An Algorithm of Combining of Advancing Front and Delaunay Triangulation to 3-D Domain Tesselation

Jan Kucwaj[*], Barbara Borowik[†]

Institute of Computer Modelling, Cracow University of Technology, Warszawska 24, 31-150 Cracow, Poland

Abstract

The paper presents an algorithm of volume meshing by using the Advancing Front Technique (AFT) combined with the Delaunay Triangulation. The tetrahedronization starts with the surface mesh with the elements oriented towards the interior 3-D domain. The main idea is based upon AFT, with simultaneous points insertion and tetrahedra creation. The characteristic feature of the approach is the part of AFT in the case, when a new calculated point on the current face of the front is not accepted then the existing point in the front is found to create a new tetrahedron by using the Delaunay triangulation on the given set of points. Additionally the algorithm takes a mesh size function into account.

1. Introduction

The purpose of the paper is to describe a method of grid generation for 3-D domains. The problem plays an important role in finite element applications. According to the generation methods, we have the following classification:
• structured grid generation techniques,
• unstructured grid generation techniques.

---

[*]*E-mail address:* jkucwaj@pk.edu.pl
[†]*E-mail address:* bborowik@pk.edu.pl

Generation of structured grids takes less computer time, but the range of geometries is restricted. The only possibility to control the size of the meshes is to place points over the surfaces, and then to fix the height of every tetrahedron.

In the case of unstructured volume grids, if the meshes generated over the surfaces have the required sizes, then by using the Advancing Front Technique the placement of new points can be controlled. Additionally, the Advancing Front Technique gives the possibility of terahedronization of complex geometries [**1**]. Unfortunately, the algorithms of generation of unstructured grids are much more complicated. The program performance is much more time consuming. However, when the Advancing Front Technique is combined with the Delaunay triangulation, then very neatly looking meshes can be obtained. Recently, these methods have been investigated by Lo, George, Frey, Borouchaki [**2, 3**].

The approach presented here is based mainly upon AFT with simultaneous points insertion and triangulation [**4, 5**], but when different parts of the front "meet" and the point to be inserted at this time is not accepted, then on the considered triangle of the front the tetrahedron is built by using as a lacking vertex the point from the front. It is done by using the idea of the Delaunay Triangulation.

It is not guaranteed that the proposed approach leads to complete Delaunay meshes, but usually in the application of AFT the most delicate point is to "finish" the front, just in this case the Delaunay approach is applied giving optimal meshes on the given set of points.

Unstructured grid generation is connected with creation of a mesh on the boundary of the domain [**6, 3**], i.e. in this case, a closed surface or a disjoint union (such as surfaces). If the domain is topologically and geometrically complicated, the problem of surface grid generation becomes more complex.

In this paper the method of Advancing Front Approach coupled with the Delaunay triangulation [**5, 7**] is proposed. In the case, when eventually two different parts (faces) of the front are close enough in the sense of a defined threshold value, for the sake of a new tetrahedron insertion the existing point on the front is used (a new point is not inserted). The point is selected using the Delaunay technique approach. As the criterion of points insertion the mesh density function is used.

The new elements of the paper are:
• the method of point creation over the chosen face of the front for a new tetrahedron insertion,
• the algorithm of choosing points of the front for the sake of a new element creation,
• taking the mesh density function into account.

2. Domain representation and data structure

In this section the following notations will be used:

F – face,
T – tetrahedron.

It is assumed that there exists generated mesh on the boundary surface of the three dimensional domains Ω. The boundary surface is represented by the following sets of data:

$\mathcal{P}_b = \{\mathbf{P}_i : \quad i = 1, \ldots, N_{\mathcal{P}}\}$ – the set of boundary points,
$\mathcal{F}_b = \{F_i : \quad i = 1, \ldots, N_{\mathcal{F}}\}$ – the set of triangles on the boundary surface Ω.

For the sake of the data structure description every boundary triangle is represented by three boundary points from $\mathcal{P}_b$. In other words, an arbitrary surface element is represented as a triple of integer numbers, being the number of its vertices.

Furthermore, for the representation volume mesh the following sets are introduced:

$\mathcal{F} = \{F_i : \ i = 1, \ldots, N_{\mathcal{F}}\}$ – the set of triangles being the faces of the tetrahedra, Ω,
$\mathcal{T} = \{T_i : \ i = 1, \ldots, N_{\mathcal{T}}\}$ – the set of the tetrahedra filling Ω.

In the data structure every tetrahedron is represented by its numbers of the four faces. Additionally, the following sets necessary for the algorithm representation are introduced:
$\mathcal{B}$ – the generation front, at the start consisting of boundary faces,
$\mathcal{E}$ – the set of edges of $\mathcal{B}$.

3. A boundary surface orientation

A single triangle on the boundary surface can be oriented as every orientable surface by its normal (Fig. 1). According to the general algorithm of tetrahedronization, it is necessary to make orientation with normal towards the interior of the domain Ω. The applied algorithm is based on the papers of [**8, 9**].
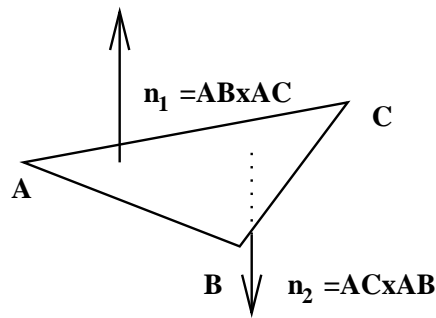
Fig. 1. A single triangle orientation

It is based on establishing the orientation of the arbitrarily chosen surface face. Then orientations of the rest of the surface triangles are adapted progressively to that one. For the algorithm realization, the Advancing Front Technique is used (AFT).

4. The orientation of an arbitrary surface triangle

If we have an arbitrary triangle it is checked whether the normal to the face determined by its orientation is inward or outward the domain Ω. The half-line predicted by this normal is led from the barycenter of the triangle. If the half–line intersects the boundary surface the odd number of times (Fig. 2), then the normal is inward the domain; otherwise it is outward.
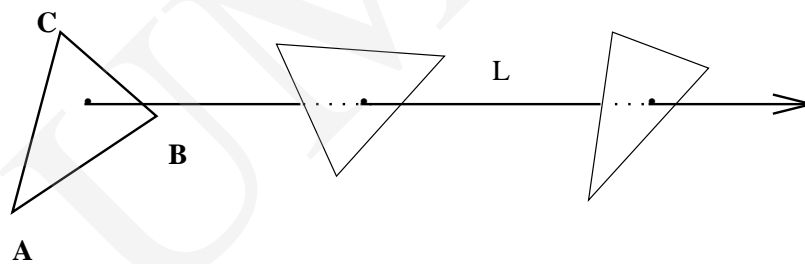

Fig. 2. The intersection of a half-line coming from the $\triangle$**ABC**

As it was previously mentioned the triangle T is represented by the sequence of its three points' numbers, i.e. $T = klm$. The change of the T orientation consists in the transposition of any of the pairs of numbers from the set $\{k, l, m\}$. For example, in this case the following change can be done $klm \longrightarrow kml$.

5. A unification of all the boundary triangles

To adapt the boundary surface to AFM for the three-dimensional domain triangulation, it is necessary that every triangle of the surface would be directed inward the domain Ω (which is a direction of the normal vector). The

closed external surface should have all the triangles directed inward the domain surrounded by the surface, thus the internal closed surfaces representing holes, outward the volume surrounded by them.

The algorithm of unification of triangle orientations is due to one closed connected boundary component. It is assumed that there exists a triangle T_0 (Fig. 3) with a requested orientation. The presented method is the modification of the method from paper [9], and is based mainly on AFM. The initial front consists of three edges of the triangle T_0, then all the triangles, which have as an edge one of the frontal edges are checked and if necessary then reoriented. The frontal edges adjacent to the verified triangles are removed from the front and the adjacent edges of those triangles are added to the front. This process will be repeated until all the triangles are verified.

The modification of the algorithm from paper [9] is based on the fact that the process continues till all the faces are reviewed, contrary to the condition when the front becomes an empty set. This modification allows to uniformize as well the open surfaces. Both these algorithms have the same computational complexity.
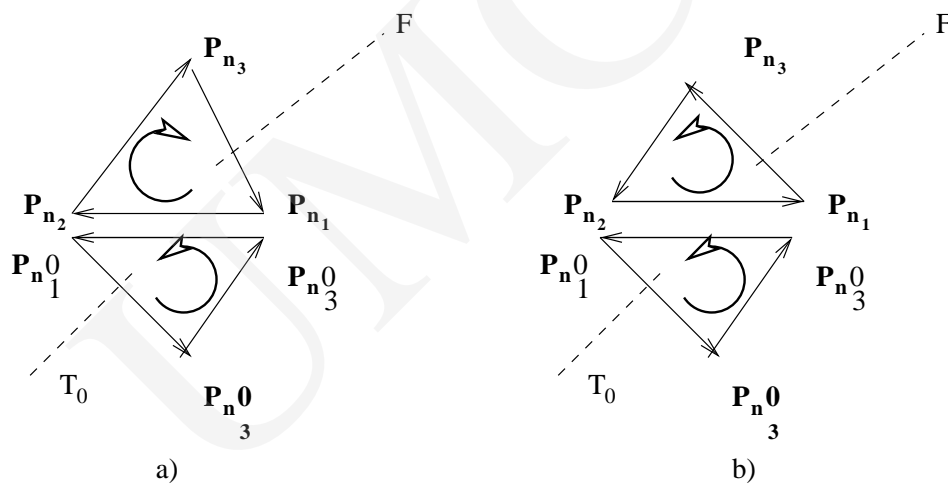

Fig. 3. a) Change to the orientation of T b) No change to the orientation of T

The summary of the algorithm for the uniform front orientation is as follows:

Algorithm 1. 1. $U \longleftarrow \mathcal{T}_b$.
2. $U \longleftarrow U \setminus \{F_0\}$.
3. Find the set $M = L \bigcap \bigcup_{F \in \mathcal{T}_b} \overline{F}$.
4. IF($\#M$ is an even number) THEN change the orientation
   of $T_0 = (n_1^0, n_2^0, n_3^0)$ onto $T_0 \leftarrow (n_1^0, n_3^0, n_2)$
   ENDIF

5. Take as the initial front Γ the set of the edges T_0
   $$\Gamma = (\mathbf{P}_{n_1^0}\mathbf{P}_{n_1^3}, \mathbf{P}_{n_3^0}\mathbf{P}_{n_1^2}, \mathbf{P}_{n_2^0}\mathbf{P}_{n_1^1}).$$
6. IF ($U \neq \emptyset$) THEN
   choose the edge $\mathbf{l} \in \Gamma$, let $\mathbf{l} = \mathbf{PQ}$
   ELSE
   Finish the unification process.
   ENDIF
7. Find such a triangle $F \in U$ that the vector $\mathbf{l}$, taken as an edge
   would be the edge of F,
   let $F = \mathbf{P}_{n_1^0}\mathbf{P}_{n_1^2}\mathbf{P}_{n_1^3}$.
8. IF (vector $\mathbf{l} = \mathbf{P}_{n_1^0}\mathbf{P}_{n_1^2} \lor \mathbf{l} = \mathbf{P}_{n_1^0}\mathbf{P}_{n_1^3} \lor \mathbf{l} = \mathbf{P}_{n_3^0}\mathbf{P}_{n_1^1}$) THEN
   change the orientation of F, i.e. $F \longleftarrow (n_1, n_3, n_2)$,
   ENDIF
9. $\Gamma \longleftarrow \Gamma \setminus \{\mathbf{l}\}$.
10. $U \longleftarrow U \setminus \{F\}$.
11. **Modify the front**
    (a) *IF* ($\mathbf{l} = \mathbf{P}_{n_1}\mathbf{P}_{n_2} \lor \mathbf{l} = \mathbf{P}_{n_2}\mathbf{P}_{n_1}$) *THEN*
        $\Gamma \longleftarrow \Gamma \bigcup \{\mathbf{P}_{n_1}\mathbf{P}_{n_3}, \mathbf{P}_{n_3}\mathbf{P}_{n_1}\}$
        ENDIF.
    (b) *IF* ($\mathbf{l} = \mathbf{P}_{n_1}\mathbf{P}_{n_3} \lor \mathbf{l} = \mathbf{P}_{n_3}\mathbf{P}_{n_1}$) *THEN*
        $\Gamma \longleftarrow \Gamma \bigcup \{\mathbf{P}_{n_1}\mathbf{P}_{n_2}, \mathbf{P}_{n_2}\mathbf{P}_{n_1}\}$
        ENDIF.
    (c) *IF* ($\mathbf{l} = \mathbf{P}_{n_2}\mathbf{P}_{n_3} \lor \mathbf{l} = \mathbf{P}_{n_3}\mathbf{P}_{n_2}$) *THEN*
        $\Gamma \longleftarrow \Gamma \bigcup \{\mathbf{P}_{n_2}\mathbf{P}_{n_1}, \mathbf{P}_{n_1}\mathbf{P}_{n_2}\}$
        ENDIF.
12. go to 6.

6. The algorithm of volume tetrahedronization

Using the notations from the previous point the algorithm 3-D domain triangulation can be presented as follows:

Algorithm 2. 1. Adapt the orientation of the boundary faces inward the domain Ω, according to the algorithm from the previous point.
2. $\mathcal{F} \longleftarrow \mathcal{F}_b$.
3. Find all the edges of the boundary faces.
4. $\mathcal{B} \longleftarrow \mathcal{T}_b$.
5. IF($\mathcal{B} \neq \emptyset$) THEN
   (a) Choose the last triangle from the front $\mathcal{B}$,
       let $F = \mathbf{ABC}$,

(b) Find point **D** that is on the left side of F (which has the inward direction) and which satisfies the following condition:

$$||\mathbf{AD}|| = \rho(\mathbf{A}), \quad ||\mathbf{BD}|| = \rho(\mathbf{B}),$$

$$||\mathbf{CD}|| = \rho(\mathbf{C}).$$

(c) IF edges **AD**, **BD**, **CD** do not intersect the front **B**, and the triangles **ABD**, **BCD**, **CAD** do not intersect all the frontal egdes $\mathcal{B}$, i.e. the set $\mathcal{E}_b$ and

$$dist(\mathbf{D}, \mathcal{B}) > \frac{\rho(\mathbf{A}) + \rho(\mathbf{B}) + \rho(\mathbf{C})}{3 \cdot 2}$$

THEN

  i. add **D** the list of internal points **P** and triangles $F_1 = \mathbf{ABD}$, $F_2 = \mathbf{BCD}$, $F_3 = \mathbf{CAD}$, to the list of triangles $\mathcal{T}$.

  ii. add tetrahedron $T = F\,F_1\,F_2\,F_3$ to the list $\mathcal{T}$ of the tetrahedra.

  iii. Update the front $\mathcal{B}$, i.e. remove F from $\mathcal{B}$ and add the faces $F_1\,F_2\,F_3$ do **B**.

  iv. Update the set of the frontal edges $\mathcal{E}_b$ by adding the following edges **AD**, **BD**, **CD**.

OTHERWISE

  i. By using the Delaunay condition find the point $\mathbf{D} \in \mathcal{B}$. Create a tetrahedron which would have the face $F = \mathbf{ABC}$.

  ii. Remove F from $\mathcal{B}$.

  iii. for i=1,2,3
  IF $(F_i \in \mathcal{B})$ *THEN* remove F_i from **B**
  OTHERWISE
  add F to the list of triangles and to the front $\mathcal{B}$
  ENDIF
  end loop i.

  iv. Modify the set $\mathcal{E}_b$ of the frontal edges.
  Let $E_1 = \mathbf{AD}$, $E_2 = \mathbf{BD}$, $E_3 = \mathbf{CD}$.
  for i=1,2,3
  IF ($E_i \notin \mathcal{E}_b$) *THEN* add E_i to the set $\mathcal{E}_b$
  OTHERWISE
  IF(E_i is not an edge of the newly created faces) *THEN*
  remove E_i from the set $\mathcal{E}_b$.
  ENDIF
  ENDIF
  end loop i.

ENDIF

OTHERWISE
Finish the terahedronization.
ENDIF

6.1. The auxiliary test.

With respect to the newly generated points, it is necessary to check, if every point could be accepted. Let us say the point D is being built over the face ABC, to form a tetrahedron with the vertices A, B, C, D. It is necessary to check whether the faces ABD, BCD, CAD intersect the front.

In papers [**10, 4**] the intersection of the tetrahedron faces with every face of the front is also verified.

In this paper the intersection of the edges AD, BD, CD with the frontal faces is performed and the intersection of the front edges with the faces ABD, BCD, CAD as well. It easy to show that the condition ensures no intersection of the newly obtained tetrahedron with the front. It is less time consuming.

Additionally, there is a test, whether the point D is not too close to the front and the following inequality is taken into consideration:

$$dist(D, \Gamma) = dist(D, P) > \frac{\rho(D) + \rho(P)}{3 \times 2},$$

where:
Γ – the generation front,
P – a point belonging to Γ and realizing the distance D to Γ,
$dist(D, \Gamma)$ – a distance from D to Γ.

6.2. Forming a tetrahedron by the Delaunay method.

When the point D on the face ABC is not accepted to form a tetrahedron with the vertices A, B, C, D, then the point from the front is taken to form a tetrahedron which now would have the vertices A, B, C, and the point from the front. The algorithm has the following steps:

1. Find the ball B_{ABCD} passing through the points A, B, C and with radius r equal to:

$$r = \frac{\rho(A) + \rho(B) + \rho(C)}{3}.$$

2. Find the set S of candidate points from the front belonging to the ball B_{ABCD} and lying on the right side of the plane Π_{ABC} passing through the face ABC.

3. in the set S the following relation $\mathcal{R}$ is introduced:

$$X\mathcal{R} \text{ iff } Y \in B_{ABCY} \ \forall X, Y \in S.$$

4. The relation $\mathcal{R}$ is the equivalence relation, not an ordering relation, as every two elements in S are in relation (are comparable), so the set Z of the points from S is found that satisfies the condition:

$$\forall X \in Z \ X\mathcal{R} \ Y \ \forall Y \in S.$$

5. In the set Z find the point D which satisfies the auxiliary tests.
6. Form a tetrahedron with the vertices A, B, C, D and add it to the list.

7. The numerical examples

In this section some examples of the tetrahedronization of 3–D domains with the mesh size function will be presented. The examples illustrated here are obtained in the way that those tetrahedra of triangulated 3–D domain are shown, whose barycenters lie on one side of a given plane.

In Figs 4, 5 the tetrahedronized cube with the mesh density function given by formula (1) is presented.

$$\rho(x, y, z) = 0.33\sqrt{(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 + (z - \frac{1}{2})^2} \tag{1}$$
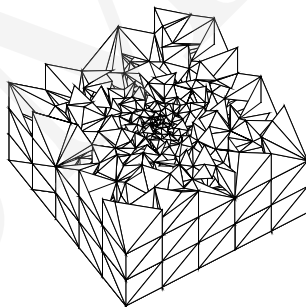

Fig. 4. A cube filled with 4188 tetrahedra, 10629 faces and 1276 points

In Figs 6, 7 the triangulated torus with the constant mesh density function is shown.

In Figs 8, 9 the domain obtained from the union of cylinder and the part of sphere filled with tetrahedra is presented. The size function was given by formula (2):

$$\rho(x, y, z) = \begin{cases} 0.3 \sqrt{x^2 + y^2 + z^2} + 0.02, \text{ if } r < 1.0. \\ 0.2 \quad \text{otherwise.} \end{cases} \tag{2}$$
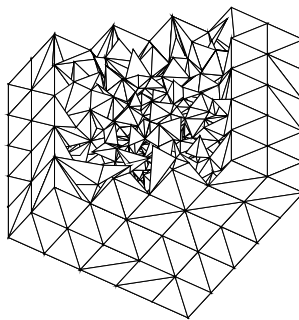
Fig. 5. The meshed domain from Fig. 4 with another sharing plane
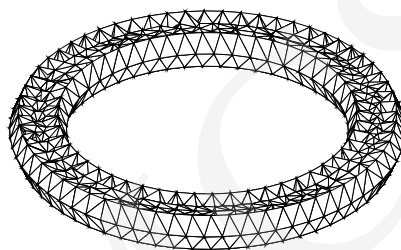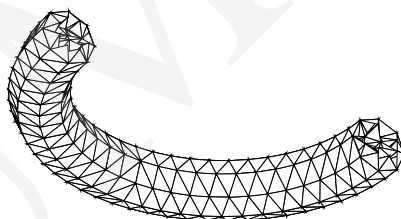

Fig. 6. Mesh in torus

Fig. 7. The tetrahedronized volume from Fig. 6 with another sharing plane
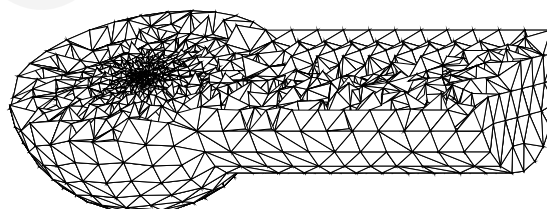

Fig. 8. A cylinder united with a part of the sphere with a symmetrical sharing plane -
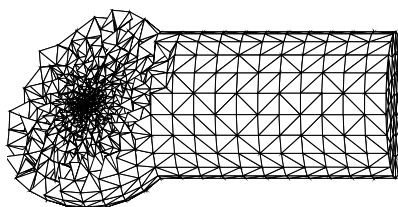80401 faces, 6712 points and 39467 tetrahedra

Fig. 9. The tetrahedronized domain from Fig. 8 with another sharing plane

8. Conclusions

In this paper an algorithm of unstructured grid generation is presented. It is based on coupling of the Delaunay triangulation and Advancing Front Technique. Algorithm starts from the grid generation on the boundary surface and then by using the AFT points are inserted and tetrahedrons are formed. When a newly inserted point does not pass the auxilliary test then the point from the front is chosen using the Delaunay condition. The presented approach takes a mesh size function into account. The paper describes in detail the algorithm implemented by the main author and provides many numerical results. Such a grid generator can be used for an adaptation technique [**11**] based on remeshing for 3–D problems. The remeshing algorithm modifies a mesh size function taking into account an error indicator function.

The original elements of the paper are:

• The method of combining Advancing Front Method and Delaunay Triangulation.

• Taking a mesh size function into account.

• Programming source code written by the main author himself.

• Numerical examples obtained from many programs prepared by the main author.

References

[1] Pippa S., Caligiana G., *GradH-Correction: guaranteed sizing gradation in multi-patch parametric surface meshing*, Int. J. Num. Meth. Engng. 62, (2005) 495.

[2] Borouchaki H., Lo S.H., *Fast Delaunay Triangulation in Three Dimensions*, Comp. Meth. Appl. Mech. Engrg. 128 (1995) 153.

[3] Thompson J.F., Soni B.K., Weatherwill N.P. *Handbook of Grid Generation,* CRC Press, Boca Raton (1999).

[4] Jin H., Tanner R.I., *Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique*, Int. J. Num. Meth. Engng. 36 (1993) 1805.

[5] Krysl P., Ortiz M., *Variational Delaunay approach to the generation of tetrahedral finite element meshes*, Int. J. Num. Meth. Engng. 45 (2001) 1681.

[6] Owen S.J., Saigal S., *Formation of pyramid elements for hexahedra to tetrahedra transition*, Comp. Meth. Appl. Mech. Engng. 190 (2001) 4505.

[7] Kucwaj J., *Delaunay Triangulation of Surfaces*, ZAMM, 76, S.3 (1996) 249.

[8] Lo S.H., *Volume discretization into tetrahedra-I, verification and orientation of boundary surfaces*, Computers and Structures 39(5) (1991) 495.

[9] Lo S.H., *Volume discretization into tetrahedra-II, 3D triangulation by Advancing Front Approach*, Computers and Structures 39(5) (1991) 501.

[10] Armstrong Ch., Ford R.W., Gurd J.R., Luján M., Mayes K.R., Riley G.D., *Performance control of scientific coupled models in Grid environments*, Concurrency and Computation: Practice and Experience 17 (2005) 259.

[11] Kucwaj J., *The Application of Adaptive Algorithm to Potential Problem*, Annales UMCS Informatica AI 2 (2004) 37.