An algorithm for tetrahedral mesh generation based on conforming constrained Delaunay tetrahedralization

Yi-Jun Yang, Jun-Hai Yong, Jia-Guang Sun

An algorithm for tetrahedral mesh generation based on conforming constrained Delaunay tetrahedralization

Yi-Jun Yang[a,*], Jun-Hai Yong[b], Jia-Guang Sun[a,b]

[a]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China*
[b]*School of Software, Tsinghua University, Beijing 100084, People's Republic of China*

Abstract

An unstructured tetrahedral mesh generation algorithm for 3D model with constraints is presented. To automatically generate a tetrahedral mesh for model with constraints, an advancing front algorithm is presented based on conforming constrained Delaunay tetrahedralization (CCDT). To reduce the number of visibility tests between vertices with respect to model faces as well as the computation of constrained Delaunay tetrahedra, a sufficient condition for DT (constrained Delaunay tetrahedralization whose simplexes are all Delaunay) existence is presented and utilized coupled to uniform grid and advancing front techniques in our algorithm. The mesh generator is robust and exhibits a linear time complexity for mechanical models with uniform density distribution.

1. Introduction

Mesh generation algorithms play an important role in finite element methods (FEM) [1], biomechanics [2], virtual surgery [3,4] and geometric modeling [5]. Quality of the mesh strongly influences accuracy and efficiency of analysis and simulation. Delaunay tetrahedra have desirable properties such as their tendency to favor round tetrahedra over skinny tetrahedra and suitability for Delaunay refinement algorithms [6]. However, Delaunay tetrahedralizations (DT) are convex. Mechanical models usually are not. Modes have boundaries and constraints that must be respected by the mesh (Fig. 1). To address this problem, three methods have been established: conforming DT, variational Delaunay tetrahedralization and conforming constrained Delaunay tetrahedralization (CCDT).

Conforming Delaunay approaches [7–11] repeatedly insert Steiner points into the mesh, while mesh is Delaunay refined accordingly, until boundary and constraints are respected by the mesh: every face (edge) is represented by a union of triangular faces (edges) of DT. The question of where to insert vertices to obtain boundary and constraint conformity is difficult. Radovitzky and Ortiz [10] used a combination of local transformations and subdivision to guarantee that the surface triangulation be a subcomplex of the DT. However, this algorithm fails when a minimal mesh size is touched. Murphy et al. [9] offered an algorithm that can mesh model with small dihedral angle boundary. But, a large number of small edges and additional points are introduced into the tetrahedral mesh accordingly. How to reconcile the need for boundary and constraint conformity with the need for good-quality tetrahedra is the most important issue for conforming Delaunay algorithm.

---

*Corresponding author. Tel.: +86 10 62795459;
fax: +86 10 62795460

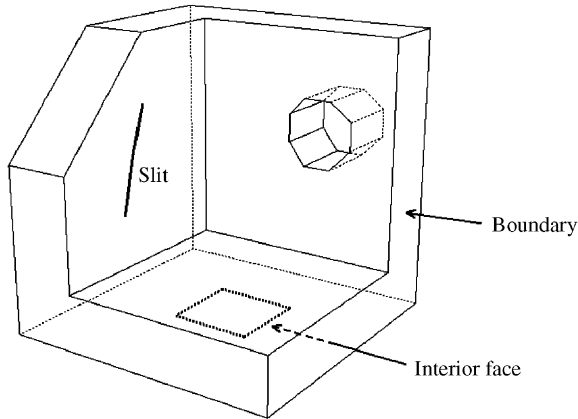*E-mail address:* yang_yj03@hotmail.com (Y.-J. Yang).

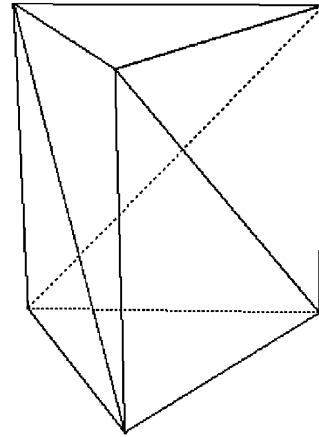Fig. 1. A bracket with constraints (slit and interior face).

Fig. 2. An untetrahedralizable polyhedron.

Variational Delaunay approaches [12–15] adjust the mesh by adding additional points where the boundary and constraints intersect Delaunay tetrahedral faces or moving mesh vertices toward boundary or constraint faces to recover missing boundaries and constraints, while Delaunay property of some simplexes are violated. Fuchs [12] used body-centered-cubic (BCC) lattice refinement to obtain desired node density and projected exterior DT vertices to the boundary to obtain boundary conformity. Molino et al. [13] utilized a defined distance function and uniform crystal lattice to guide mesh subdivision for highly deformable models and obtain boundary fitting via masses and springs. Reliance on heuristics and undesirable quality tetrahedra near the boundary are big problems for this type of algorithms.

In 2D, a constrained Delaunay triangulation (CDT) [16–18] involves a set of edges and vertices while maintaining most of the favorable properties of DT (such as maximizing the minimum angle). However in 3D, without Steiner points and surface modification, tetrahedralization is not available for all models [19]. An untetrahedralizable example is shown in Fig. 2. Thus, there is no CDT for some models in 3D for additional points are prohibited in CDT. Shewchuk [6] presented a sufficient condition for CDT existence in 3D and a tetrahedral mesh generation algorithm utilizing this condition. His algorithm should be called CCDT instead since additional points are inserted into the mesh to guarantee the existence of CDT. In comparison with conforming DT, CCDT adds fewer vertices and all tetrahedra are constrained Delaunay instead. For clarity, triangulations of a planar straight line graph (PSLG) in 2D are given in Fig. 3. Conforming Delaunay triangulation adds more points to recover missing edges (faces involved in 3D) than CDT (CCDT in 3D). Furthermore, no Steiner points are needed for 2D CDT.

In this paper, an advancing front tetrahedral mesh generation algorithm is presented for models with
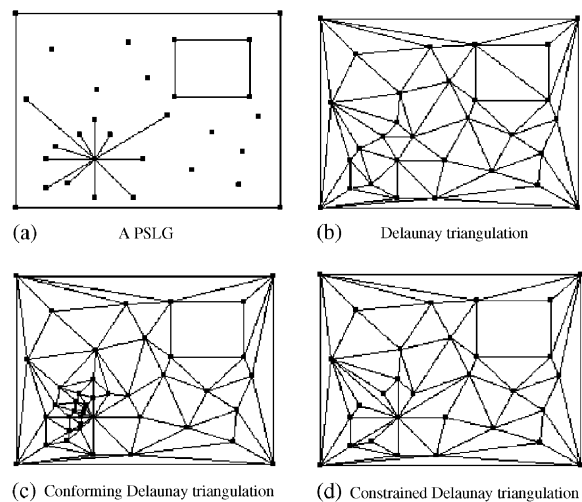

Fig. 3. PSLG and its triangulations: (a) a PSLG; (b) Delaunay triangulation; (c) conforming Delaunay triangulation and (d) constrained Delaunay triangulation.

constraints based on CCDT. First, a surface meshing algorithm [10] is utilized to mesh the model. To guarantee CDT existence of the mechanical model, middle points of missing grazeable edges are repeatedly added to the mesh until all grazeable edges are represented as set of continuous edges of DT. Second, interior points are inserted in a face-centered-cubic (FCC) crystal lattice arrangement coupled to octree spatial subdivision, which gives better tetrahedra than other Bravais lattice arrangements for an error bound [10]. Finally, to accelerate tetrahedral mesh generation, a sufficient condition for DT existence is presented and an advancing front algorithm utilizing this condition and uniform grid technique is implemented. The mesh generator is robust and exhibits a linear time complexity for models with uniform density distribution.

This paper is organized as follows. Section 2 describes grazeable edge protection theorem. Section 3 describes method of inserting interior points. Section 4 describes how to construct uniform grid. Section 5 presents a sufficient condition for DT and an advancing algorithm utilizing this condition. Section 6 gives some mechanical examples. Performance of face protection and superior quality meshes of presented algorithm are exhibited by given examples. Section 7 concludes the paper.

2. Grazeable edge protection

Shewchuk [6] presented a sufficient condition for CDT existence. If a model is grazeable edge protected, the existence of CDT can be guaranteed. However, in FEM, surface mesh can be modified by local transformations as long as boundary and constraints are not violated. Based on this character, all non-constraint interior edges of the surface triangulation are removed before addition of Steiner points. To obtain grazeable edge protection, middle points of missing grazeable edges are repeatedly inserted to the mesh until all grazeable edges are represented as set of continuous edges of DT. Models in which five or more vertices are cospherical may cause the CDT algorithm to fail. So symbolic perturbation [20] is introduced to simulate the circumstance where no five vertices are on a common sphere.

3. Interior points insertion via FCC lattice

The size of tetrahedra is controlled by the density function that may vary throughout the body. In order to generate tetrahedra with prescribed size, interior points are inserted in a FCC crystal lattice arrangement coupled to octree space subdivision, with the local lattice parameter determined by the prescribed density function. Bounding box of the model is set to be the initial octant. Octants are classified into three types according to their location corresponding to the model: exterior octant, boundary octant and interior octant. Boundary octant is intersected by model faces or contains model vertices, while exterior octant lies outside the model and interior octant lies inside the model, respectively. Exterior octants are never further split any more. Boundary and interior octants are subdivided into eight equally sized children and boundary and interior son octants are recursively split as long as they do not match the prescribed size function and do not touch the maximal user-defined octree depth. As is common practice, the generational difference between adjacent octree leaves is restricted to one [21]. As a direct consequence of this restriction, the number of types of
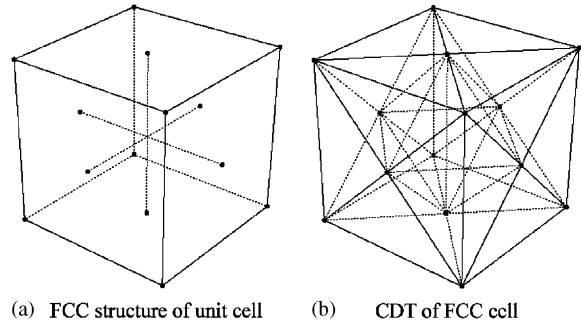

Fig. 4. FCC structure and its tetrahedralization: (a) FCC structure of unit cell and (b) CDT of FCC cell.

tetrahedra in transition area between different mesh sizes is finite.

Once octree is constructed, all boundary and interior leaves are designated as FCC cells (Fig. 4(a)) whose corners and centers of the cell faces lying inside the body are inserted to the mesh. Radovitzky and Ortiz [10] compared FCC with other 13 Bravais lattice and concluded that for a unit cell, CDT tetrahedra quality (Fig. 4(b)) of FCC lattice arrangement is better than others according to their defined aspect ratio.

According to FCC lattice arrangement, corners and centers of faces of boundary leaves may lie outside the body. To handle this problem, a conventional algorithm is implemented to judge whether point is inside the model or not: shoot a radial from the point and compute the number of intersections of the radial with boundary faces. If the number of intersections is even, point is located outside the body; otherwise point is located inside the body. Finally, interior points lying in minimal circumspheres of model triangular faces are eliminated to avoid slivers [22]. Insertion of interior points does not affect CDT existence. That is to say when a model $\mathbf{X}$ has a CDT, the model $\mathbf{X}_1$ that is composed of $\mathbf{X}$ and a set of additional interior points also has a CDT.

4. Construction of uniform grid

Bentley et al. [23] used uniform grid to solve nearest neighbor searching problem and proved that, under the assumption that n points are chosen independently from a uniform distribution on the unit square in 2D, the nearest neighbor of a query point can be found by spiral search in constant expected time. Uniform grid has been extensively utilized to accelerate mesh generation in 2D [17,24]. Here, we utilize uniform grid to accelerate tetrahedral generation in our algorithm. Given a model with faces $\mathbf{F} = \{\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{m-1}\}$ where $\mathbf{f}_i = (\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$, $0 \leqslant i \leqslant m-1$ and vertices $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}\}$ where $\mathbf{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})$, $0 \leqslant i \leqslant n-1$, all the vertices and faces are put into the uniform grid to accelerate vertex search

and visibility determination. The uniform grid can constrict vertex search and visibility determination to a local area around current face, so tetrahedron derived from current face can be generated in a constant expected time for mechanical models with uniform density distribution. To construct the uniform grid, the first step is to compute bounding box of the body and shift the box outward by a tolerance *TOL*

$$x_{\min} = x_{\min} - TOL, \quad x_{\max} = x_{\max} + TOL,$$
$$y_{\min} = y_{\min} - TOL, \quad y_{\max} = y_{\max} + TOL,$$
$$z_{\min} = z_{\min} - TOL, \quad z_{\max} = z_{\max} + TOL,$$

where "=" is the assignment operator. Then compute the grid size

$$size = \left\{ \frac{(x_{\max} - x_{\min})(y_{\max} - y_{\min})(z_{\max} - z_{\min})}{\max(m, n)} \right\}^{1/3}.$$

Finally, put vertices into corresponding cells and faces into intersected or located cells, respectively. For a vertex $\mathbf{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})$, calculate

$$cell\_x = \text{int}\left(\frac{p_{i,x} - x_{\min}}{size} \right),$$

$$cell\_y = \text{int}\left(\frac{p_{i,y} - y_{\min}}{size} \right),$$

$$cell\_z = \text{int}\left(\frac{p_{i,z} - z_{\min}}{size} \right).$$

Vertex $\mathbf{p}_i$ is put into the uniform grid cell ($cell\_x, cell\_y, cell\_z$). For a face $\mathbf{f}_i(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$, the bounding box of $\mathbf{f}_i$ is first calculated and cells intersected by the bounding box are determined. For each determined cell, if $\mathbf{f}_i$ lies inside or intersects with the cell, then put $\mathbf{f}_i$ into the cell. Vertices and faces in the grid cells are administrated by double linked list. The grid structure accelerates the search for the fourth vertex. Given a face $\mathbf{f}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$, the grid cell containing barycenter of the face is first determined. Vertex search is spread spirally around the cell until a candidate vertex visible from vertices of face $\mathbf{f}$ with respect to $\mathbf{F}$ is found. Two vertices are visible from each other with respect to $\mathbf{F}$ if the edge between the two vertices does not intersect any face of $\mathbf{F}$. A similar procedure utilizing the uniform grid is to verify that for a chosen vertex $\mathbf{p}_4$ there are no visible vertices inside the circumsphere of formed tetrahedron. For this purpose, the bounding box of formed tetrahedron is first calculated and cells intersected by the bounding box are determined. All vertices lying inside these cells are tested whether they are inside the tetrahedral circumsphere and whether they are visible from vertices of face $\mathbf{f}$. If an interior visible vertex is found, the current vertex is replace by it. This

verification and replacement are performed repeatedly until no interior visible vertices are found anymore.

5. Tetrahedralization

DT [25–27] of model vertices is needed when testing grazeable edge protection and face protection. Fang and Piegl [24] utilized uniform grid to accelerate 2D DT and showed that their algorithm has an approximate linear time complexity for randomly distributed points. Fang's algorithm is extended to generate DT in our algorithm. Without explicit explanation, we suppose model with constraints has perturbation involved and is grazeable edge protected. In this section, a model is said to have a DT when the model has a CDT whose tetrahedra are all Delaunay. Visibility determination is the most time consuming procedure of CDT. In order to reduce the number of visibility tests between model vertices with respect to model faces, a sufficient and necessary condition for existence of DT is put forward in our advancing front algorithm. To utilize this condition, all model faces are classified into two categories: Delaunay faces (also strongly Delaunay) and undelaunay faces. A model is face protected if all its faces are strongly Delaunay. To test whether a model is face protected is just like edge protection test in Section 2. A model $\mathbf{X}$ with constraints has a DT (also has a unique DT) if $\mathbf{X}$ is face protected and grazeable edge protected. If a model $\mathbf{X}$ has a DT, DT can be obtained from DT of model vertices by a labeling procedure; otherwise undelaunay faces are first processed in our advancing front algorithm. Boundary and constraint triangles are designated as the initial front. As undelaunay faces are processed, front marches and is updated. If current front is face protected during mesh process, Delaunay tetrahedra enveloped by current front are appended to the tetrahedral list by a labeling procedure. After mechanical model is meshed, quality of the mesh is improved by edge-face swapping and smoothing operations [28–30].

5.1. Tetrahedron generation

Without special explanation, no circumstance where five or more vertices are on a common sphere is considered (for symbolic perturbation has been introduced). Klein [17] presented a formal description of CDT in 2D, which is extended to describe 3D tetrahedral generation in our algorithm. Given a model with faces $\mathbf{F} = \{\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{m-1}\}$ where $\mathbf{f}_i = (\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$, $0 \leqslant i \leqslant m - 1$ and vertices $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}\}$ where $\mathbf{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})$, $0 \leqslant i \leqslant n - 1$, let Ω be the body enveloped by boundary faces except interior constraints and $\mathbf{f}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$ be active front triangle. $\tilde{\mathbf{S}}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4})$ denotes the region strictly inside circumsphere $\mathbf{S}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2},$

$\mathbf{p}_{i_3}, \mathbf{p}_{i_4}$) of tetrahedron $\mathbf{t}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4})$. For face $\mathbf{f}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$, tetrahedron $\mathbf{t}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_4)$ is constrained Delaunay if and only if

(1) $\mathbf{p}_{i_4} \in \mathbf{V}_{i_1 i_2 i_3}$, where $\mathbf{V}_{i_1 i_2 i_3} = \{\mathbf{p} \in \mathbf{P} \mid \mathbf{p}_{i_1} \mathbf{p}_{i_2} \mathbf{p} \subset \Omega, \ \mathbf{p}_{i_2} \mathbf{p}_{i_3} \mathbf{p} \subset \Omega, \ \mathbf{p}_{i_3} \mathbf{p}_{i_1} \mathbf{p} \subset \Omega\}$ and
(2) $v\mathbf{S}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4}) \cap \mathbf{V}_{i_1 i_2 i_3} = \emptyset$.

Let

$$\mathbf{H}_{i_1 i_2 i_3} = \{\mathbf{p} \mid (\mathbf{p} - \mathbf{p}_{i_1}) \cdot \mathbf{d}_i > 0\}$$

be the half-space defined by $\mathbf{f}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$ and front advancing direction $\mathbf{d}_i$. Then we have $\mathbf{V}_{i_1 i_2 i_3} = \mathbf{V}_{i_1 i_2} \cap \mathbf{V}_{i_2 i_3} \cap \mathbf{V}_{i_3 i_1} \cap \mathbf{H}_{i_1 i_2 i_3}$, where $\mathbf{V}_{i_1 i_2}, \mathbf{V}_{i_2 i_3}$ and $\mathbf{V}_{i_3 i_1}$ are the set of vertices $\mathbf{p} \in \mathbf{P}$ that are visible from $\mathbf{p}_{i_1} \mathbf{p}_{i_2}, \mathbf{p}_{i_2} \mathbf{p}_{i_3}, \mathbf{p}_{i_3} \mathbf{p}_{i_1}$, respectively, with respect to $\mathbf{F}$. If a vertex satisfying (1) is found, then vertex satisfying (1) and (2) can be found by repeatedly replacing $\mathbf{p}_{i_4}$ with vertex $\mathbf{p} \in \tilde{\mathbf{S}}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4}) \cap \mathbf{V}_{i_1 i_2 i_3}$.

Uniform grid is used to accelerate these three steps (Section 4).

5.2. A sufficient and necessary condition for DT existence

In order to reduce the number of visibility tests between model vertices, a sufficient and necessary condition for DT existence given by Radovitzky is that all boundary triangles have empty minimal circumspheres [10]. But this condition is so strict that few bodies can fit it without local transformations and subdivision. Here, we present a sufficient and necessary condition for DT existence. Given a model with surface triangulation, a necessary and sufficient condition for triangular face $\mathbf{f}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$ to be strongly Delaunay is that there exists a vertex $\mathbf{p}_{i_4}$, where

(1) $\mathbf{p}_{i_4} \in \mathbf{V}_{i_1 i_2 i_3}, \mathbf{V}_{i_1 i_2 i_3} = \{\mathbf{p} \in \mathbf{P} \mid \mathbf{p}_{i_1} \mathbf{p}_{i_2} \mathbf{p} \subset \Omega, \ \mathbf{p}_{i_2} \mathbf{p}_{i_3} \mathbf{p} \subset \Omega, \ \mathbf{p}_{i_3} \mathbf{p}_{i_1} \mathbf{p} \subset \Omega\}$ and
(2) $(\mathbf{S}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4}) \cup \tilde{\mathbf{S}}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4})) \cap \mathbf{P} = \{\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4}\}$.

If all triangular faces of $\mathbf{F}$ are strongly Delaunay, we call the model is face protected.

Theorem 1. *If and only if a model $\mathbf{X}$ is grazeable edge protected and face protected, $\mathbf{X}$ has a DT.*

Proof. Grazeable edge protection ensures that dangling edges (not included in any face) do not violate Delaunay property. If all faces are strongly Delaunay, they must exist in DT of model vertices. Therefore, Delaunay tetrahedra enveloped by boundary faces constitute a DT of the model. The condition is also necessary for existence of DT, provided that no five vertices are cospherical. Because no five vertices are cospherical, Delaunay edges are also strongly Delaunay. If the model

is not grazeable edge protected, there exists an edge that does not have an empty circumsphere. Tetrahedron of CDT including this edge also does not have an empty circumsphere; thus, the model does not have a DT. If the model is not face protected, while there exists a face that does not have an empty circumsphere, tetrahedron derived from the face is also not Delaunay. So the model also does not have a DT. □

Test of face protection is just like test of grazeable edge protection [6]. DT of model vertices is first constructed before CDT. If all model faces exist in DT of model vertices, the model is face protected. To utilize this condition, all model faces are classified into two categories: Delaunay faces (also strongly Delaunay) and undelaunay faces. If all model faces are all strong Delaunay, from Theorem 2, the model has a DT. DT can be obtained from DT of model vertices by a labeling procedure. When the model is not face protected, the following steps are performed:

1. Calculate the number of undelaunay faces by search for faces of $\mathbf{F}$ in DT and choose an arbitrary undelaunay face from initial front as active front face.
2. For the active front face $\mathbf{f}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3})$, find the fourth vertex $\mathbf{p}_{i_4}$ with the following properties:
   (1) $\mathbf{p}_{i_1} \mathbf{p}_{i_2} \mathbf{p}_{i_4} \subset \Omega, \ \mathbf{p}_{i_2} \mathbf{p}_{i_3} \mathbf{p}_{i_4} \subset \Omega$ and $\mathbf{p}_{i_3} \mathbf{p}_{i_1} \mathbf{p}_{i_4} \subset \Omega$;
   (2) $\mathbf{S}(\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}, \mathbf{p}_{i_4}) \cap \mathbf{V}_{i_1 i_2 i_3} = \emptyset$.
3. Update front and the undelaunay face number according to three generated faces (Section 5.3).
4. Choose an undelaunay face from current front as active front face and repeat Steps 2–4 until the undelaunay face number becomes zero.
5. CDT of the current front is obtained from DT of model vertices by a labeling procedure.

The technique can greatly accelerate CDT especially for model with almost regular boundary faces that is also a prerequisite for tetrahedral generation in FEM. For clarity, a 2D example is given (Fig. 5). In 2D, a PSLG has a CDT whose simplexes are all Delaunay if every edge is strongly Delaunay. Thus, Theorem 2 can also be applied to 2D CDT.

5.3. Front update

As a new tetrahedron is formed, three faces are generated and front advances. The active triangle is always deleted from the front and the undelaunay face number is decreased by 1. The undelaunay face number is updated as follows: for each generated face, if the face is

1. a new undelaunay face, the number is increased by 1,
2. a new Delaunay face, the number is unchanged,
3. an undelaunay face that already exists in the front, the number is decreased by 1,
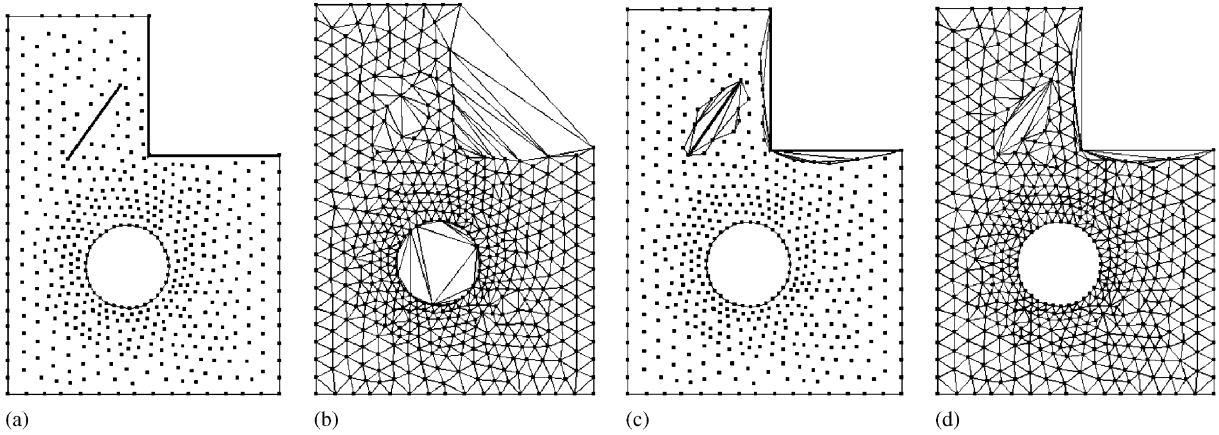
Fig. 5. Procedure of advancing front algorithm utilizing Theorem 2 in 2D: (a) PSLG **X** with undelaunay edges (bold lines); (b) Delaunay triangulation of **X**; (c) stage of CDT with all undelaunay edges processed and (d) CDT of **X**.
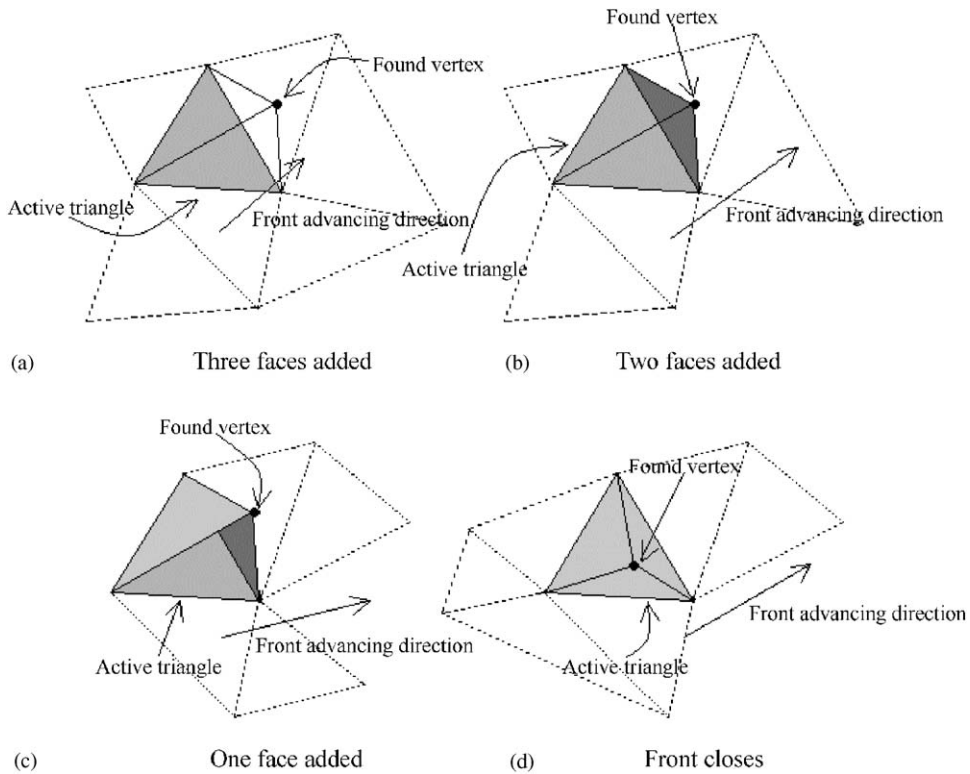

Fig. 6. Front update: (a) three faces added; (b) two faces added; (c) one face added and (d) front closes.

4. a delaunay face that already exists in the front, the number is unchanged.

The front is updated as follows: if the vertex found in Section 5.1 is

1. not a vertex of front triangles adjacent to the active triangle, three triangles are added to the front, Fig. 6(a),

2. a vertex of one front triangle adjacent to the active triangle, two faces are added to the front and the adjacent triangle is deleted from the front, Fig. 6(b),

3. a vertex of two front triangles adjacent to the active triangle, one face is added to the front and the adjacent two triangles are deleted from the front, Fig. 6(c),

4. a vertex of three front triangles adjacent to the active triangle, no faces are added and three adjacent triangles are deleted, Fig. 6(d).

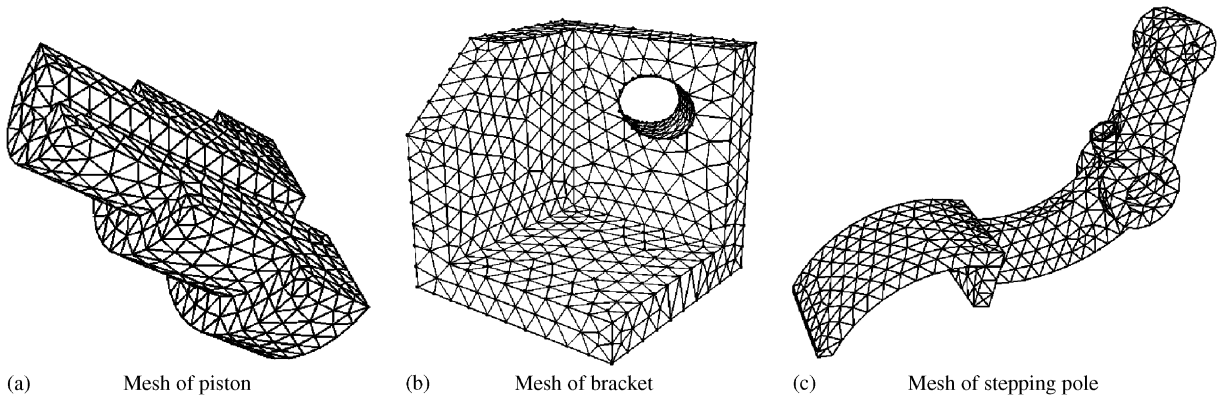|		
(a) Mesh of piston	(b) Mesh of bracket	(c) Mesh of stepping pole

Fig. 7. Tetrahedral mesh of three examples: (a) mesh of piston; (b) mesh of bracket and (c) mesh of stepping pole.

Table 1
Time comparison for three models

	Piston	Bracket	Stepping pole
Delaunay triangulation time(s)	25.53	33.14	35.83
CDT time(s) with face protected check	26.01	34.26	37.14
CDT time(s) without face protected check	602.25	607.15	610.94

6. Examples

In this section several examples (Fig. 7) are given which emphasis the performance of Theorem 2 and the quality of the resulting meshes. Table 1 shows the difference of CDT with face protection check and CDT without face protection check. CDT with face protection check is much faster than CDT without face protection check. The run time of CDT with face protection check and DT is of the same order of magnitude for given models while CDT without face protection check is time consuming relatively.

Fig. 8 shows the run time of CDT with face protection check and CDT without face protection check as well as their linear fittings for mechanical models with uniform density distribution. It can be seen that CDT utilizing uniform grid technique exhibits a linear time complexity for mechanical models with uniform density distribution. Furthermore, CDT with face protection check is much faster than CDT without face protection check.

In order to assess the quality of the meshes, the following tetrahedron quality measure has been adopted:

$$r = \frac{((1/6)\sum_{i=1}^{6} S_i^2)^{3/2}}{8.48528V},$$

where S_i is the tetrahedron edges, V is the volume of the tetrahedron and the factor 8.48528 normalizes the quality of a regular tetrahedron to one [10]. Figs. 9–11 show the quality comparison of the three resulting
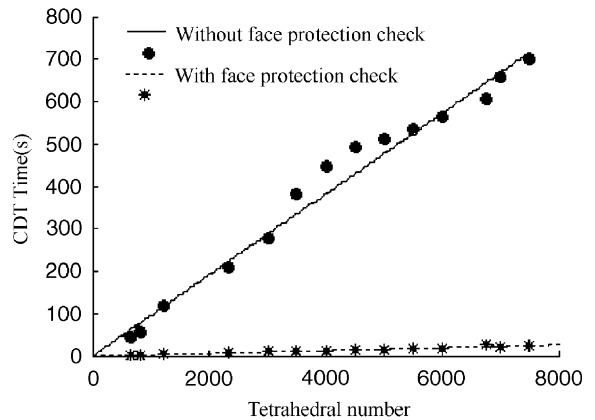

Fig. 8. Comparison between CDT time(s) with face protection check and CDT time(s) without face protection check for some mechanical models with uniform density distribution.

meshes between presented algorithm and Shewchuk's algorithm. The superiority of presented algorithm over Shewchuk's algorithm is evident from these figures. The algorithm introduces the Shewchuk's method into Radovitzky's algorithm to avoid its failure cases while the superior mesh quality is preserved. Furthermore, the linear time property of presented algorithm is absent from the two algorithms mentioned above.

All the above algorithms are run in the environment with Intel Pentium IV CPU 2.0 GHz, 256 Mb, Microsoft Windows 2000 and Microsoft Visual C + + 6.0.
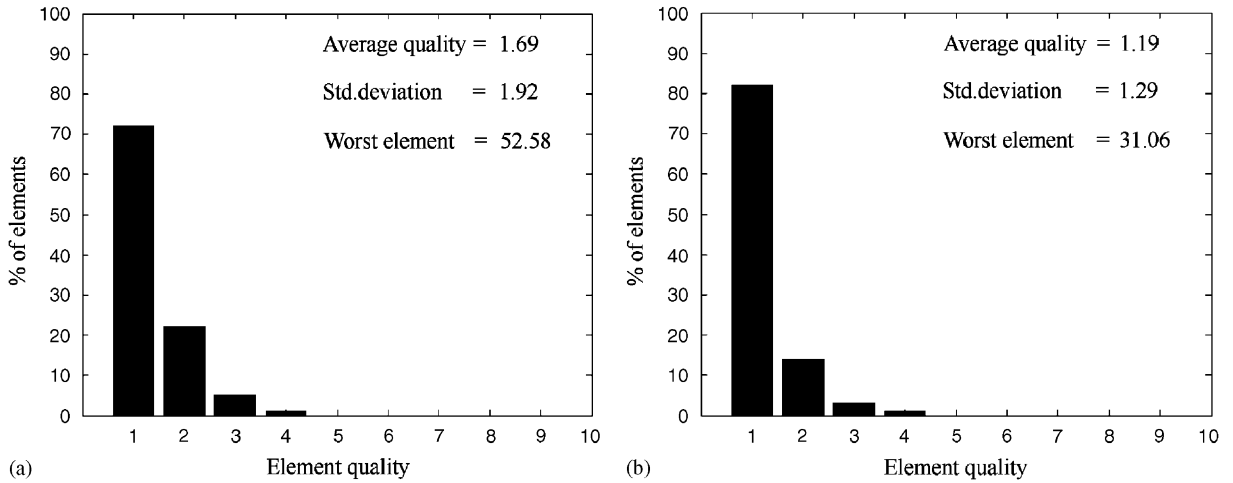
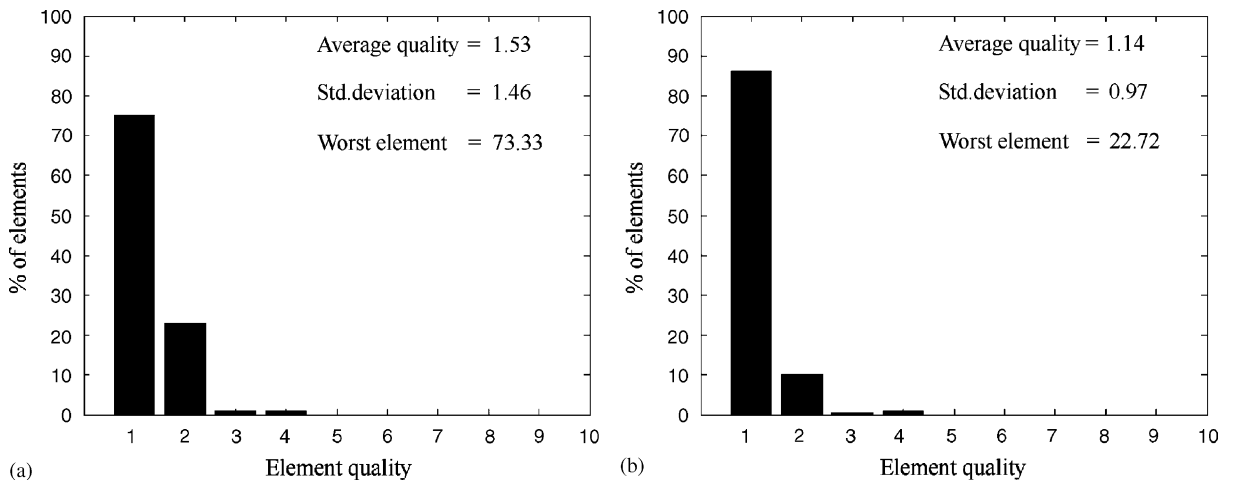Fig. 9. Mesh quality statistics of piston: (a) Shewchuk's algorithm and (b) presented algorithm.

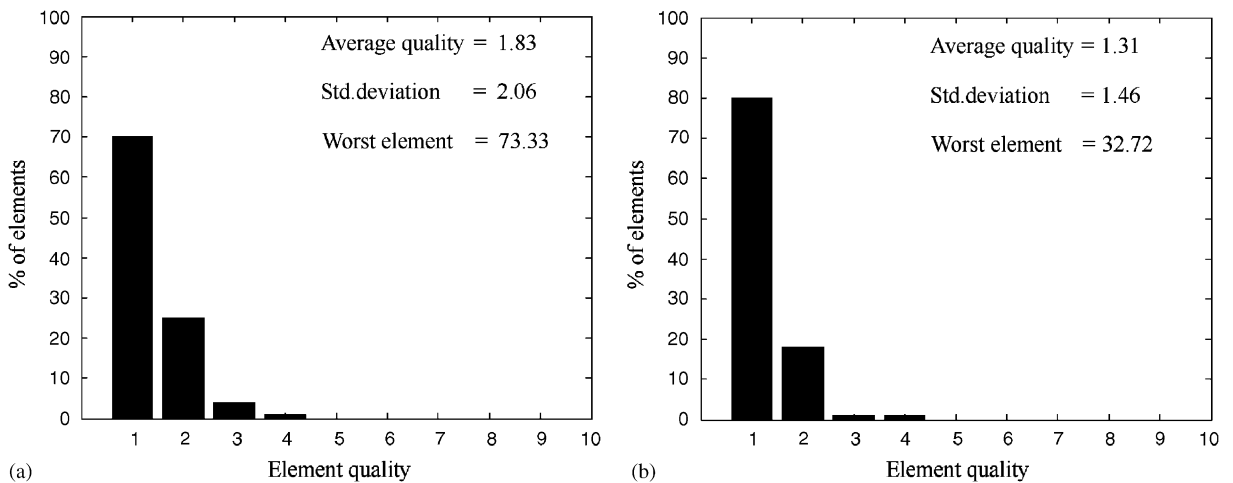Fig. 10. Mesh quality statistics of bracket: (a) Shewchuk's algorithm and (b) presented algorithm.

Fig. 11. Mesh quality statistics of bikestep: (a) Shewchuk's algorithm and (b) presented algorithm.

7. Summary

We have presented a tetrahedral mesh generation algorithm based on CCDT for 3D models with constraints. First, a surface meshing algorithm is performed to construct high-quality surface mesh and perturbation is introduced to simulate the circumstance where no five points are cospherical. Missing grazeable edges of the model are identified and bisected repeatedly until each grazeable edge is represented as set of continuous edges of DT. Surface triangulation on same plane is modified to satisfy constrained Delaunay property. Second, interior points are inserted in a FCC crystal lattice arrangement coupled to octree spatial subdivision. Finally, CDT of the model is constructed via an advancing front method. In order to reduce the number of visibility tests between model vertices, uniform grid is utilized and a sufficient and necessary condition for existence of DT is put forward. The mesh generator is robust and generates good-quality mesh. Furthermore, the CDT algorithm exhibits a linear time complexity for mechanical models with uniform density distribution.

Acknowledgements

References

[1] Susanne C, Brenner L, Scott LR. The mathematical theory of finite element methods. Berlin: Springer; 1994.

[2] Martins J, Pires E, Salvado R, Dinis P. A numerical model of passive and active behavior of skeletal muscles. Computer Methods in Applied Mechanics and Engineering 1998;151:419–33.

[3] Bro-Nielsen M, Cotin S. Real-time volumetric deformable models for surgery simulations using finite elements and condensation. Computer Graphics Forum 1996; 15(3):57–66.

[4] Ganovelli F, Cignoni P, Montani C, Scopigno R. A multiresolution model for soft objects supporting interactive cuts and lacerations. Computer Graphics Forum 2000;19(3):271–82.

[5] Farin G. Geometric modeling: algorithms and new trends. Philadelphia, PA: SIAM; 1987.

[6] Shewchuk JR. Constrained Delaunay tetrahedralizations and provably good boundary recovery. In: Eleventh international meshing roundtable. Ithaca, New York; 2002. p. 193–204.

[7] Cheng SW, Poon SH. Graded conforming Delaunay tetrahedralization with bounded radius-edge ration. Proceedings of the fourteenth annual ACM–SIAM symposium on discrete algorithms. 2003. pp. 295–304.

[8] Li XY. Spacing control and sliver-free Delaunay mesh. Proceedings of the ninth international meshing roundtable. 2000. p. 295–306.

[9] Murphy M, Mount DM, Gable CW. A point-placement strategy for conforming Delaunay tetrahedralization. Proceedings of the eleventh annual symposium on discrete algorithms. 2000. p. 67–74.

[10] Radovitzky R, Ortiz M. Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front Delaunay triangulation. Computer Methods in Applied Mechanics and Engineering 2000;187: 543–69.

[11] Shewchuk JR. Tetrahedral mesh generation by Delaunay refinement. Proceedings of the fourteenth annual symposium on computational geometry. 1998. p. 86–95.

[12] Fuchs A. Automatic grid generation with almost regular Delaunay tetrahedra. Proceedings of the seventh international meshing roundtable. 1998. p. 133–48.

[13] Molino N, Bridson R, Teran J, Fedkiw R. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. Twelfth international meshing roundtable. 2003. p. 103–14.

[14] Perucchio R, Saxena M, Kela A. Automatic mesh generations from solid models based on recursive spatial decompositions. International Journal for Numerical Methods in Engineering 1989;28:2469–501.

[15] Schroeder WJ, Shephard MS. A combined octree/Delaunay method for fully automatic 3-d mesh generation. International Journal for Numerical Methods in Engineering 1990;29:37–55.

[16] Chew LP. Constrained Delaunay triangulations. Algorithmica 1989;4:97–108.

[17] Klein R. Construction of the constrained Delaunay triangulation of a polygonal domain. CAD Systems Development 1995:313–26.

[18] Lee DT, Lin AK. Generalized delaunay triangulations for planar graphs. Discrete & Computational Geometry 1986;1:201–17.

[19] Schönhardt E. Über die Zerlegung von Dreieckspolyedern in Tetraeder. Mathematische Annalen 1928;98: 309–12.

[20] Edelsbrunner H, Mücke EP. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. ACM Transactions on Graphics 1990;9(1): 66–104.

[21] Shephard MS, Georges MK. Automatic three-dimensional mesh generation by the finite octree technique. International Journal for Numerical Methods in Engineering 1991;32(4):709–49.

[22] Cavendish JC, Field DA, Frey WH. An approach to automatic three-dimensional finite element mesh

generation. International Journal for Numerical Methods in Engineering 1985;21:329–47.

[23] Bentley JL, Weide BW, Yao AC. Optimal expected-time algorithms for closest point problems. ACM Transactions on Mathematical Software 1980;6(4):563–80.

[24] Fang TP, Piegl LA. Delaunay triangulation using a uniform grid. IEEE Computer Graphics and Applications 1993:36–47.

[25] Bowyer A. Computing dirichlet tessellations. Computer Journal 1981;24(2):162–6.

[26] Kenneth L, Clarkson, Peter WS. Applications of random sampling in computational geometry II. Discrete & Computational Geometry 1989;4(1):387–421.

[27] Watson DF. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. Computer Journal 1981;24(2):167–72.

[28] Freitag LA, Carl OG. A comparison of tetrahedral mesh improvement techniques. Proceedings of the fifth international meshing roundtable. 1996. p. 87–100.

[29] Joe B. Construction of three-dimensional improved-quality triangulations using local transformations. SIAM Journal on Scientific and Statistical Computing 1995;16(6):1292–307.

[30] Joe B. Three-dimensional triangulations from local transformations. SIAM Journal on Scientific and Statistical Computing 1989;10:718–41.