Available online at www.sciencedirect.com
ScienceDirect
Journal of Molecular Graphics and Modelling 26 (2008) 1370–1380
Journal of Molecular Graphics and Modelling
www.elsevier.com/locate/JMGM

Feature-preserving adaptive mesh generation for molecular shape modeling and simulation

Zeyun Yu [a,*], Michael J. Holst [a], Yuhui Cheng [b], J.Andrew McCammon [b]

[a] Department of Mathematics, University of California, San Diego, La Jolla, CA 92093, United States
[b] Department of Chemistry & Biochemistry and Department of Pharmacology, University of California, San Diego, La Jolla, CA 92093, United States

Received 17 October 2007; received in revised form 22 January 2008; accepted 27 January 2008
Available online 7 February 2008

Abstract

We describe a chain of algorithms for molecular surface and volumetric mesh generation. We take as inputs the centers and radii of all atoms of a molecule and the toolchain outputs both triangular and tetrahedral meshes that can be used for molecular shape modeling and simulation. Experiments on a number of molecules are demonstrated, showing that our methods possess several desirable properties: feature-preservation, local adaptivity, high quality, and smoothness (for surface meshes). We also demonstrate an example of molecular simulation using the finite element method and the meshes generated by our method. The approaches presented and their implementations are also applicable to other types of inputs such as 3D scalar volumes and triangular surface meshes with low quality, and hence can be used for generation/improvement of meshes in a broad range of applications.

Keywords: Mesh generation; Molecular simulation; Molecular shape modeling; Finite element method; Numerical analysis

1. Introduction

In addition to providing a geometric representation of an object for graphical purposes, mesh generation is also in great demand in numerical simulation using finite/boundary element methods and has been extensively studied in both applied mathematics [7] and computational engineering [1,27]. Although different types of meshes may be generated depending on the numerical solvers being employed, we restrict ourselves in this paper to triangular (surface) and tetrahedral (volumetric) meshes. In particular, we consider mesh generation for molecular applications, namely, meshes that are generated from a set of centers and radii of atoms in a molecule and are then used for solving various types of partial differential equations (PDE) arising in molecular modeling.

Molecular mesh generation requires good approximation of molecular surfaces. There are two primary ways to construct such surfaces: one is based on the ''hard sphere'' model [36] and the other is based on the level set of a ''soft'' or smooth function [23]. In the first model, a molecule is treated as a list of ''hard'' spheres with different radii, from which three types of surfaces can be extracted. The van der Waals surface (or *envelope*) is defined as the union of the spheres with their intersecting portions removed. An alternative way of defining molecular surfaces is by rolling a sphere over the van der Waals surface, where the loci of the probing sphere gives rise to the so-called solvent accessible surface (SAS) [31]. The radius of the probe sphere is chosen as the size of the solvent molecules (usually water). Another widely used surface is known as solvent excluded surface (SES), which is defined as the ''inward-facing'' part of the probing sphere as it rolls over the molecules [18,23,36]. The molecular surface can be represented analytically by a list of seamless surface patches [4,16,43] and triangular meshes can be generated using such tools as MSMS [37].

In contrast, the ''soft'' model treats each atom typically as a Gaussian-like smoothly decaying scalar function in $\Re^3$ that approximates certain characteristic function of the atom [12,20,23]. The molecular surfaces, including the van der Waals surface, SAS and SES, are then approximated by appropriate level sets (or isocontours) [12,20]. In addition to the Gaussian function, some other functions such as piecewise

* Corresponding author.
E-mail address: zeyun.yu@gmail.com (Z. Yu).

1093-3263/$ – see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmgm.2008.01.007

polynomial splines could also be used to approximate the characteristics of atoms [28,34,38]. Once a volumetric function is computed, the molecular surfaces are extracted and triangulated using iso-contouring techniques such as the marching cube [33] and the dual contouring method [30]. The triangulated surface mesh can be better represented by NURBS [5] or algebraic spline models [54]. Another more recent approach to molecular surface generation, based on a "soft" model, was described in [9], where the molecular surface was given by minimizing the surface free energy.

A good mesh in general should have the following properties: (1) feature-preserving, (2) adaptivity, and (3) high quality. The first two properties require that the mesh generation should capture the important features of a molecule and have the control of producing dense meshes at regions of interest and coarse meshes elsewhere. The quality of a mesh can be measured by either geometry-dependent [32] or solution-dependent [11] criteria. Some quantities based on specific biochemical applications, such as the sensitivity of a mesh to small atomic displacements [8,44], may also be considered as an indicator to the mesh quality. The one we use in our study is by measuring the angles of the triangulated molecular surfaces, a strategy commonly used in the mesh generation and smoothing community [46,55]. Specifically we try to maximize the minimal angle and minimize the maximal angle in a triangular mesh, so that the resulting angle distribution (or histogram) would be centering as much as possible near 60°. The mesh quality should be guaranteed in molecular modeling and simulation due to the fact that "skinny" triangles often cause poor approximation quality in finite/boundary element methods [27]. The molecular mesh generation approaches mentioned above can usually capture the features of a molecule (although the definition of a feature may vary), but they have either no mesh adaptation (e.g., [16]) or very low mesh quality (e.g., [9,37]).

While almost all of the methods discussed above were developed for molecular surface generation, there are increasing demands for volumetric meshing methods/tools in the biomolecular modeling community, especially when finite element numerical approaches are taken into consideration. LBIE [52] is one of very few existing tools that can generate volumetric meshes for biomolecular applications. As a Gaussian-based "soft" model, it integrates the surface triangulation, tetrahedron generation and smoothing into a stand-alone software package, reading PDB files or 3D volumes as its inputs. An octree-based data structure is employed to construct both the surface and volumetric meshes that preserve important geometric features and show adaptive triangles (tetrahedra) as well. While the qualities of the interior tetrahedral meshes generated by LBIE are usually good, there are many "sharp" triangles on the surfaces due to the iso-contouring method used in LBIE, and hence many tetrahedra near the surfaces can be poorly shaped. A new version of LBIE has been introduced in [53] by applying surface and tetrahedral mesh smoothing techniques (such as edge contraction and weighted averaging). However, the tetrahedral mesh smoothing turns out to be slow and sometimes the algorithm fails.

In the present paper we describe a new mesh generation framework in which we aim to produce feature-preserving, adaptive, and high quality surface (triangular) and volumetric (tetrahedral) meshes for molecular modeling and other applications. Similar to LBIE [52], our method is based on the level set of a Gaussian kernel function that approximates the molecular surface of a given molecule. We use the marching cube method to extract iso-surfaces; hence our method can deal with volumes of arbitrary sizes, unlike LBIE that requires the volume sizes be $2^n + 1$ for an integer n. An angle-based approach is adapted to improve the mesh quality while retaining the features and smoothness of molecular surfaces. In our toolchain we also develop an adaptive mesh coarsening technique, by which the resulting meshes are made finer in regions of high curvatures or by some other user-specified criteria and coarser elsewhere. Along with the mesh coarsening, a normal-based surface mesh smoothing technique is employed to guarantee the smoothness of the meshes while they are being coarsened. Once a good surface mesh is generated, the tetrahedra (interior and exterior of the molecular surfaces) are constructed using the constrained Delaunay triangulation as implemented in *Tetgen* [39,40].

2. Methods

In this section we describe the algorithms that we use to construct the triangular surface and tetrahedral volumetric meshes. For the particular purpose of molecular modeling, the inputs to our toolchain are a list of centers and radii of atoms (e.g., PQR files [19] or PDB files [10] with radii defined by users), although our tool can also read as inputs an arbitrary 3D scalar volume or a triangulated surface mesh with very low quality. Fig. 1 shows the pipeline of our mesh generation toolchain. In the following subsections, we shall explain each step in detail.

2.1. Initial surface meshing

In our mesh generation methods, the initial surface mesh is defined by the level set of the Gaussian kernel function that is

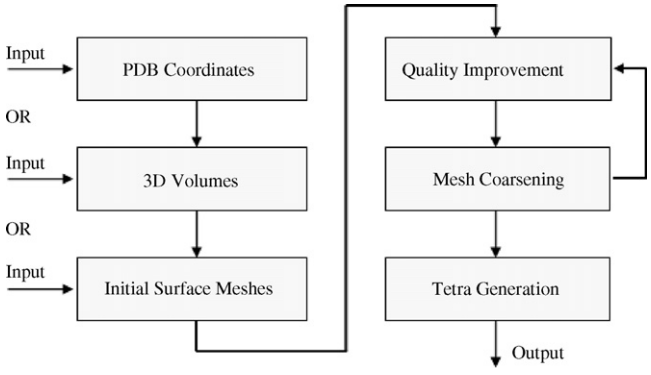

Fig. 1. Illustration of our mesh generation toolchain. The inputs can be a list of atoms (with centers and radii), a 3D scalar volume, or a user-defined surface mesh. The latter two can be thought of as special cases of the first one.

computed from a list of atoms with centers ($\mathbf{c}_i$) and radii (r_i) as follows:

$$F(\mathbf{x}) = \sum_{i=1}^{N} e^{B_i((|\mathbf{x}-\mathbf{c}_i|^2)/r_i^2-1)}, \tag{1}$$

where the negative parameter B_i is called the *blobbyness* that controls the spread of the characteristic function of each atom. We usually treat the blobbyness as a constant parameter (denoted by B_0) for all atoms. When B_0 goes to zero, $F(\mathbf{x})$ becomes constant and all features disappear. Our experiments on a number of molecules show that a blobbyness of -0.5 produces a good resolution for molecular simulations.

An alternate way of defining the Gaussian kernel function is by putting the blobbyness and radius of each atom together as a constant, as formulated in the following Eq. [53]:

$$G(\mathbf{x}) = \sum_{i=1}^{N} \mathrm{e}^{\kappa(\|\mathbf{x}-\mathbf{c}_i\|^2-r_i^2)}, \tag{2}$$

where $\kappa = (B_i/r_i^2)$ is a constant for all atoms $i = 1, \ldots, N$. From a computational point of view, the formulation in Eq. (2) might be more beneficial due to the constant decay rate κ that may lead to a fast Gaussian summation using the recursive scheme as described in [48]. We shall discuss this strategy in detail in Section 4. But for all examples shown below, the Gaussian functions are calculated by $F(\mathbf{x})$ as defined in Eq. (1).

Once the Gaussian kernel function is computed and summed up for all atoms, the molecular surface is then defined by the level set as $F(\mathbf{x}) = t_0$, where t_0 is the isovalue [12,23,53]. The surface (triangular) mesh can be constructed by either one of two iso-contouring techniques – the marching cube method [33] and the dual contouring method [30]. In our mesh generator we employ an implementation of the marching cube method. Fig. 2(a) shows an example of the isosurface extracted using this method. To illustrate the details better, shown here is only a small portion of the molecular surface of the domains 3/4 of rat CD4 (PDB-ID: 1CID). From this example, we can see that: (1) the iso-contouring technique can extract very smooth surfaces, but (2) many triangles are extremely "sharp".

2.2. Surface mesh quality improvement

The mesh quality can be improved by a combination of three major techniques: inserting/deleting vertices, swapping edges/faces, and moving the vertices without changing the mesh topology [22]. The last one is the main strategy we use to improve the mesh quality in our methods. For surface meshes, however, moving the vertices may change the shape of the surface. Therefore, two other criteria should also be taken into account: (1) important features (e.g., sharp boundaries, concavities, holes, etc.) on the original surface should be preserved as much as possible, and (2) the surface should be kept smooth while the vertices are moved.

To characterize the important features on a surface mesh, we compute so-called *local structure tensor* [21,45,50,51] as follows:

$$T(\mathbf{v}) = \sum_{i=1}^{M'} \begin{pmatrix} n_x^{(i)}n_x^{(i)} & n_x^{(i)}n_y^{(i)} & n_x^{(i)}n_z^{(i)} \\ n_y^{(i)}n_x^{(i)} & n_y^{(i)}n_y^{(i)} & n_y^{(i)}n_z^{(i)} \\ n_z^{(i)}n_x^{(i)} & n_z^{(i)}n_y^{(i)} & n_z^{(i)}n_z^{(i)} \end{pmatrix}, \tag{3}$$

where $(n_x^{(i)}, n_y^{(i)}, n_z^{(i)})$ is the normal of the ith neighbor of a vertex $\mathbf{v}$ and M' is the total number of neighbors (Note: the neighbors considered may be more than the incident ones of a vertex. They could extend by several "layers" and the number of the "layers" considered depends on the size of the local features). The normal of a vertex is defined by the weighted average of the normals of all its incident triangles. The idea of the local structure tensor is derived from the well-known principal component analysis (PCA) technique [29]. It basically captures the principal orientations of a set of vectors in space. Let the eigenvalues of Eq. (3) be $\lambda_1, \lambda_2, \lambda_3$ and $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then the local structure tensor can capture the following features: (a) Spheres and saddles: $\lambda_1 \approx \lambda_2 \approx \lambda_3 > 0$; (b) Ridges and valleys: $\lambda_1 \approx \lambda_2 \gg \lambda_3 \approx 0$; (c) Planes: $\lambda_1 \gg \lambda_2 \approx \lambda_3 \approx 0$.

As we mentioned earlier, the quality of a mesh can be improved by maximizing the minimal angles. Let $\mathbf{x}$ be the vertex that we want to move for quality improvement, and the (incident) neighbors be orderly denoted by $\mathbf{v}_i, i = 1, \ldots, M$, where M is the total number of the neighbors. The idea of the angle-based method described in [55] is to move $\mathbf{x}$ to a new

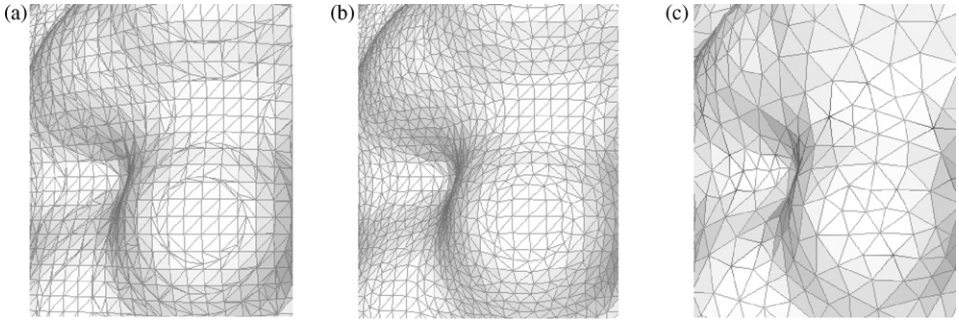

Fig. 2. Illustration of the surface generation and post-processing. (a) A 3D volume is first generated using the Gaussian blurring approach (Eq. (1)) from the molecule (PDB: 1CID). Shown here is part of the surface triangulation by the marching cube method. (b) The surface mesh after two iterations of running the quality improvement algorithm. (c) After coarsening, the mesh size becomes about seven times smaller than the original one. The mesh is also smoothed by a normal-based technique.

position $\bar{\mathbf{x}}$ such that $\bar{\mathbf{x}}$ maximizes the angles $\angle(\mathbf{x}, \mathbf{v}_i, \mathbf{v}_{i-1})$ and $\angle(\mathbf{x}, \mathbf{v}_{i-1}, \mathbf{v}_i)$ for $i = 1, \ldots, M$ with $\mathbf{v}_0 := \mathbf{v}_M$. Let $\mathbf{x}_i$ be the projection of $\mathbf{x}$ to the bisector of the angles $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$ for $i = 1, \ldots, M$ with $\mathbf{v}_{M+1} := \mathbf{v}_1$ (see Fig. 3), then $\bar{\mathbf{x}}$ is approximated in [55] by $\bar{\mathbf{x}} = (\mathbf{x}_1 + \mathbf{x}_2 + \cdots, +\mathbf{x}_M)/M$. This method has been extended in [46] for improving quadrilateral mesh quality, but only two-dimensional meshes were discussed in both papers.

We make some modifications of the 2D angle-based method as described in [55], in order to deal with the surface mesh processing. First, $\mathbf{x}$ is projected onto the bisecting plane, instead of the bisecting line, of $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$, as shown in Fig. 3. Secondly, the average of $\mathbf{x}_i$ is weighted by a decreasing function of the angle $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$:

$$\bar{\mathbf{x}} = \frac{1}{\sum_{i=1}^{M}(d_i + 1)} \sum_{i=1}^{M} (d_i + 1)\mathbf{x}_i, \tag{4}$$

where d_i is the dot product of the normalized vectors of $(\mathbf{v}_{i-1} - \mathbf{v}_i)$ and $(\mathbf{v}_{i+1} - \mathbf{v}_i)$. The use of weighted average is due to the fact that a smaller angle of $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$ is more sensitive to the change of $\bar{\mathbf{x}}$.

The above two modifications are usually sufficient in practice to improve the 2D/3D mesh quality. However, as we mentioned earlier, dealing with surface meshes requires additional care – the geometric features on the surface mesh should be preserved as much as possible. To this end, we take the advantage of the local structure tensor by mapping the new position $\bar{\mathbf{x}}$ to each of the eigenvectors of the tensor calculated at the old position $\mathbf{x}$ and scaling the mapped vectors with the corresponding eigenvalues. Let $\vec{e}_1, \vec{e}_2, \vec{e}_3$ denote the eigenvectors and $\lambda_1, \lambda_2, \lambda_3$ be the corresponding eigenvalues of the local structure tensor valued at $\mathbf{x}$. The modified vertex $\hat{\mathbf{x}}$ is calculated as follows:

$$\hat{\mathbf{x}} = \mathbf{x} + \sum_{k=1}^{3} \frac{1}{1 + \lambda_k} ((\bar{\mathbf{x}} - \mathbf{x})\vec{e}_k)\vec{e}_k. \tag{5}$$

The use of the eigenvalues as a weighted term in the above equation is essential in preserving the features (with high curvatures) and to keep the improved surface mesh as close as possible to the original mesh by encouraging the vertices to move along the eigen-direction with small eigenvalues (or in other words, with low curvatures). Fig. 2(b) shows the surface mesh with quality improved, compared to the original mesh as shown in Fig. 2(a).

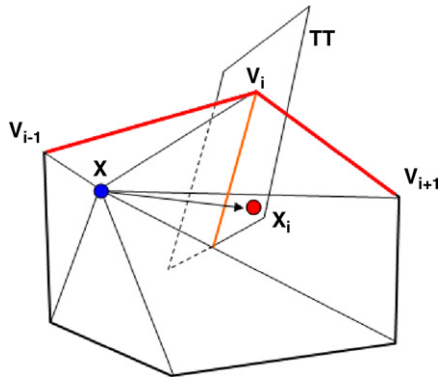

Fig. 3. Illustration of the angle-based quality improvement algorithm.

2.3. Surface mesh coarsening and smoothing

The surface meshes extracted by iso-contouring techniques (e.g., the marching cube method) often contain a large number of elements and are nearly uniform everywhere. To reduce the computational cost, adaptive meshes are usually preferred where fine meshes only occur in regions containing features. The idea of mesh coarsening is straightforward – choosing a node to delete and re-triangulating the region containing the incident neighbors. The local structure tensor is again used as a way to quantify the features. Let $\mathbf{x}$ denote the node being considered for deletion and the neighboring nodes be $\mathbf{v}_i, i = 1, \ldots, M$, where M is the total number of the neighbors. The maximal length of the incident edges at $\mathbf{x}$ is denoted by $L(\mathbf{x}) = \max_{i=1}^{M}\{d(\mathbf{x}, \mathbf{v}_i)\}$ where $d()$ is the Euclidean distance. Apparently $L(\mathbf{x})$ indicates the sparseness of the mesh at $\mathbf{x}$. Let $\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x}), \lambda_3(\mathbf{x})$ be the eigenvalues of the local structure tensor calculated at $\mathbf{x}$, satisfying $\lambda_1(\mathbf{x}) \geq \lambda_2(\mathbf{x}) \geq \lambda_3(\mathbf{x})$. Then the node $\mathbf{x}$ is deleted if and only if the following condition holds:

$$L(\mathbf{x})^{\alpha} \left(\frac{\lambda_2(\mathbf{x})}{\lambda_1(\mathbf{x})} \right)^{\beta} < T_0, \tag{6}$$

where α and β are chosen to balance between the sparseness and the curvature of the mesh. In our experiments, they both are set as 1.0 by default. The threshold T_0 is user-defined and also dependent on the values of α and β chosen. When α and β are fixed, larger T_0 will cause more nodes to be deleted.

Mesh coarsening can greatly reduce the mesh size to a user-specified order. However, the nodes on the "holes" are often not co-planar; hence the re-triangulation of the "holes" often results in a bumpy surface mesh. The bumpiness can be reduced or removed by smoothing the surface meshes. We employ the idea of anisotropic vector diffusion [35,49] and apply it to the normals of the surface mesh being considered. This normal-based approach turns out to preserve sharp features and prevent volume shrinkages [15,47] better than the traditional vertex-based approach. Fig. 2(c) shows the result after the mesh coarsening and normal-based mesh smoothing.

2.4. Tetrahedron generation

Tetgen [39,40] is an outstanding software tool that can generate high-quality tetrahedral meshes from a triangulated surface, using constrained Delaunay triangulation. However, the input surface mesh must have decent quality; otherwise, *Tetgen* would insert a large number of additional nodes or fail. Our surface mesh generation and post-processing algorithms described above provide meshes with quality high enough to be used as inputs in *Tetgen* to produce tetrahedral meshes. Besides the triangulated surface, our toolchain has three other outputs for a given molecule: the interior tetrahedral mesh, the exterior

tetrahedral mesh, and both meshes together. For the interior tetrahedral mesh, we force all atoms of the given molecule to be on the mesh nodes. The exterior tetrahedral mesh is generated between the surface mesh and a bounding sphere whose radius is usually set as about 40 times larger than the size of the molecule being considered. The size of a molecule is defined as follows. We first compute the average of all atom centers; then the maximal distance between the atoms to the averaged center is defined as the size of the molecule.

3. Results

In this section we show the mesh generation results on a number of molecules. The atom coordinates are extracted from the PDB files [10] and the radius for each atom is given in a user-specified table. Our program can also read the PQR format in which the radius information is attached to each atom [19].

3.1. Surface mesh generation and post-processing

All examples shown here are generated using the following parameters, and the initial surface meshes are coarsened only once unless otherwise specified. The blobbyness of the Gausian kernel function is set as −0.5. All 3D volumes are discretized based on the spacing distance of 0.5 Å per grid. The initial molecular surfaces are extracted using the marching cube method [33] at the isovalue of 1.0. The threshold T_0 for the mesh coarsening in Eq. (6) is set as 0.2.

The first example that we demonstrate here is domains 3/4 of rat CD4, a small molecule of 1381 atoms, taken from the Protein Data Bank (PDB-ID: 1CID). A volume of 130 × 132 × 176 grids is generated using the Gaussian-based blurring technique as described in Section 2.1. Part of the initial surface was shown in Fig. 2(a). The exterior surface view of the entire molecule after quality improvement is shown in Fig. 4(a). There are 62,454 nodes and 124,904 triangles in this mesh. After mesh coarsening and normal-based smoothing followed by another round of quality improvements, the mesh size is reduced to 8846 nodes and 17,688 triangles. The mesh quality is also greatly improved, from 0.02° (minimal angle) and 179.10° (maximal angle) for the initial mesh, to 20.22° (minimal angle) and 130.71° (maximal angle) for the processed mesh. Fig. 4(b) and (c) shows the exterior and interior views of the processed mesh. Thanks to the feature-based adaptivity in our meshing tools, most important features in the original mesh are well preserved in spite of a significant size reduction. Note that we can adjust the blobbyness parameter in Eq. (1) to generate molecular surfaces at different ''resolutions''. Fig. 4(d) shows the surface of 1CID when the blobbyness is chosen as −2.5.

Another example we investigated is the mouse acetylcholinesterase (mAChE) molecule, which has a total of 8362 atoms yielding a volume of 162 × 142 × 192grids. The initial mesh contains 96,046 nodes and 192,088 triangles and is reduced to 19,795 nodes and 39,586 triangles after the mesh coarsening, as shown in Fig. 5(a). The active site is indicated by a dashed rectangle near the center, and an interior and closer look at the active site, rotated by 90°, is also shown. The mesh quality is significantly improved: 23.22° (minimal angle) and 125.93° (maximal angle), in contrast to the minimal angle (0.02°) and maximal angle (179.55°) in the original mesh (not shown).

Another motivation of studying the mAChE molecule here is to demonstrate the constrained mesh coarsening that turns out to be very useful in refining a region of particular interest. In molecular simulation, it is a common strategy to have dense meshes near the active sites and sparser meshes elsewhere, in order to maximize the simulation accuracy while still retaining a low computational cost. This can be easily done by restricting the mesh coarsening everywhere except in user-defined regions (e.g., the active sites). In case of the mAChE molecule we use a few spheres to define the boundary of the active site. The union of the spheres is where the mesh coarsening is prohibited. Fig. 5(b) shows the constrained mesh coarsening, where the active site contains much denser meshes than the rest of the molecular surface. The resulting mesh has 22,753 nodes and 45,502 triangles and the mesh quality is 24.09° (minimal angle) and 125.94° (max angle).

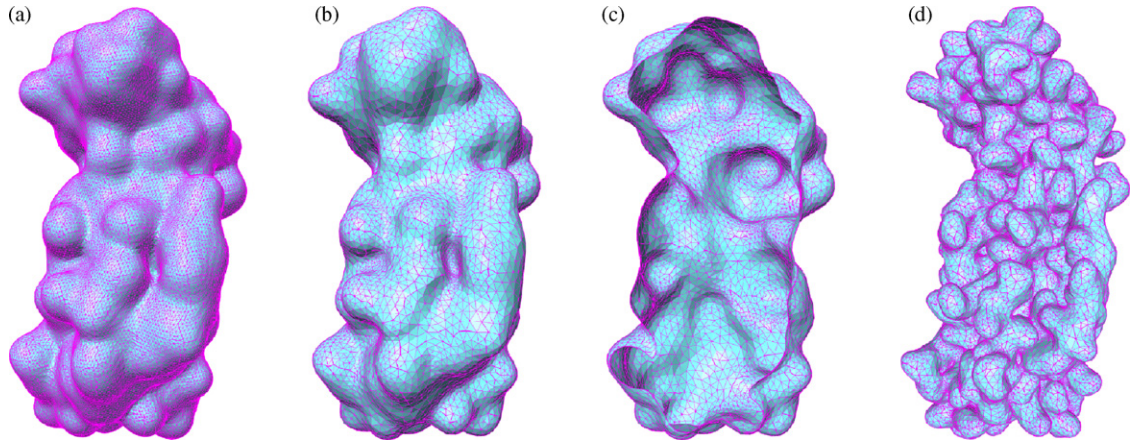

Fig. 4. Illustration of the surface generation and post-processing. This molecule is taken from PDB (ID: 1CID) and has 1381 atoms. (a) The mesh is generated using the marching cube method and the quality is improved based on the approach we described in Section 2.2. (b) While the mesh is coarsened and smoothed, which significantly reduces the mesh size, most important features are still well preserved. (c) An interior view of the processed mesh. (d) The molecular surface generated with the blobbyness at −2.5.

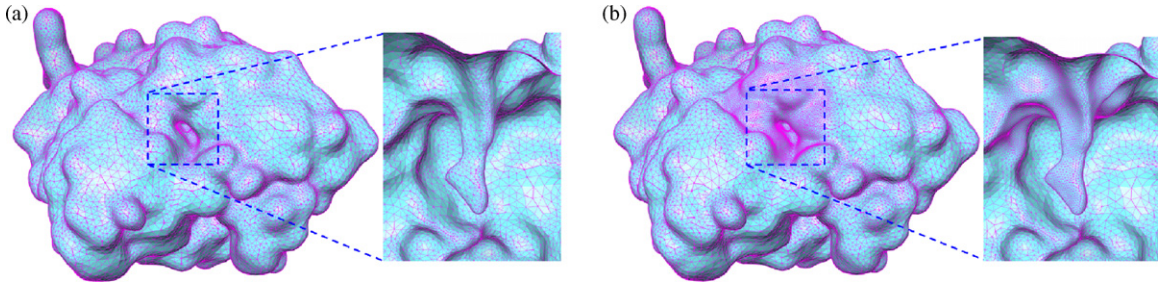

Fig. 5. Illustration of the regular and constrained mesh coarsening on the mAChE molecule. (a) The mesh is generated by a regular mesh process as presented in Section 2. The region indicated by a dashed rectangle is the active site that has the same sampling rate as other regions. (b) No coarsening is applied near the active site; therefore, the active site has much denser meshes than elsewhere. An interior and zoomed-in look at the active site, rotated by 90°, is shown in both (a) and (b).

In Fig. 6, we show the meshes generated using our tools on two large molecules. One is the trimeric bluetongue virus VP97 (PDB-ID: 1BVP) with 8109 atoms and the other is the mAChE tetramer with 36,650 atoms. The volume size for 1BVP is $206 \times 172 \times 216$. The original mesh has 150,182 nodes and 300,360 triangles. It was improved, coarsened, and smoothed using the methods described in Section 2. The processed mesh, as shown in Fig. 6(a), has 27,658 nodes and 55,312 triangles. In addition, the mesh quality was improved from 0.02° (minimal angle) and 179.84° (maximal angle) to 21.63° (minimal angle) and 133.99° (maximal angle) respectively. The Gaussian blurring technique on mAChE tetramer results in a volume of $298 \times 250 \times 266$ grids. The original mesh has 444,362 nodes and 888,748 triangles and is coarsened (two iterations) to 54,951 nodes and 109,926 triangles. The mesh quality is also improved – the minimal angle increases from 0.00° to 19.57° and the maximal angle decreases from 179.75° to 127.79°. An exterior view of the processed mesh is shown in Fig. 6(b).

3.2. Adaptive tetrahedral generation

The adaptive tetrahedral meshes are generated by constrained Delaunay triangulation as implemented in *Tetgen* [39,40], based on the surface meshes triangulated and processed by the methods described above. Fig. 7(a) and (b) shows the interior and exterior tetrahedral meshes of the molecule 1CID respectively. Since all 1381 atoms are forced to be on the interior mesh nodes, the interior mesh looks more uniform and denser than the exterior mesh. The bounding sphere for the exterior mesh is usually chosen to be about 40 times larger than the size of the molecule being considered. But for better illustration, the sphere we show here is about twice as big as the molecule. The sphere is meshed into 4350 nodes and 8696 triangles. Fig. 7(c) shows a closer look at the exterior mesh near the molecular surface. The interior mesh has a total of 14,769 and 64,080 tetrahedra. The size of the exterior mesh is, however, slightly varying on the radius of the bounding sphere chosen. If the sphere is about twice as big as the molecule, as shown in Fig. 7(b), the exterior mesh has 21,899 nodes and 94,736 tetrahedra. However, if the sphere is chosen to be 40 times bigger than the molecule, the exterior mesh will contain 22,318 nodes and 97,229 tetrahedra.

Fig. 8 shows the tetrahedralization of the mAChE molecule with finer meshes at the active site. The interior mesh and a close look at the active site are shown in Fig. 8(a). The interior mesh consists of 46,908 nodes and 222,925 tetrahedra. In contrast, the exterior tetrahedral mesh with a bounding sphere about 40 times larger than the molecule has a total of 42,964 nodes and 181,103 tetrahedra, which, unlike the 1CID molecule, is smaller than the interior mesh. This is because a large number of interior nodes have to be created to ensure that all 8362 atoms are on the mesh nodes. Fig. 8(b) shows the exterior mesh and a closer look at the active site.

We summarize the mesh generation results on a number of molecules in Table 1, where 1BBH, 1L3R, and 1TIM are PDB IDs representing respectively the cytochrome *c*', the catalytic

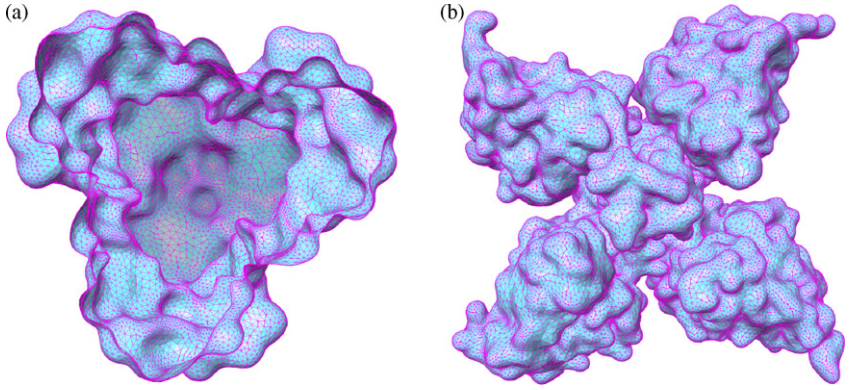

Fig. 6. Mesh generation on two very large molecules. (a) An interior view of the 1BVP trimer with 8109 atoms. (b) An exterior view of the mAChE tetramer that has a total of 36,650 atoms.

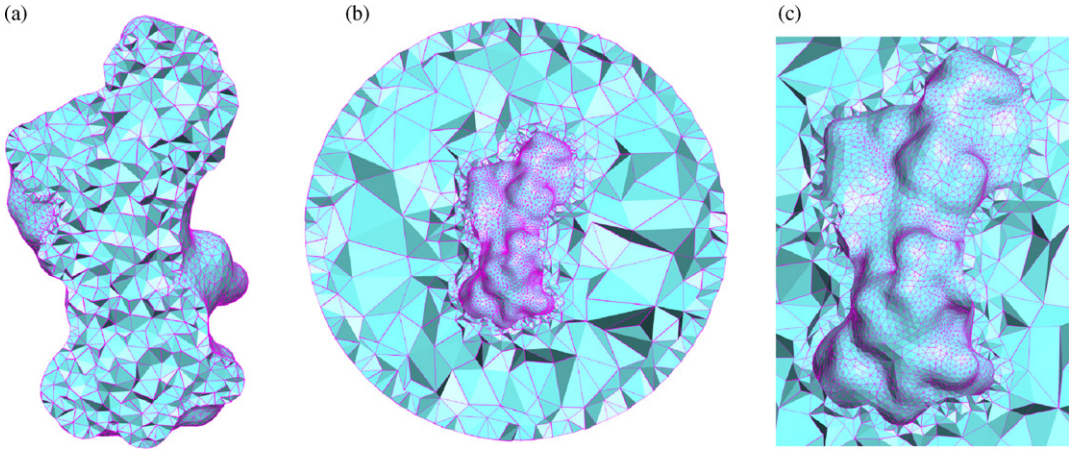

Fig. 7. Tetrahedral meshes of the molecule 1CID. (a) The interior mesh, having all atoms on the mesh nodes. (b) The exterior mesh between the molecular surface and the bounding sphere. The size of the sphere is about twice as big as the molecule. (c) A closer look at the exterior mesh near the molecular surface.

subunit of cAMP-dependent protein kinase, and the triose phosphate isomerase from chicken muscle. All surface meshes are generated and post-processed using the parameters as given in Section 3.1. The mesh coarsening was executed only once for all molecules except 1CID, 1BBH, mAChE tetramer that are coarsened and improved for two iterations (see the flowchart in Fig. 1 for illustration). Note how the mesh size and quality of 1CID changed when the initial mesh was coarsened/improved once (as previously discussed) and twice (as shown in Table 1). In general, repeating the mesh coarsening process can reduce the mesh size, but the mesh quality is less sensitive to the number of iterations executed. The errors in Table 1 represent the dislocations (in Å), estimated on all the nodes of the processed surface meshes, with respect to the initial meshes extracted by the marching cube method. As we can see, the maximal dislocations (the first row for each molecule) are around 0.5 Å or one voxel size, and the average dislocations (the second row) are about 0.05 Å. The computer time shown in Table 1 was estimated on a workstation with a single Intel Xeon 2.0 GHz CPU.

3.3. Mesh quality analysis

Although the minimal and maximal angles are good indicators of the mesh quality, a histogram showing the distribution of all angles in a mesh would provide us additional information about the mesh quality. To this end, we divide the angles ranging from 0° to 180° into 18 intervals by every 10°. The normalized histogram in each interval $[i \times 10° - 10°, i \times 10°], i = 1, \ldots, 18$, stands for the percentage of the angles lying in that interval over the total number of angles in a mesh. We consider four molecules as examples here: 1CID (1381 atoms), 1TIM (3740 atoms), mAChE monomer (8362 atoms), and mAChE tetramer (36,650 atoms). Interestingly the histograms of the initial meshes for all four molecules are almost identical. Therefore we use only one curve to represent the histograms of the initial meshes, as shown in blue (the thick and dashed one) in Fig. 9. The remaining four curves (the one for 1TIM is almost identical to that of the mAChE monomer) are the histograms of the meshes after the quality improvement using the method described in Section 2.2. From these curves we can see that about 85%–90% of the angles lie in the range of $[40°, 80°]$, while in the original meshes there are only 50% of the angles in this range. Like the histograms of the initial meshes, the improved meshes also show very similar histograms, implying that the modified angle-based method we described is somewhat data-independent, at least for the meshes extracted from the Gaussian-blurred molecular maps when the same set of parameters are used.

4. Discussion

4.1. A fast Gaussian-blurring implementation

As we mentioned in Section 2.1, the Gaussian-blurring function in Eq. (2), compared to Eq. (1), has an advantage that

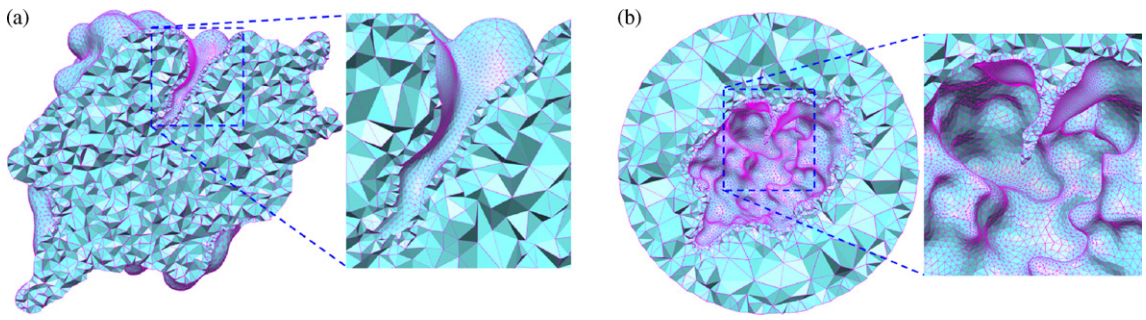

Fig. 8. Tetrahedral meshes for the mAChE molecule. (a) The interior mesh, having all atoms on the mesh nodes. (b) The exterior mesh between the molecular surface and the bounding sphere whose size is about twice as big as the molecule. A closer look at the active site is shown in both (a) and (b).

Table 1
A summary of surface and tetrahedral mesh generations

Molecules	# of Atoms	Surf-N	Surf-T	Min-A	Max-A	Tet-N	Tet-T	Errors	Costs
1CID	1,381	62,454	124,904	0.02°	179.10°	6,657	30,774	0.43	37
		3,500	6,996	24.74°	119.01°	13,228	57,531	0.05	8
1BBH	1,934	73,156	146,308	0.02°	179.69°	12,261	57,326	0.38	53
		6,370	12,736	21.67°	117.56°	17,660	75,571	0.05	13
1L3R	2,855	79,314	158,624	0.02°	179.55°	24,210	109,129	0.43	64
		13,542	27,080	23.39°	128.25°	28,986	123,307	0.05	24
1TIM	3,740	90,912	181,820	0.02°	179.45°	31,161	140,826	0.53	77
		17,332	34,660	22.49°	123.39°	35,194	149,759	0.05	29
1BVP	8,109	150,182	300,360	0.02°	179.84°	54,070	253,799	0.72	139
		27,658	55,312	21.63°	133.99°	51,136	215,669	0.06	48
mAChE1	8,362	96,046	192,088	0.02°	179.55°	42,456	204,821	0.70	82
		19,795	39,586	23.22°	125.93°	38,968	165,648	0.06	36
mAChE2	8,362	96,046	192,088	0.02°	179.55°	46,908	222,925	0.70	84
		22,753	45,502	24.09°	125.94°	42,964	181,103	0.05	40
mAChE3	36,650	444,362	888,748	0.00°	179.75°	134,204	682,573	0.53	432
		54,951	109,926	19.57°	127.79°	92,805	386,505	0.05	105

Surf-N and Surf-T: surface mesh sizes (nodes and triangles): original (1st row) and processed (2nd row); Min-A and Max-A: mesh quality (minimal/maximal angles): original (1st row) and processed (2nd row); Tet-N and Tet-T: volumetric mesh sizes (nodes and tetrahedra): interior (1st row) and exterior (2nd row); Errors: dislocations (Å) of the processed surface meshes: maximal (1st row) and average (2nd row); Costs: computer time (seconds): surface mesh generation and processing (1st row) and tetrahedral generation (2nd row); mAChE1: the mAChE monomer with regular mesh coarsening; mAChE2: the mAChE monomer with constrained mesh coarsening; mAChE3: the mAChE tetramer with regular mesh coarsening.

the decay rate κ is a constant for all atoms. This can lead to a fast Gaussian-blurring implementation using the recursive scheme as described in [48]. Let us rewrite Eq. (2) into the following standard Gaussian function:

$$G(\mathbf{x}) = \sum_{i=1}^{N} g_i \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-((|\mathbf{x}-\mathbf{c}_i|^2)/(2\sigma^2))}, \tag{7}$$

where

$$g_i = \sqrt{\frac{-\pi}{\kappa}} \mathrm{e}^{-\kappa r_i^2} \tag{8}$$

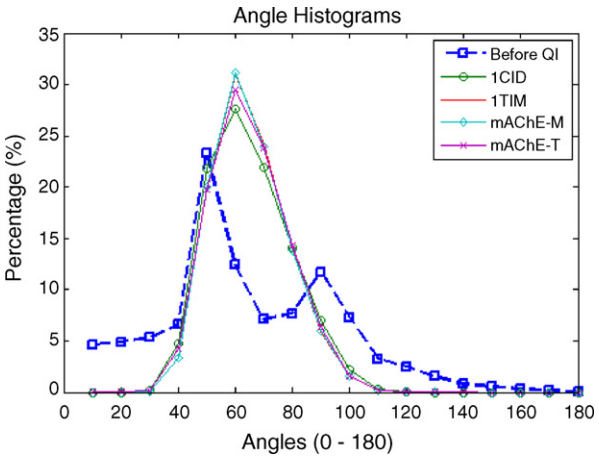

Fig. 9. Histograms (in percentage) of the angles in the initial and processed meshes. Four molecules are considered: 1CID, 1TIM, mAChE monomer and mAChE tetramer. Since all four molecules demonstrate almost identical histograms before quality improvement (QI), we show only one curve (thick and dashed one in blue color) to represent the histogram for the initial meshes. The other four, shown in different colors, are the histograms of the processed meshes (1TIM has an almost identical histogram to that of mAChE monomer). Most angles in the processed meshes lie in the range of $[40°, 80°]$.

stands for the contribution of ith atom centered at $\mathbf{x}_i$ with radius r_i, and $\sigma = \sqrt{-0.5\kappa}$ is the standard deviation of the Gaussian function. Given a negative constant κ and a list of atoms, the following steps can be implemented to achieve fast Gaussian-blurring calculations:

1. Compute σ and g_i.
2. Digitize the space into grids (same as we did in Section 2.1). For each atom at $\mathbf{x}_i$, find the nearest grid $\mathbf{m}_i$. Initialize the value at $\mathbf{m}_i$ with g_i.
3. Run the recursive Gaussian filtering [48].

It was shown in [48] that the numerical error of the recursive implementation of Gaussian filter is relatively less than 0.68% compared to the analytical Gaussian function. But the biggest gain of this scheme is that it can be even faster than the optimized FFT-based implementation. In case of the molecular maps as we outlined above, there is another type of error that comes from the approximation of $\mathbf{x}_i$ with the nearest grid point $\mathbf{m}_i$. However, this digitization error should be quite small when the space is densely sampled.

4.2. Quality improvements on user-defined meshes

In addition to the PQR [19] or PDB [10] formats, our mesh toolchain can also read and process a user-defined arbitrary triangular mesh that has very low quality or is too large to be handled by standard numerical simulators. As an example, we consider the mAChE monomer again and generate the mesh by the widely used MSMS tool [37]. Fig. 10(a) shows the mesh generated by MSMS. As we can see, this mesh looks a little "bumpy" on the surface and contains a lot of very "sharp" triangles. The minimal and maximal angles of this mesh are 0.00° and 176.92° respectively. There are only 47% of the total

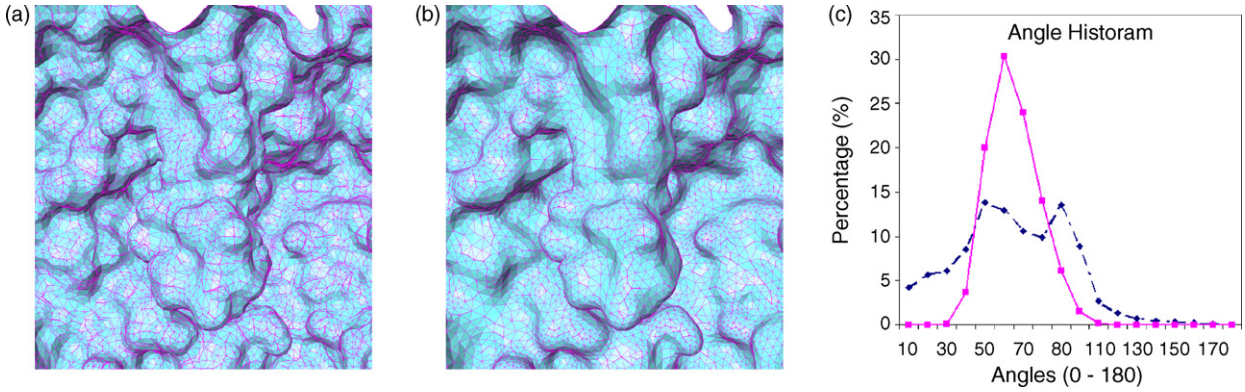

Fig. 10. Mesh quality improvement on a user-specified surface mesh. (a) Original mesh generated by MSMS. (b) After mesh coarsening and quality improvement, the mesh size is about half as large as the original one, and the quality becomes significantly better. (c) The angle distributions of the meshes before (dashed or black curve) and after (solid or purple curve) the mesh post-processing.

angles lying in the range of $[40°, 80°]$. The numbers of nodes and triangles of this mesh are 62,402 and 124,804 respectively. After our mesh quality improvement, coarsening, and smoothing, we reduce the mesh size to 35,022 nodes and 70,044 triangles. The mesh quality is significantly improved to 25.56° (minimal angle) and 125.46° (maximal angle). The picture showing the same region of the mAChE monomer can be seen in Fig. 10(b). In contrast to the 47%, we now have 88% of the angles lying in the range of $[40°, 80°]$. Fig. 10(c) depicts the histograms of angles for the meshes before and after our mesh post-processing algorithms.

4.3. Multilevel mesh generation for multigrid-based solvers

By coarsening the surface triangular meshes using different values of T_0 in Eq. (6), we can achieve a multilevel of meshes with different details. Fig. 11 illustrates the meshes of the 1BVP trimeric molecule at four different levels: $T_0 = 0.1, 0.3, 0.5$ and 0.7. We can see that, as T_0 increases, the mesh size decreases and some small features start to disappear. From our experiments, $T_0 = 0.2$ seems to be a good tradeoff to balance the accuracy and the mesh size (cost). An initial mesh, if it is too large, may be subject to coarsening for two or more times, as is the case shown in Fig. 11. This turns out to preserve features better than simply coarsening the mesh once with a large T_0.

It is well-known that the most efficient numerical methods for solving certain classes of PDEs (called *elliptic* PDEs) are based on multilevel methods, which are extensions of the original multigrid algorithms developed for the Poisson equation in the 1970s [13,24]. The basic idea behind such multilevel methods for solving a linear system $Au = f$ is the recursive application of a simple two-level method consisting of "*Smoothing–Restriction–Solution–Prolongation*" steps [14,25]. Having access to a hierarchy of meshes can be leveraged to build fast solvers for partial differential equations (PDEs) posed in the volumetric mesh around the biological structure, or posed on the surface mesh (cf. [2,3] for detailed discussions of this class of methods and further references).

4.4. Applications: molecular simulation

We have successfully applied the exterior tetrahedral mesh of the mAChE monomer as shown in Fig. 8(b) to studying the ACh diffusion behavior near the mAChE molecule. The exterior mesh alone consists of 42,964 nodes and 181,103 tetrahedra. With the SMOL solver (cf. [17]), the linear system converges very well and the outputs obtained are consistent with other studies [17,41]. Fig. 12 depicts the ACh concentration at the steady-state Smoluchowski diffusion at 0.150M ionic strength with the external boundary as Dirichlet using OpenDX.

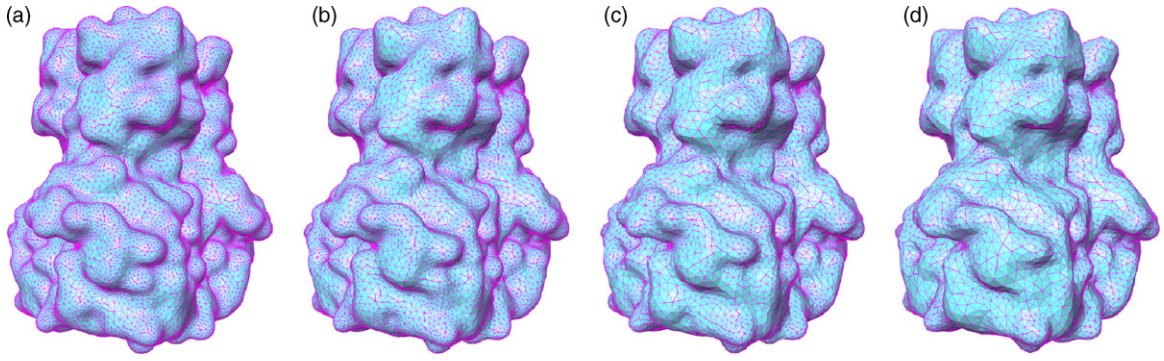

Fig. 11. Multilevel mesh generation by coarsening (experiments on 1BVP). The parameter T_0 is described in Eq. (6). The original mesh has 150,182 nodes and 300,360 triangles and is coarsened for two iterations with different coarsening parameters T_0. (a) $T_0 = 0.1$. Nodes $= 26,842$, Triangles $= 53,680$. (b) $T_0 = 0.3$. Nodes $= 19,588$, Triangles $= 39,172$. (c) $T_0 = 0.5$. Nodes $= 12,882$, Triangles $= 25,760$. (d) $T_0 = 0.7$. Nodes $= 9,480$, Triangles $= 18,956$.

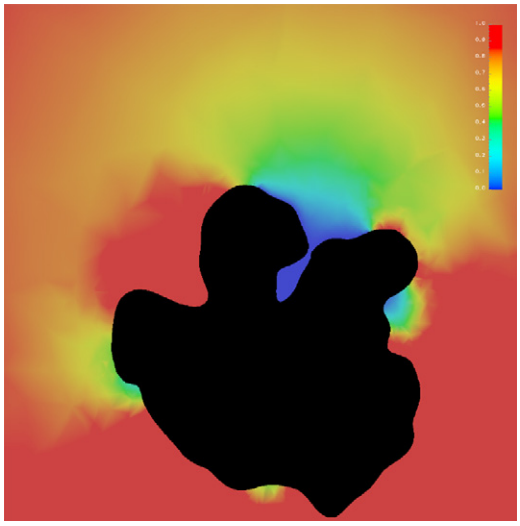

Fig. 12. ACh distribution around the mAChE enzyme at the steady-state diffusion stage simulated by the SMOL solver [17]. Red corresponds to ACh concentration in the bulk, blue to a concentration of zero.

The red represents high concentration, while the blue low concentration.

All algorithms described in Section 2 have been implemented in ANSI-C and incorporated into the Finite Element ToolKit (*FETK*) software suite (http://www.fetk.org) [26] as one of its major components, called *GAMER* (*G* eometry-preserving *A* daptive *M* esh*ER*). The *FETK* libraries and tools are developed by the Holst Research Group at UCSD and are designed to solve coupled systems of partial differential equations (PDE) and integral equations (IE) using parallel adaptive multilevel finite element methods. Built on top of this highly portable software toolkit are two widely used application-oriented molecular modeling tools: the APBS [6] and the SMOL [42]. APBS is an adaptive Poisson-Boltzmann solver, designed for simulating electrostatic properties of molecules in salty, aqueous media (http://apbs.sourceforge.net/). SMOL is designed to solve the Smoluchowski diffusion equation (http://mccammon.ucsd.edu/smol/). The *GAMER* together with other libraries in *FETK* will be made available in source form under the GNU Lesser General Public License.

5. Conclusion

We presented a set of efficient algorithms dealing with surface and volumetric mesh generation for molecules with arbitrary sizes and shapes. Our methods have been demonstrated on a number of examples to generate smooth, adaptive, and high quality meshes with features well preserved. Although this paper focused on molecular mesh generation having the center and radius of each atom as inputs, the approaches described here can also be applied to generate and process meshes from general-type scalar volumes (e.g., reconstructed 3D imaging data) or low-quality surface meshes provided by the user (e.g., molecular surfaces by MSMS [37]). The availability of our software toolchain will give researchers in molecular/cellular modeling and simulation areas easy access to high-fidelity geometric models.

Acknowledgments

Z. Yu was supported in part by NSF Award 0411723, DOE Award DE-FG02-05ER25707, and NBCR (http://nbcr.sdsc.edu/). M. Holst was supported in part by NSF Awards 0411723, 0511766, and 0225630, and DOE Awards DE-FG02-05ER25707 and DE-FG02-04ER25620. Y. Cheng and J.A. McCammon were supported in part by the NSF (MCB-0506593), NIH (GM31749), HHMI, CTBP, NBCR, W.M. Keck Foundation, and Accelrys, Inc.

References

[1] M. Ainsworth, J.T. Oden, A Posterori Error Estimation in Finite Element Analysis, Wiley-Interscience, 2000.
[2] B. Aksoylu, S. Bond, M. Holst, An odyssey into local refinement and multilevel preconditioning III: implementation and numerical experiments, SIAM J. Sci. Comput. 25 (2003) 478–498.
[3] B. Aksoylu, M. Holst, Optimality of multilevel preconditioners for local mesh refinement in three dimensions, SIAM J. Numer. Anal. 44 (2006) 1005–1025.
[4] C. Bajaj, H. Lee, R. Merkert, V. Pascucci, NURBS based B-rep models from macromolecules and their properties, in: Proceedings of Fourth Symposium on Solid Modeling and Applications, 1997, pp. 217–228.
[5] C.L. Bajaj, V. Pascucci, A. Shamir, R.J. Holt, A.N. Netravali, Dynamic maintenance and visualization of molecular surfaces, Discrete Appl. Math. 127 (1) (2003) 23–51.
[6] N.A. Baker, D. Sept, S. Joseph, M. Holst, J.A. McCammon, Electrostatics of nanosystems: application to microtubules and the ribosome, Proc. Natl. Acad. Sci. 98 (2001) 10037–10041.
[7] R. Bank, M. Holst, A new paradigm for parallel adaptive meshing algorithms, SIAM Rev. 45 (2003) 291–323.
[8] D. Bashford, D.A. Case, Generalized Born models of macromolecular solvation effects, Annu. Rev. Phys. Chem. 51 (2000) 129–152.
[9] P.W. Bates, G.W. Wei, S. Zhao, Minimal molecular surfaces and their applications, J. Comput. Chem. 29 (3) (2008) 380–391.
[10] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucl. Acids Res. 28 (2000) 235–242.
[11] M. Berzins, A solution-based triangular and tetrahedral mesh quality indicator, SIAM J. Sci. Comput. 19 (1998) 2051–2060.
[12] J.F. Blinn, A generalization of algegraic surface drawing, ACM Trans. Graph. 1 (3) (1982) 235–256.
[13] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comp. 31 (1977) 333–390.
[14] W.L. Briggs, V.E. Henson, S.F. McCormick, A multigrid tutorial, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.
[15] C.-Y. Chena, K.-Y. Cheng, A sharpness dependent filter for mesh smoothing, Comput. Aided Geom. Des. 22 (5) (2005) 376–391.
[16] H.L. Cheng, X. Shi, Quality mesh generation for molecular skin surfaces using restricted union of balls, Proc. IEEE Vis. (2005) 51–57.
[17] Y.H. Cheng, J.K. Suen, D.Q. Zhang, S.D. Bond, Y.J. Zhang, Y.H. Song, N.A. Baker, C.L. Bajaj, M.J. Holst, J.A. McCammon, Finite element analysis of the time-dependent Smoluchowski equation for acetylcholinesterase reaction rate calculations, Biophys. J. 92 (2007) 3397–3406.
[18] M.L. Connolly, Analytical molecular surface calculation, J. Appl. Cryst. 16 (5) (1983) 548–558.
[19] T.J. Dolinsky, J.E. Nielsen, J.A. McCammon, N.A. Baker, PDB2PQR: an automated pipeline for the setup, execution, and analysis of poisson-boltzmann electrostatics calculations, Nucl. Acids Res. 32 (2004) 665–667.
[20] B.S. Duncan, A.J. Olson, Shape analysis of molecular surfaces, Biopolymers 33 (1993) 231–238.
[21] J.-J. Fernandez, S. Li, An improved algorithm for anisotropic nonlinear diffusion for denoising cryo-tomograms, J. Struct. Biol. 144 (2003) 152–161.

[22] L.A. Freitag, C.F. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, Int. J. Numer. Methods Eng. 40 (21) (1997) 3979–4002.
[23] J.A. Grant, B.T. Pickup, A Gaussian description of molecular shape, J. Phys. Chem. 99 (1995) 3503–3510.
[24] W. Hackbusch, A fast iterative method solving Poisson's equation in a general region, in: Proceedings of the Conference on the Numerical Treatment of Differential Equations (Lecture Notes in Mathematics), Springer-Verlag, 1978, vol. 631, pp. 51–62.
[25] W. Hackbusch, Multi-grid Methods and Applications, Springer-Verlag, Berlin, Germany, 1985.
[26] M. Holst, Adaptive numerical treatment of elliptic systems on manifolds, Adv. Comput. Math. 15 (2001) 139–191.
[27] K.H. Huebner, D. Dewhirst, D.E. Smith, T.G. Byrom, The Finite Element Method for Engineers, Wiley-IEEE, 2001.
[28] W. Im, D. Beglov, B. Roux, Continuum solvation model: electrostatic forces from numerical solutions to the poisson-boltzmann equation, Comp. Phys. Commun. 111 (1998) 59–75.
[29] I. Jollife, Principal Component Analysis, Springer-Verlag, NY, 1986.
[30] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, in: Proceedings of SIGGRAPH'02, 2002, pp. 339–346.
[31] B. Lee, F.M. Richards, The interpretation of protein structures: estimation of static accessibility, J. Mol. Biol. 55 (3) (1971) 379–400.
[32] A. Liu, B. Joe, Relationship between tetrahedron shape measures, BIT 34 (1994) 268–287.
[33] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, Comput. Graph. 21 (4) (1987) 163–169.
[34] M. Nina, W. Im, B. Roux, Optimized atomic radii for protein continuum electrostatics solvation forces, Biophys. Chem. 78 (1–2) (1999) 89–96.
[35] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639.
[36] F.M. Richards, Areas, volumes, packing, and protein structure, Ann. Rev. Biophys. Bioeng. 6 (1977) 151–156.
[37] M.F. Sanner, A.J. Olson, J.C. Spehner, Reduced surface: an efficient way to compute molecular surfaces, Biopolymers 38 (1996) 305–320.
[38] M.J. Schnieders, N.A. Baker, P. Ren, J.W. Ponder, Polarizable atomic multipole solutes in a poisson-boltzmann continuum, J. Chem. Phys. 126 (2007) 124114–124135.
[39] H. Si, Tetgen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator. Technical Report 9, Weierstrass Institute for Applied Analysis and Stochastics, 2004 (software download: http://tetgen.berlios.de).
[40] H. Si, K. Gartner, Meshing piecewise linear complexes by constrained delaunay tetrahedralizations, in: Proceedings of the 14th International Meshing Roundtable, 2005.
[41] Y. Song, Y. Zhang, C.L. Bajaj, N.A. Baker, Continuum diffusion reaction rate calculations of wild type and mutant mouse acetylcholinesterase: adaptive finite element analysis, Biophys. J. 87 (2004) 1558–1566.
[42] Y. Song, Y. Zhang, T. Shen, C.L. Bajaj, J.A. McCammon, N.A. Baker, Finite element solution of the steady-state Smoluchowski equation for rate constant calculations, Biophys. J. 86 (2004) 2017–2029.
[43] M. Totrov, R. Abagyan, The contour-buildup algorithm to calculate the analytical molecular surface, J. Struct. Biol. 116 (1996) 138–143.
[44] J.A. Wagoner, N.A. Baker, Assessing implicit models for nonpolar mean solvation forces: The importance of dispersion and volume terms, Proc. Natl. Acad. Sci. 103 (22) (2006) 8331–8336.
[45] J. Weickert, Anisotropic Diffusion In Image Processing. ECMI Series, Teubner-Verlag, Stuttgart, 1998.
[46] H. Xu, T.S. Newman, 2D FE quad mesh smoothing via angle-based optimization, in: Proceedings of the 5th International Conference on Computational Science, 2005, pp. 9–16.
[47] H. Yagou, Y. Ohtake, A. Belyaev, Mesh Smoothing via Mean and Median Filtering Applied to Face Normals, 2002, pp. 124–131.
[48] I.T. Young, L.J. Vliet, Recursive implementation of the Gaussian filter, Signal Process. 44 (1995) 139–151.
[49] Z. Yu, C. Bajaj, A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion, in: Proceedings of International Conference Computer Vision and Pattern Recognition, 2004, pp. 415–420.
[50] Z. Yu, C. Bajaj, A structure tensor approach for 3D image skeletonization: applications in protein secondary structural analysis, in: Proceedings of IEEE International Conference on Image Processing, 2006, pp. 2513–2516.
[51] Z. Yu, C. Bajaj, Computational approaches for automatic structural analysis of large bio-molecular complexes. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2007, http://doi.ieeecomputersociety.org/10.1109/TCBB.2007.70226.
[52] Y. Zhang, C. Bajaj, B.-S. Sohn, 3D finite element meshing from imaging data, Comput. Methods Appl. Mech. Eng. 194 (48–49) (2005) 5083–5106.
[53] Y. Zhang, G. Xu, C. Bajaj, Quality meshing of implicit solvation models of biomolecular structures, Comput. Aided Geom. Des. 23 (6) (2006) 510–530.
[54] W. Zhao, G. Xu, C. Bajaj, An algebraic spline model of molecular surfaces, Proc. ACM Symp. Solid Phys. Model. 1 (2007) 297–302.
[55] T. Zhou, K. Shimada, An angle-based approach to two-dimensional mesh smoothing, in: Proceedings of the Ninth International Meshing Roundtable, 2000, pp. 373–384.