International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:3, No:10, 2009

Development of Improved Three Dimensional Unstructured Tetrahedral Mesh Generator

Ng Yee Luon, Mohd Zamri Yusoff, Norshah Hafeez Shuaib

***Abstract*—**Meshing is the process of discretizing problem domain into many sub domains before the numerical calculation can be performed. One of the most popular meshes among many types of meshes is tetrahedral mesh, due to their flexibility to fit into almost any domain shape. In both 2D and 3D domains, triangular and tetrahedral meshes can be generated by using Delaunay triangulation. The quality of mesh is an important factor in performing any Computational Fluid Dynamics (CFD) simulations as the results is highly affected by the mesh quality. Many efforts had been done in order to improve the quality of the mesh. The paper describes a mesh generation routine which has been developed capable of generating high quality tetrahedral cells in arbitrary complex geometry. A few test cases in CFD problems are used for testing the mesh generator. The result of the mesh is compared with the one generated by a commercial software. The results show that no sliver exists for the meshes generated, and the overall quality is acceptable since the percentage of the bad tetrahedral is relatively small. The boundary recovery was also successfully done where all the missing faces are rebuilt.

***Keywords*—**Mesh generation, tetrahedral, CFD, Delaunay.

I. Introduction

THE mesh quality can have a considerable impact on the computational analysis, for both CFD and FEA, in terms of the accuracy of the solution and the time taken to obtain it. Tetrahedrons formed to fill the computational domain need to be of required quality in term of shapes and sizes. Delaunay triangulation based algorithm is an effective tool in mesh generation. In two-dimension (2D), a set of triangles is a Delaunay triangle when none of the triangle having vertex inside their circum-circle. This criterion is called empty circle properties. The empty circle properties ensure that Delaunay triangle will tend to avoid long and skinny triangle (since they have larger circum-circle), making them good in numerical simulation.

One of the most popular Delaunay triangulation algorithms for two or higher dimensional space was first introduced independently by Bowyer [1] and Watson [2]. In the Bowyer-Watson algorithm, each point is processed one point at a time and any elements that violate the empty circle properties is deleted. The speed of the algorithm depends on how fast the elements that violate the Delaunay criteria can be determined and a new element set is updated. Triangles/tetrahedral generated can be used as a guide to decide where should the next vertex to be inserted. In general, in order to ensure acceptable quality, the vertex should be inserted as far as possible from other vertices [3]. Most of the bad quality elements normally have at least one short edge. Therefore, inserting a vertex further from existing vertices will reduce the likelihood of a new bad element to appear.

Although one can use small size elements to produce as many tetrahedral as they want to capture the critical region, the CPU time for a flow simulation is proportional to the number of grid points raised to the power of 3/2 [4]. Hence putting large number of elements will certainly take more time and resources to solve the problem. Shewchuk used the grading refinement algorithm from Jim Ruppert, and achieved good angle bounding [2].

In this paper a complete three dimensional tetrahedral mesh generation algorithm based on Delaunay triangulation is presented. This includes methods to improve the mesh quality and delete elements outside the domain. Triangular surface mesh of the domain is used as input for the mesher.

II. DELAUNAY TRIANGULATION KERNEL

Following the Bowyer-Watson methodology, the algorithm is started with one large cube with five tetrahedral that enclosed all the input vertices. Point is inserted one at a time. The region affected by this new vertex is determined and a new Delaunay triangulation which contains the newly introduced point is produced. The process can divided into three stages :

(1) BASE searching
(2) Correction of the CORE
(3) Triangulation of the CORE and update

A. BASE Searching

When a new vertex, P is inserted into a set of Delaunay triangles/tetrahedral, some of the cells will violate the empty circle criteria as P is inside their circum-sphere. These tetrahedral are removed from the mesh set and replaced with new tetrahedral set that included point P.

The easiest method is to check for all existing tetrahedral to

Ng Yee Luon is with the OYL Research and Development Sdn. Bhd, Sungai Buloh, Selangor, Malaysia (e-mail: ngyl@oyl.com.my).
Mohd Zamri Yusoff is with the Centre for Advanced Computational Engineering (CACE), College of Engineering, Universiti Tenaga Nasional, 43009 Kajang, Selangor, Malaysia (phone: +603-89287255; fax: +603-89212116; e-mail: zamri@uniten.edu.my).
Norshah Hafeez Shuaib is with the Centre for Advanced Computational Engineering (CACE), College of Engineering, Universiti Tenaga Nasional, 43009 Kajang, Selangor, Malaysia (e-mail: hafeez@uniten.edu.my).

identify which of them contain P inside their circum-sphere. But this is very time consuming process as the number of tetrahedral are large and increasing as the process progress. Furthermore, those tetrahedral containing P only consist of small portion from entire cells (except for some rare cases). Checking all the cells that do not contain P for the inclusion test is not efficient.

To localize the operation, the location of P with respect to the tetrahedral set, need to be determined. Following [5], the tetrahedron which contains the point P is called BASE. The ideal case is to start the search in the vicinity of point P. But, in reality, the location to start the search process is not known.

The search can be started from the last constructed tetrahedron when no information is provided. The search travel from cell to cell, moves across the faces that make it closer to P. The decision to cross the face is only made when the "unique node" (node that not belong to the common face) of the neighboring tetrahedron is on the same side as the point P. That is, the volume sign of the tetrahedron formed by taking common face, F with the "unique node" and point P are the same. The searching is terminated when no "unique node" of the neighbor tetrahedron is on the same side as point P, because point P is inside the current tetrahedron. Fig. 1 shows the search path to the BASE in 2D. The elements outside the path are ignored.

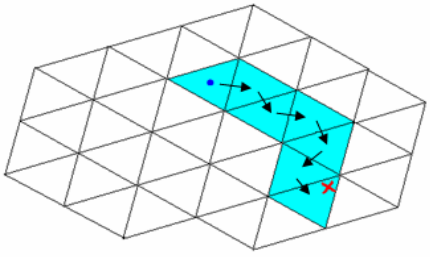

Fig. 1 Search path to BASE. Only highlighted elements are checked. The rest are skipped

B. Correction of the CORE

CORE is the cavity that is left behind after the tetrahedral that violate the empty circle properties are deleted. The CORE can be easily identified when the BASE is known. The search can start from the adjacent cells of the BASE, when they are found to be part of the CORE, their adjacent cells will be checked. The "branch" search process is continued until no new tetrahedral is part of the CORE.

The boundary of the CORE is the faces remain after the tetrahedral is removed. Tetrahedral sharing the face give positive and negative results to sphere inclusion test. The new tetrahedral set is constructed simply by connecting new point to the boundary face of the CORE. But, for some rare situations, the tetrahedron formed using CORE boundary is inverted. Fig. 2 shows one of the case in two dimensional problem where one of the new triangle formed is invalid.

The problem can be solved by applying visibility check. That is, every face from the boundary of the CORE must be visible to point P. The volume sign of the tetrahedron deleted must be the same as the new tetrahedron using the same CORE face if point P can "see" the CORE face. Or in other words, the vector projected to the new vertex cannot cross any CORE boundary. Fig. 3 shows the example of the visibility test. The red point is the newly introduced vertex.

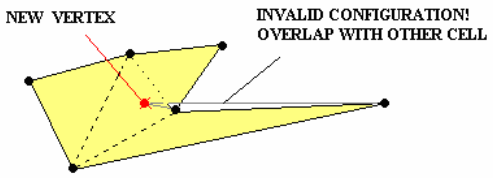

Fig. 2 Invalid elements formed

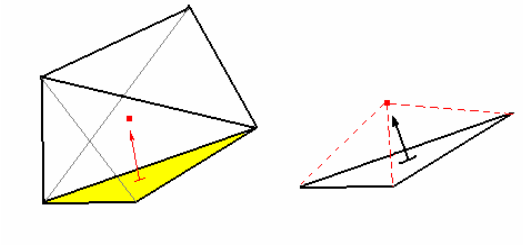

Fig. 3 Visibility Test

If the inverted cell is detected, the CORE boundary face that contributes to this cell is removed by removing tetrahedron from the CORE that contains this face. New CORE boundary face is established and the visibility check continue until all new tetrahedral are valid. This visibility test, although simple, is important to ensure the validity of the CORE. The volume calculated is used to update the volume data later during the triangulation of the CORE.

C. Triangulation of the CORE

New triangles/tetrahedral are created by connecting vertex P to all the CORE boundary faces. Creating the cell is simple. The complicated part is when updating the adjacency data for fast searching later. The adjacency data for CORE boundary is calculated before the actual construction. The common edge of each face pair is also recorded. During actual construction, after a CORE face is used, its adjacent faces immediately appear in "next to use" list. The adjacency data can be updated without the need of search and match since the next three cells built must be the neighbors. If the neighbor's faces are already used, only the adjacency data are updated.

The important parameters that enhance the search such as circum-center and circum-radius are calculated right after new tetrahedron is built. They can be easily calculated based on the same quantities as the old tetrahedron [5] since they are using same CORE face. But, experience shows that the error from previous calculation is accumulated and causing inaccurate result in sphere inclusion test after some iteration. Hence, this was not implemented.

D. Data Structure

In the current mesh generator, double-linked connectivity for face and element is used. In this, each tetrahedron will store the four vertices and four faces. The face stores three vertices and the two tetrahedral that share the face. The reason for storing face data instead of adjacent cell data is for the ease of extraction of output data for the face based solver. Face based solver required information on which cells belong to the face. The current data structure can directly provide the information without the need to generate them from cell information. The adjacent cell of each cell can be obtained through visiting the face used by the tetrahedron.

III. MAINTAINING DOMAIN VALIDITY

Bowyer-Watson algorithm creates a large convex hull that encloses all input data. After input vertices are tetrahedralized, some of the tetrahedral generated are outside the domain and they may not conform to the input boundary. That is, some input triangles are actually missing and result in the surface no longer "closed". The recovery of these missing faces is necessary to maintain the validity of the domain.

A. Boundary Recovery

The missing segment in 2D can be easily recovered through the recursive splitting procedure until the missing segment is represented by few sub segments. Since the Delaunay triangulation always connects a vertex to its nearest neighbor, the entire length can be represented once the vertices spacing along the edge is sufficiently small [3].

But in 3D, a missing face may contain one or more missing edges. In the present algorithm, the missing segments and faces are recovered separately. All the input edges are first checked to know if they already exist. If not, a new vertex is added at the center of the missing edge. This process is looped until no input segment is missing.

To recover the missing faces, all missing faces were first identified by comparing all input faces to all the faces that exist in tetrahedralization data. If no face in tetrahedralization data is matched with the input face, the input face is flagged as missing.

The recovery process is divided into two procedures. First the swapping process is carried out to recover missing faces as much as possible. If the face still cannot be recovered, second procedure is done by adding vertex at the missing face. Shewchuk [3] added the vertex at the circum-center of the missing triangle. In the present procedure, vertex is added at the circum-center only when the circum-center lies inside the triangle. If not, the vertex is added at the intersecting point between the missing triangle and the edge crossing it. Since the triangle is missing, they must be crossed by at least one edge. If not the triangle is actually not missing and is unflagged.

After swapping and re-meshing, the surface may be recovered but in the union of triangles different from the input set. In order to identify which triangle lie on the surface, Shewchuk [3] proposed that two independent separate data sets are maintained for two dimensional and the tetrahedralization data. When vertex is added in the three dimensional data, the two dimensional data is updated with the vertex as well. This will always ensure both triangles in two and three dimensional is comparable, making the identification easy.

Due to the different in input style, the two dimensional data is not maintained in the present code. This problem is countered by maintaining list of "which node is used by which surface". The vertices of new triangle are checked, and all possible surfaces are listed out. The new triangle is inside a surface when its centroid is inside one of the triangles belonging to that surface. When this is true, this new triangle will be flagged as belonging to this surface.

The check and match procedure can be very time consuming since it requires comparisons against all existing faces. The real flow simulation problem can easily produce more than 100,000 faces during initial tetrahedralization. A fast search method is used to limit the number of faces that need to be checked. In this procedure, a new vertex is added at the center of the missing edge or face and the tetrahedron containing this vertex is identified. The edge or face exists only when they exist at this tetrahedron. By doing this, the rest of the faces can be skipped. This method greatly enhanced the speed of the boundary recovery process and no extra algorithm is needed since the searching algorithm is the same with that used in tetrahedralization. The only different is the new vertex is deleted after the search, without any tetrahedralization.

B. Domain Recovery

Since the Delaunay triangulation start by few large cells that contain entire input vertices, unwanted cells are also generated. They are removed before refinement process to avoid unnecessary refinement at the tetrahedral outside domain. Surprisingly, despite its importance, only few of the grid generation papers discuss this topic in depth.

Some of the tetrahedral which lie outside the domain can be easily identified since they use at least one of the non input vertices. The problem arises when the domain shape is non convex or having internal boundary. They cannot be identified through node index since all their nodes are input vertices.

The point in solid test is used to identify tetrahedral which are outside the domain. Taking advantage that the centroid of the tetrahedron will always lies inside itself, it is used as point to test to decide if this tetrahedron is inside the domain. The point in solid test algorithm presently used is based on Jordan Curve Theorem [6]. The theorem says, any simple closed curve C divides the points of the plane not on C into two distinct domains (with no points in common) of which C is the common boundary [7]. So in practice, casting a ray from a point to any direction will give a number of intersections with the boundary. If the intersecting count is odd, the point is inside closed surface. If the count is even or zero, the point is outside. The process is also called "ray-casting". Fig. 4 shows example of the ray casting. Red ray having even intersecting,

hence the starting point of the ray is outside domain (grey region). Blue ray having odd intersecting count (inside) and green ray is special case (parallel with the segment). Another ray must recast to different direction.

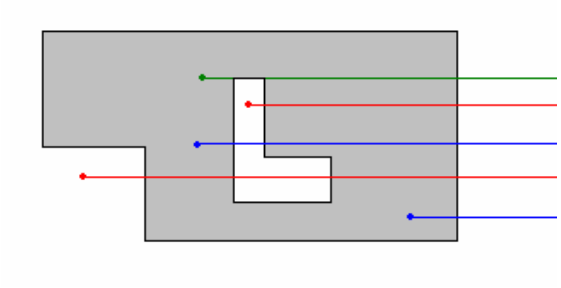

Fig. 4 Five points tested using ray casting

The advantage of the ray-casting method is that it needs no pre-computed data. The input triangles can be used directly as boundary faces. The concept is very simple and easy to implement. The drawback of the ray-casting is uncertainty arise when the ray is intersecting with the edge or vertex, or coplanar face is found. In this case, a new ray must be cast at the different direction until no special case detected. The intersecting calculation must be restarted from beginning when such special case is encountered resulting in waste of time.

The ray casting test can be very time consuming when the input face number is large (which will lead to many tetrahedral generated) and complex (more special case encountered). In order to limit the triangles that need to check for intersecting, only the triangles at the location that possibly crossed by ray will be check for intersection. For example, if a ray starts from point P toward +x direction, triangles that are having all three vertices lie behind point P (in –x direction when viewed from point P) can be discarded, since they are on the opposite side of the ray direction.

The domain recovery process can be even faster if the number of tetrahedral that need to undergo ray-casting check can further reduced. This can be done by taking advantage of the flagged face data from previous boundary recovery process. All triangles that represent the input surface is flagged and internal triangles is flagged as zero (default value). The process can be started from any of the tetrahedron that uses triangle having non zero flagged, which indicate that this tetrahedron is using boundary face. If it lies outside the domain after being tested by ray-casting, all of its neighbors will be put into delete list, except for the neighbor that is sharing boundary face with the current cell. The process is very similar to that described in [8], which employs a “triangle eating virus” at a point specify by user. The “virus” spread around until it stops by segments. The present method uses ray-casting to automatically decide where this “virus” should be planted, hence no user intervention is needed.

This domain recovery method is strongly tied up with the robustness of previous boundary recovery scheme. Even with only one of the surface triangle missing, the “virus” will infiltrate into the domain through the hole on the surface. Hence all tetrahedral is considered outside domain and deleted.

IV. REFINEMENT SCHEME

A. Quality Measurement

In numerical simulation, the shape of the elements affects the accuracy and stability of the simulation. For tetrahedral, large angle will give large interpolation error, which lead to instability of the simulation; while small angle tend to produce ill condition matrix [9]. Hence the refinement scheme must be able to avoid the creation of these two types of elements (often tetrahedron with large angle will contain small angle at the other side also). Quality measurement used must be able to detect such elements.

In 2D, radius to shortest edge ratio is a good quality measurement, since minimizing this ratio will also maximize minimum angle of the triangle. But in 3D this is not always the case. The radius to shortest edge ratio in 3D can detect most of the bad quality tetrahedral, except for sliver, which is the tetrahedron that has its four vertices equally spaced, with one vertex slightly off the equator [3]. Sliver has nearly zero volume, and angle close to 0˚or 180˚. The creation of the sliver is the natural output of the Delaunay triangulation rather than numerical error [5]. Even well spaced input cannot prevent sliver [10,11]. Fig. 5 shows example of some bad elements and sliver. Most of the bad elements except for sliver contain at least one short edge compare to the ideal tetrahedron.

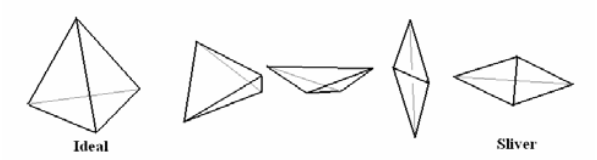

Fig. 5 Bad elements (middle 3) compared to ideal tetrahedron (left) and sliver (right)

Volumetric skewness is the ratio of the cell volume to the maximum possible volume under cells circum-radius. It can detect almost any kind of bad tetrahedron including sliver (near to zero in volumetric skewness). But it is too strict as quality measurement since it also identify tetrahedral that have good angle but skinny as very bad quality. In reality these elements are harmless, except that they are wasting vertices [9]. Attempt to improve them will create even more cells and vertices. Radius ratio (ratio of circum-radius to the inscribe radius) give the similar problem.

B. Vertex Addition

After the initial tetrahedralization and domain recovery, the tetrahedral are refined by inserting vertex at circum-centre of bad quality cells. The process is repeated until all tetrahedral are within the bound of the quality parameter (except those near the boundary, since their circum-centre may be outside of the domain).

Despite the failure to detect sliver, the radius to shortest

edge ratio is still preferred as the primary quality parameter during the mesh refinement. If the maximum radius to shortest edge allowed, is set to 1, the algorithm is guaranteed to terminate when a new vertex at the circum-centre of the elements is inserted, as the edge of the new tetrahedron is at least the same length as the radius of the deleted tetrahedron [3,12]. No edge shorter than pre-existing edge can be created. So, the algorithm will terminate when it run out of space to add new vertex.

"Auto sizing" based on the input surface can also be included. Here, each vertex is assigned a weight based on the length of the edges using the vertex. Later the maximum of the tetrahedral is the average weight of the tetrahedral vertices. Hence the tetrahedron using small input surface has small final size. A weight of the new vertex added (at the circum-centre) is slowly increased by the expansion rate of 1.2 until the maximum allowable size is achieved. Hence elements away from boundary will have uniform size.

C. Topology Transformation

After the vertex insertion process, the only bad tetrahedral lefts are those having high "sliverness". They are improved by applying swapping. Swapping is the process of changing connectivity of the cell while keeping the vertex location static. Swapping is also called topology transformation.

Since sliver cannot be detected through radius to shortest edge ratio, after vertex insertion process the quality parameter used is dihedral angle. Both minimum and maximum dihedral angle is recorded to keep the current quality state of the tetrahedron. The mesh improvement scheme only target elements which have quality worst than a specified angle.

Most of the sliver can be removed by applying 3-2 swap. The diagonal edge of the sliver is used as reference and the two other tetrahedral that are using this edge are found. Later this common edge is swapped hence the sliver will disappear. The swap 3-2 can also be reversed to become swap2-3. In swap 2-3 common face of the two tetrahedral is destroyed and replaced with common edge that are used by three elements. Cheng et al. [10] removed most of the Sliver through this method. Fig. 6 shows the swap 3-2 and swap 2-3 operation.

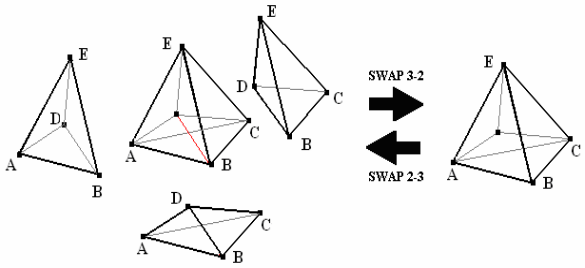

Fig.6 Swap 3-2 and swap 2-3 Operation

Other topology transformations are swap 4-4 and swap 2-2. Swap 4-4 is used to change the common edge connectivity of the 4 tetrahedral without changing the number of elements. Swap 2-2 is very similar to swap 4-4 but working on the common edge that lies on the boundary and shared by two tetrahedral. Fig.7 shows these two operations.

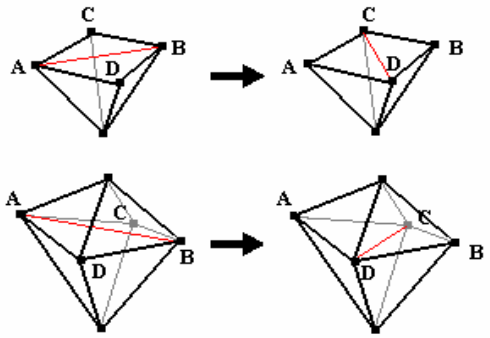

Fig. 7 Swap 4-4 and swap 2-2

The more general topology transformations used for any number of tetrahedral larger than four are edge removal and multi-face removal. They are the reverse of each other. Edge removal replaces common edge of a group of tetrahedral with a few faces created by non common nodes of those elements. Multi-face removal, as the name suggests, remove all the faces between two common nodes and replace with an edge connecting them. Fig. 8 shows these two operations, where common edge AB is replaced with 3 internal faces in edge removal process, and in multi-face removal these three internal faces are deleted and edge AB is restored.

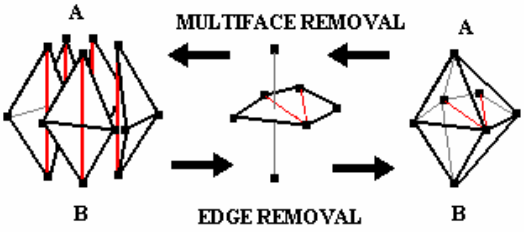

Fig. 8 Multi-face removal and edge removal

Topology transformation is local. It only changes small number of elements around the targeted region. All dihedral angles after transformation are calculated and compared with the value before. The transformation is only accepted when the worst case after swapping is better than before. This is to ensure the mesh can only improved and never become worst.

D. Smoothing

Smoothing is the opposite of swapping. It changes the position of the vertices without touching their connectivity. For triangular and tetrahedral elements, the popular smoothing method is Laplacian smoothing, which move the vertex toward the centroid of the polygon formed by surrounding vertices [9]. With the connectivity table, the surrounding nodes can easily be found and the process is local as in topology transformation. The drawback of Laplacian smoothing is that its quality function of the tetrahedron is not a smooth function of the vertex coordinate. Hence the quality

after smoothing may be worst than before. Pierre et al. [11] uses variational tetrahedral meshing to optimize vertex position by minimizing the energy function, but their mesh generated is not conform to input boundary. Klingner et al. [9] uses non-smooth optimized smoothing to counter this problem, and extends it to boundary smoothing for flat surface.

In the present work, Laplacian smoothing is used due to its speed and ease of implementation. A similar approach as in topology transformation is used to ensure that no element after smoothing is worst than before.

The swapping- smoothing procedure is repeated iteratively. This means that after all tetrahedral are swapped, smoothing is applied to the remaining bad elements that failed to improve by swapping. It was found that, this approach successfully improve many tetrahedral that cannot be improved by swapping or smoothing alone. After smoothing, even though the element quality is still far from ideal, it allowed the swapping operation to be applied on tetrahedron that failed to swap during the last iteration.

V.RESULTS AND DISCUSSION

Two models in CFD problem are tested and compared with results from a popular commercial meshing software to generate tetrahedral elements.

A. Exhaust Tube

The flow inside a typical exhaust tube of an automobile is investigated to find out why the outer tube is dirtier, which means more carbon is passing through outer side of the tube. The geometry and surface mesh is shows in Fig. 9. Uniform surface mesh is used in this case.

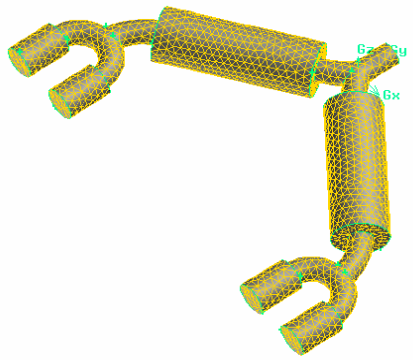

Fig. 9 Exhaust tube and the input surface mesh

Using the same surface mesh, the result obtained from the present mesher and the commercial mesh generator are compared. The elements generated using the present mesher is 11183, while the commercial mesh generator generated 13858 elements. Fig. 10 shows the cutting view of both meshes. Elements form the commercial mesh generator looks slightly smaller. This may explained why their number of elements is higher.

The cutting view shows that the mesh distribution and size is uniform for both cases. No extreme skew or unrefined element is found. A more in-depth quality investigation is done using mesh examination tools. Only tetrahedral having angular skewness larger or equal to 0.6 ($\theta min \leq 28.2^{\circ}$, $\theta max \geq 136.2^{\circ}$) are displayed. Fig. 11 shows the result obtained using the present mesher.

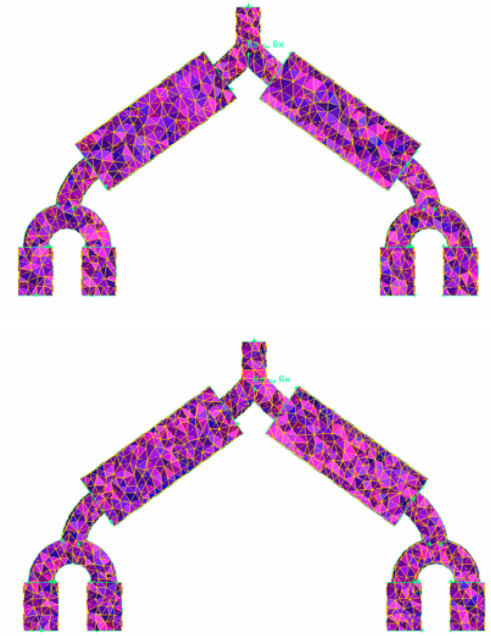

Fig 10. Cutting view of the elements generated. Top: Present Mesh Generator, Bottom: Commercial Mesh Generator

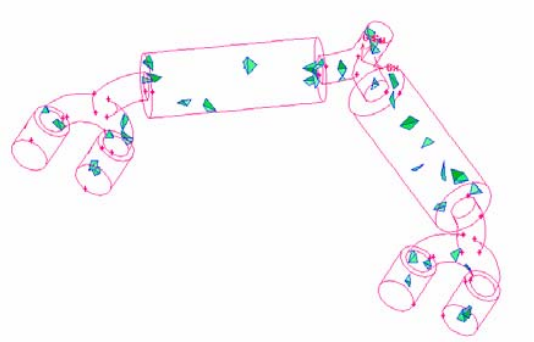

Fig. 11 Bad elements from the present mesh generator

Only 48 elements (0.43%) are having angular skewness higher than 0.6. The worst element detected is 0.69 (21.86°, 146.06°), which is still inside acceptable range. Many of the elements are in between 0.3 (49.37°, 103.37°) and 0.4 (42.32°, 114.32°), consist of 37.71%. The commercial mesh generator generated more tetrahedral higher than 0.6 skewness, with total number of 224 elements (1.62%) with the worst element with skewness of 0.77 (16.22°, 154.82°). As in the present mesh generator, large number of elements is in between 0.3 and 0.4 (38.92%). Fig. 12 shows the result obtained from the commercial mesh generator.

B. Car

Flow over a car is a typical CFD problem in automobile industry. Often, to save the meshing and simulation time and due to the symmetry, only one half of the car will be simulated. The car is placed inside a large domain box. Mesh is generated in the region between surface of the car and domain box boundary. The geometry of the car and is shown in Fig. 13. Fig. 14 shows the entire flow domain.

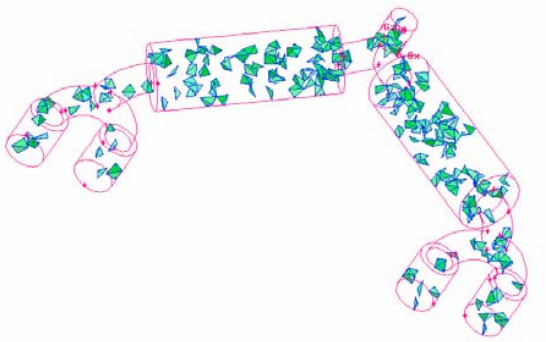

Fig. 12 Bad elements from the commercial mesh generator

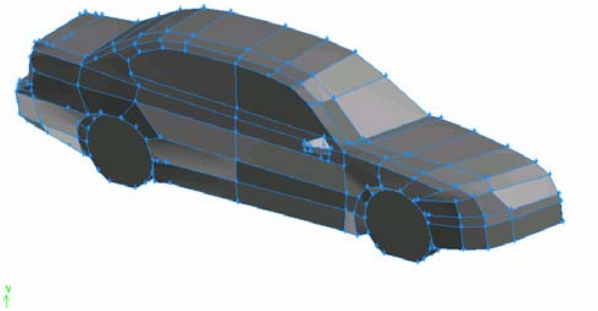

Fig. 13 Car Geometry

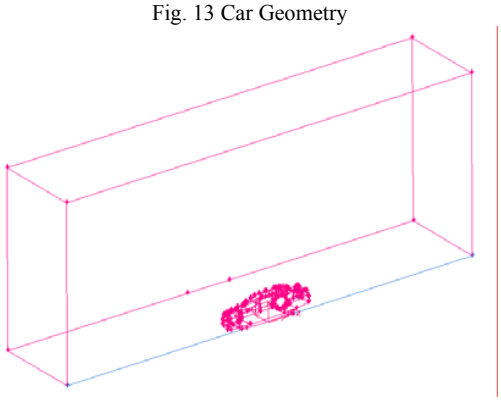

Fig. 14 Meshing Domain

The car will have surface mesh much finer than the domain box. For the domain box boundaries that are touching the car, the grading surface mesh is applied on that surface. This is to provide smooth transition between the mesh on car and the domain box. Otherwise very skewed meshes are generated around the edge segment of the car. Fig. 15 shows the surface mesh of the car and the mesh of the symmetry boundary face.

The triangle on the surface and around the car is much smaller than maximum triangle in the domain (on the surface of the domain box). This provides an opportunity to test if the auto growing function employed is good enough in real engineering application.

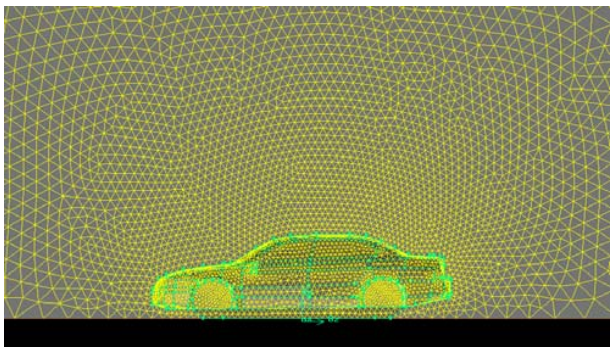

Fig. 15 Surface mesh of the car

The present mesh generator generated 84668 elements compare to the commercial mesh generator 73209 elements. It is believed that the big different in element number is due to different in growth rate. In the commercial mesh generator, only the default settings were without employing any specific growth function; hence the volume mesh size is only affected by the input surface mesh. Mesh from the commercial mesh generator has only short transition distance before it grows to large, uniform mesh. Both result from the present and the commercial mesh generators are shown in Figures 16 and 17 respectively.

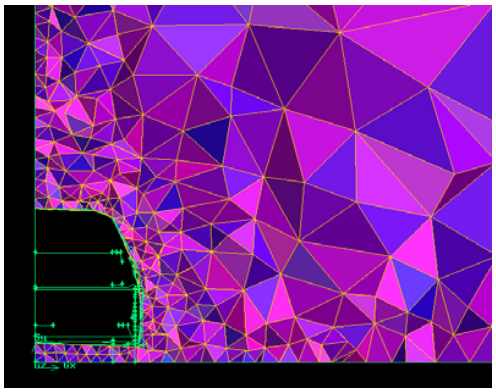

Fig. 16 Cutting view of car mesh generated by the present mesh generator

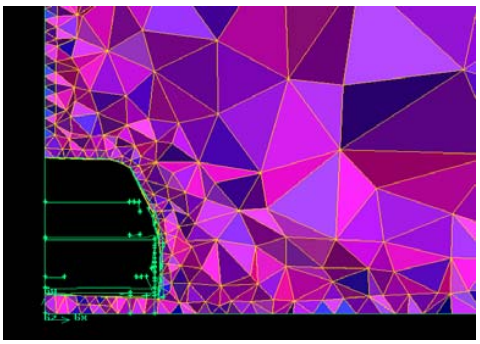

Fig. 17. Cutting view of car mesh generated by the commercial mesh generator

In term of quality, the present mesh generator is superior than the commercial mesh generator in term of angular skewness measurement. It only generated 201 elements (0.24%) with skewness 0.6 and above; while the commercial mesh generator has 2492 elements (4.02%) over 0.6 skewness. The worst element generated is 0.76 (16.93˚ ,153.73˚), compared to commercial mesh generator 0.80 (14.1˚ ,158.11˚). Figures 18 and 19 are comparison of bad elements between the present and the commercial mesh generators respectively.

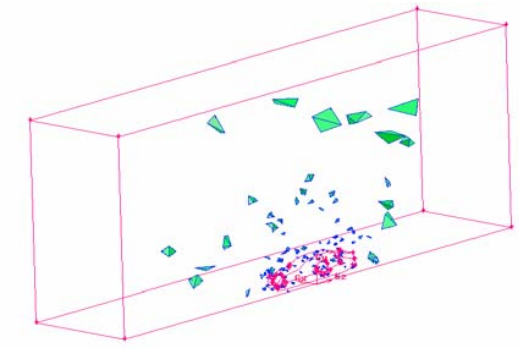

Fig. 18 Bad elements from the present mesh generator

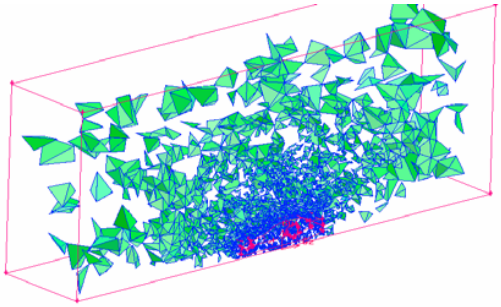

Fig. 19. Bad elements from the commercial mesh generator

Both meshes generated did not contain any sliver. But this is partly because the input surface mesh did not contain small angle and the surface is smooth and in good quality. Reports shows that most of the sliver survived is at the boundary, especially near small angle [3,9,10].

VI. CONCLUSION

A complete tetrahedral mesh generator based on the Delaunay triangulation is presented. The quality of the mesh generated is acceptable and comparable to commercial mesh generator. It was found that the iterative swap-smooth approach, although simple, is very effective in removing bad quality tetrahedral.

The disadvantage of the present scheme is that the number of tetrahedral generated is much larger when come to grading, due to the use of strict growth rules. Furthermore, in the present mesher, theoretically sliver can still survive. But as in other Delaunay refinement scheme, in practice sliver can often be removed during mesh improvement stage.

The possible future work is the development of the automatic surface mesher. Currently, it is assumed that the surface mesh provided is already closed, which may not always be true. Combination of surface mesher and current volume mesher can further increase the robustness of boundary and domain recovery scheme since the surface patch is now known, which provide the possibility of improving surface mesh to suite the need of the volume mesh.

ACKNOWLEDGMENT

The work described in this paper is funded by Ministry of Science, Technology and Innovation (MOSTI), Malaysia under ScienceFund Grant No. 01-02-03-SF0130. The financial support from MOSTI which enable the author to present the work is greatly acknowledged.

REFERENCES

[1] A. Bowyer, *Computing Dirichlet tessellations*, The Computer Journal, 24(2):162–166, 1981.
[2] D. F. Watson, *Computing the n-dimensional tessellation with application to Voronoi polytopes*, The Computer Journal, 24(2):167–172, 1981.
[3] J.R. Shewchuk, *Tetrahedral Mesh Generation by Delaunay Refinement*, Proceedings of the Fourteenth Annual Symposium on Computational Geometry, 86-95, 1998.
[4] G. Timothy, *Block-Structured Applications. Handbook of Grid Generation* . Thompson, Joe F.; Bharat K.Soni, Weatherill , Nigel P..Editors. CRC Press, 13-1, 1999.
[5] H. Borouchaki and S.H. Lo, *Fast Delaunay Triangulation in Three Dimensions*, Comput. Methods Appl. Mech. Engrg. 128: 153-167, 1995.
[6] R. Maehara, *The Jordan curve theorem via the Brouwer fixed point theorem*, American Mathematical Monthly 91, no. 10, pp. 641–643, 1984.
[7] C.J. Ogayar, R.J. Segura, F.R. Feito, F.R. *Point in solid strategies*, Computer & Graphics 29, 616-624, 2005.
[8] J.R. Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. In First Workshop on Applied Computational Geometry. ACM, 124-133, 1996.
[9] B.M. Klingner, J.R. Shewchuk, *Aggressive Tetrahedral Mesh Improvement*, Proceedings of the Sixteenth International Meshing Roundtable. 3-23, 2007.
[10] S.W. Cheng, T.K. Dey, H. Edelsbrunner, M.A. Facello, S.H. Teng, Sliver Exudation, J.ACM, 47, 883-904, 2000.
[11] A. Pierre, S.D. Cohen, Y. Mariette, D. Mathieu, *Variational Tetrahedral Meshing*, Special issue on Proceedings of SIGGRAPH, ACM Transactions on Graphics 24: 617-625, 2005.
[12] J.R. Shewchuk, *Theoretically Guaranteed Delaunay Mesh Generation – In practice*, Short course, Fourteenth International Meshing Roundtable, San Diego, 2005.

Ng Y. Luon is currently a research engineer at the OYL R&D Sdn. Bhd., Malaysia. He is also currently a graduate student at the Centre for Advanced Computational Engineering (CACE), Universiti Tenaga Nasional, Malaysia.

Mohd. Z. Yusoff is currently the Dean of the College of Engineering, Universiti Tenaga Nasional, Malaysia. He is also the Head of the Centre for Advanced Computational Engineering (CACE), Universiti Tenaga Nasional.

Norshah H. Shuaib is currently the Head of the Department of Mechanical Engineering, College of Engineering, Universiti Tenaga Nasional, Malaysia.