chentanez, N.; Feldman, BE.; Labelle, F.; O'Brien, JF.; Shewchuk, JR. Liquid simulation on lattice-based tetrahedral meshes. *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*; SCA '07; 2007. p. 219-228.
11. Chew, LP. Constrained delaunay triangulations. *SCG '87: Proceedings of the third annual symposium on Computational geometry*; ACM Press; New York, NY, USA. 1987. p. 215-222.
12. Cohen-Steiner, D.; de Verdiere, EC.; Yvinec, M. Conforming delaunay triangulations in 3d. *Proceedings of the eighteenth annual symposium on Computational geometry*; ACM Press; 2002. p. 199-208.
13. Connolly ML. Molecular surfaces: A review. *Network Science*. 1996
14. Dey T, Edelsbrunner H, Guha S, Nekhayev D. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd) (N.S.)*. 1999; 66:23–45.
15. Edelsbrunner H. Deformable smooth surface design. *Discrete Comput. Geom.* 1999; 21:87–115.
16. Edelsbrunner, H.; Mücke, EP. *ACM Trans. Graph.* Vol. vol. 9. ACM Press; 1990. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms; p. 66-104.
17. Edelsbrunner, H.; Shah, N. Triangulating topological spaces; *Proceedings 10th ACM Symposium on Computational Geometry*; 1994. p. 285-292.

18. Edelsbrunner, H.; Shah, NR. Incremental topological flipping works for regular triangulations; Symposium on Computational Geometry; 1992. p. 43-52.
19. Feng YT, Peric D. Coarse mesh evolution strategies in the galerkin multigrid method with adaptive remeshing for geometrically non-linear problems. *International Journal for Numerical Methods in Engineering*. 2000; 49:547–571.
20. Garland M, Heckbert PS. Surface simplification using quadric error metrics. *SIGGRAPH*. 1997:209–216.
21. Held M. FIST: Fast industrial-strength triangulation of polygons. *Algorithmica*. 2001; 30(4):563–596.
22. Holst M, Baker NA, Wang F. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation i. algorithms and examples. *J. Comp. Chem*. 2000; 21(15):1319–1342.
23. Holst MJ, Saied F. Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods. *J. Comp. Chem*. 1995; 16(3):337–364.
24. Kobbelt LP, Bareuther T, Seidel H-P. Multiresolution shape deformations for meshes with dynamic vertex connectivity. *Computer Graphics Forum*. 2003; 19(3):249–260.
25. Kruithof N, Vegter G. Meshing skin surfaces with certified topology. *Comput. Geom*. 2007; 36(3):166–182.
26. Lu B, Zhou YC, Huber GA, Bond SD, Holst MJ, McCammon JA. Electrodifffusion: A continuum modeling framework for biomolecular systems with realistic spatiotemporal resolution. *J. Chem. Phys*. 2007 Oct.127 5102–+
27. Matthieu, Chavent; Bruno Levy, BM. Metamol: High-quality visualization of molecular skin surface. *Journal of Molecular Graphics and Modelling*. 2007; 27:209–216.
28. Miller; Talmor; Teng. Optimal good-aspect-ratio coarsening for unstructured meshes; SODA: ACM-SIAM Symposium on Discrete Algorithms; 1997.
29. Ollivier-Gooch C. Coarsening unstructured meshes by edge contraction. *International Journal for Numerical Methods in Engineering*. 2003; 57:391–414.
30. Pebay, PP.; J.Baker, T. A comparison of triangle quality measures; Proceedings 10th International Meshing Roundtable; 2001 Oct. p. 327-340.
31. Shewchuk, JR. PhD thesis. Vol. 22. Carnegie-Mellon University; 1997. Delaunay refinement mesh generation.
32. Si, H. On refinement of constrained delaunay tetrahedralizations; Proceedings of the 15th International Meshing Roundtable; 2006.
33. Si, H.; Gaertner, K. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations; Proceedings of the 14th International Meshing Roundtable; 2005. p. 147-163.
34. Wolters HJ. Geometric modeling applications in rational drug design: a survey. *Computer Aided Geometric Design*. 2006; 23(6):482–494.
35. Zhang Y, Xu G, Bajaj C. Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design*. 2006; 23(6):510–530. [PubMed: 19809581]
36. Zomorodian A, Guibas L, Koehl P. Geometric filtering of pair-wise atomic interactions applied to the design of efficient statistical potentials. *Computer Aided Geometric Design*. 2006; 23(6):531–544.

Fig. 1. Molecular skin surface meshes for a protein. A fine mesh and a coarse mesh of the surface of a protein (PDB ID 15C8) are showed in the top row and bottom row respectively. The left panels show the whole surface meshes while the right panels show a zoomed in view of the partial mesh corresponding to the red box in the left. The fine surface mesh contains 141,262 triangles while the coarse mesh contains 15,252 triangles. All the triangles in the mesh have a minimum angle of at least 31.4° .

$$\text{link}(a) \cap \text{link}(b) = \text{link}(ab) = \{c, d\}$$

$$\text{link}(a) \cap \text{link}(b) = \{c, d, ec\}$$

$$\text{link}(ab) = \{c, d\}$$

Fig. 2.

Examples of a contractible edge (left) and a non-contractible edge (right). The link of vertices a , b and edge ab are drawn in blue, red and green color respectively.

Fig. 3. Illustration of an edge flipping operation. The two non-restricted Delaunay triangles acb and acd on the left figure are replaced by triangles abd and cbd in the right. l_t , $t \in \{a, b, c, d\}$ is the normal of the triangle with t as one of the vertices passing its circumcenter. x is the intersection of two lines. The right panel is rotated by 90° with respect to the left panel for clarity

Fig. 4. Illustration of the angle between the normal of a restricted Delaunay triangle and the normal at its restricted Voronoi vertex.

Fig. 5.

Illustration of the tetrahedral mesh generation algorithm with a 2D example. The left figure shows the initial Delaunay triangulation of the sample points on the skin curve and the red polygonal curve is the extracted interface between the interior triangles such as those two in blue and exterior triangles such as the green one. The middle figure shows the Delaunay triangulation after inserting the centers (blue) and the vertex on the bounding circle (black). The right figure shows the Delaunay triangulation after performing refinement on the interior triangles.

Fig. 6. Tetrahedral skin mesh of a protein. The coordinates for the centers of the atoms of the protein are taken from the PDB [4] file 9HVP. The left panel shows a partial tetrahedral mesh obtained by cutting the original mesh by a plane. The right panel shows a zoomed in view of the center part of the left panel. Red tetrahedra are inside the molecule while green tetrahedra are outside.

Fig. 7. Hierarchical skin surface and tetrahedral meshes for a protein (PDB ID 8PTI). The top row shows the 4 levels of surface meshes while the bottom row shows the interior tetrahedral meshes using a cutting plane.

Fig. 8. A comparison of our skin mesh coarsening algorithm with Qslim

The left panel shows the mesh generated for a Bunny that has been modeled with a skin surface consisting of 37,603 vertices and 75,202 faces. The skin model of the bunny are specified by a set of spheres output from the PowerCrust[2] with surface sample points of a Stanford Bunny model. The center and right panels show the corresponding coarse meshes generated by our adaptive coarsening algorithm and by Qslim [20], respectively. These two coarsened meshes include the same number of vertices (3,962) and the same number of faces (7,920).

Table 1

The statistics of the surface and tetrahedral mesh data at four levels of resolution. The number of vertices, triangles and tetrahedra of each level are listed in the table for 5 proteins.

PID	Level 0	Level 1	Level 2	Level 3
Surface Mesh				
	# Vert # Tri	# Vert # Tri	# Vert # Tri	# Vert # Tri
1APH	10,137 20,270	2,359 4,714	1,244 2,484	994 1,984
8PTI	27,036 54,068	6,475 12,946	3,492 6,980	2,871 5,738
9HVP	35,674 71,344	8,525 17,046	4,616 9,228	3,839 7,674
15C8	70,623 141,262	16,991 33,998	9,186 18,388	7,618 15,252
4GRT	76,740 153,480	18,420 36,840	9,934 19,868	8,255 16,510
Tetrahedral Mesh				
	# Vert # Tet	# Vert # Tet	# Vert # Tet	# Vert # Tet
1APH	15,741 93,145	4,621 28,502	2,639 16,167	2,113 12,770
8PTI	40,866 241,846	11,492 71,063	6,336 39,229	4,986 30,611
9HVP	54,719 325,198	15,937 99,307	9,295 58,021	7,547 47,025
15C8	107,884 641,898	31,493 197,393	18,138 114,230	14,722 92,398
4GRT	117,444 698,917	34,071 213,607	19,703 124,144	15,990 100,489