www.cadanda.com
Computer-Aided Design AND APPLICATIONS

Optimal Rotational Symmetry Cell Mesh Construction for FE Analysis by Symmetry-constrained Local Delaunay Refinement

Weijuan Cao[1], Ming Li[2] and Shuming Gao[2]

[1]Zhejiang University, Hangzhou Dianzi University,wjcao@cad.zju.edu.cn
[2]Zhejiang University,mli@zjucadcg.cn
[3]Zhejiang University,smgao@cad.zju.edu.cn

ABSTRACT

In performing finite element (FE) analysis for an engineered design, a symmetric model is usually reduced to a symmetry cell, a minimal part of the model from which the whole model can be restored from its symmetry pattern. The mesh of the symmetry cell is then generated separately after the cell extraction. Exploitation of such symmetry properties helps reduce the computational complexity of downstream tasks of mesh generation and engineering analysis, and saves memory usage. However, due to the separation of the symmetry cell extraction and the mesh generation, the mesh generated following such a procedure is usually not optimal in the sense that their element number is not minimal at certain quality requirements, and thus is not very computationally efficient. In order to resolve this issue, a novel approach is proposed in this paper to construct an optimal symmetry cell mesh (in its mesh element number) for a rotationally symmetric CAD model. The optimality is mainly achieved by simultaneously extracting the symmetry cell and generating its associated mesh form using an approach of symmetry-constrained local Delaunay refinement. In such way, the global symmetry information of the whole model is maximally used for mesh quality, and the mesh is only generated for the symmetry cell, instead of the whole model, and thus is very efficient.

Keywords: optimal rotational symmetry cell, Delaunay refinement, FE analysis.

1. INTRODUCTION

Symmetry widely exists in natural and man-made objects. In the finite element analysis of symmetric models, with the aid of symmetry reduction techniques, the symmetry cell instead of the full model is used to obtain the solution, so as to shorten the computation time and to improve the analysis accuracy [1]. However, a rotationally symmetric model can have multiple symmetry cells, whose mesh sizes (number of elements or nodes) may vary vastly. Figure 1 shows two different manually extracted symmetry cells for a 3-fold rotationally symmetric model, where the difference of their node numbers reaches 36% for the larger one. In order to reduce the degree of freedoms for downstream task of finite element analysis, a symmetry cell mesh with minimal elements is very necessary to achieve maximal computational efficiency.

Currently symmetric models are prepared for FE analysis usually in a two-stage manner. First, a symmetry cell is extracted from the original full model. Second, a mesh is generated for the extracted symmetry cell. There are mainly three disadvantages with such methods. First, optimal (in the sense that will be explained later soon) symmetry cells can be hardly generated, because it is not easy to explicitly describe the relationship between a shape and its mesh size (element number) in a reasonable and quantitative way. Second, the introduction of the fixed symmetry boundaries may lead to more mesh nodes. Third, the extraction of symmetry cells involves tedious Boolean operations which are hard to automate in a robust way for complex models.

To resolve the above mentioned issues, an automated method based on Delaunay refinement is proposed in this paper to simultaneously construct symmetry cells and their meshes for rotationally symmetric models. The constructed 2D symmetry cell meshes are size optimal. Following previous work [2], a symmetry cell mesh of a rotationally symmetric model is size-optimal, or optimal for simplicity, if a size optimal mesh of the global model can be trivially produced by performing the rotation transformation

Taylor & Francis
Taylor & Francis Group

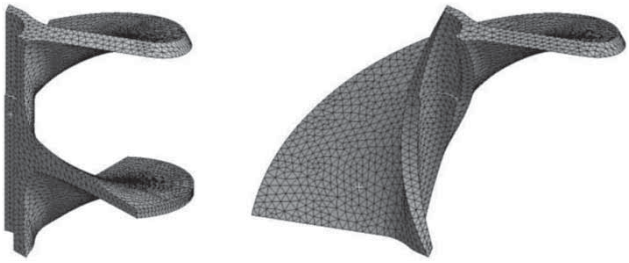

Fig. 1: Different symmetry cells lead to meshes of different node numbers. Example: (a) Symmetry cell mesh with 42839 nodes, (b) Symmetry cell mesh with 27416 nodes.

on it. This is proved under the same framework as that of Delaunay refinement mesh generation. This nice property, however, is not guaranteed in 3D cases although experimental results show that their element numbers are very close to a fraction of the element numbers of the full meshes.

2. RELATED WORK

Suresh considered the problem of automated symmetry cell construction and proposed a method to construct symmetry cells of 2-D solids [4]. In his method, symmetry cells are constructed by cutting the original solid by two carefully selected symmetric lines which originate from the centroid and terminate at the boundary. Heuristics for selecting 'locally optimal' cutting lines were also given. But coarse meshes have to be generated for each candidate symmetry cell in order to ultimately determine the best choice.

As a necessary background, basic steps of the Delaunay refinement algorithms in 2D case are introduced here. Delaunay refinement algorithms for mesh generation operate by maintaining a Delaunay triangulation or a CDT (constrained Delaunay triangulation), which is refined by inserting carefully placed vertices (called Steiner points) until the mesh meets constraints on element quality and size [3]. A Delaunay triangulation for a set P of points in a plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). A CDT can be defined for a PSLG (planar straight line graph) [2], where every input segment appears as an edge of the triangulation.

Two types of Steiner points are inserted [2]. The first type consists of midpoints of encroached subsegments, which are used to recover boundaries of the input model and prevent bad quality triangles at the boundaries. The second type consists of circumcenters of skinny triangles, which are used to eliminate small angles and improve mesh quality. Off-centers [7] can replace circumcenters as another choice of Steiner points.

Incremental insertion algorithms like Lawson's edge flipping algorithm or the Bowyer/Watson algorithm are usually used to update the Delaunay triangulation [3]. In Lawson's algorithm, when a vertex is inserted, the triangle that contains it is found, and three new edges are inserted to attach the new vertex to the vertices of the containing triangle. (If the new vertex falls upon an edge of the triangulation, that edge is deleted, and four new edges are inserted to attach the new vertex to the vertices of the containing quadrilateral.) Next, a recursive procedure tests whether the new vertex lies within the circumcircles of any neighboring triangles; each affirmative test triggers an edge flip that removes a locally non-Delaunay edge. Each edge flip reveals two additional edges that must be tested. When there are no longer any locally non-Delaunay edges opposite the new vertex, the triangulation is globally Delaunay.

Ruppert [2] defines the local feature size function for shapes in Euclid space which relates a geometric shape to vertex density of its triangulation. Using this tool he proves that triangulations generated by his Delaunay refinement method are size optimal, meaning that the number of triangles is within a constant factor of the minimum number possible.

Due to the ignorance of consideration of global symmetric constraints and accumulation of numerical errors, traditional Delaunay refinement algorithms cannot preserve symmetry of the mesh. Zeng proposed an orbit insertion technique to preserve symmetry for symmetric regions [8] by inserting a group of symmetric Steiner points in each refining step.

3. METHOD OVERVIEW

A novel approach is proposed in this paper to simultaneously construct the symmetry cell and its corresponding mesh. In this approach, no additional fixed boundaries are introduced and only one symmetry cell mesh needs to be constructed. The main idea is to directly generate the symmetry cell mesh by symmetrizing and localizing basic operations of the global meshing process. Delaunay refinement algorithms are adopted as the mesh generation method considering its theoretical merits.

Symmetry cell meshes are constructed by a newly introduced symmetry constrained local Delaunay refinement algorithm. The algorithm operates by maintaining a symmetry cell mesh (local mesh for short) of the full model. The full mesh restored from the local mesh by rotational transformations is a Delaunay mesh. The local mesh is iteratively refined by inserting Steiner points until all triangles meet specified quality criteria. Unlike meshing some predefined symmetry cells, symmetry boundaries of the local mesh in the proposed method are not fixed. They are allowed to change during the refining process, leading to evolving symmetry boundaries. The finally resulted local mesh after refinement is taken as the desired symmetry cell mesh.

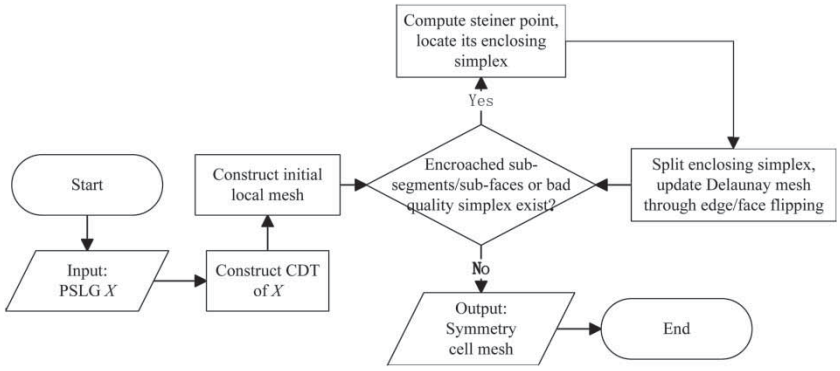

Fig. 2: Symmetry cell mesh construction.

Figure 2 illustrates the process of the symmetry cell mesh construction in 2D. The input to the method is a rotationally symmetric model represented by PSLG. The output is a quality symmetry cell mesh of the input model. The overall process is similar to classical Delaunay refinement algorithms, with the following four major differences: (1) some symmetry cell of the initial CDT instead of the CDT itself is taken as the basis of refinement; (2) symmetry transformations are used to determine the proper Steiner point and its enclosing triangle in the point location step; (3) not only the Steiner point but also its symmetric points are used to find encroached sub-segments; (4) symmetry edges are allowed to be flipped in mesh update.

Key features of the proposed method are presented in section 4 with 2D examples. Issues specific to 3D cases are discussed in section 5. The optimality of the 2D symmetry cell mesh is proved in section 6.

4. SYMMETRY CONSTRAINED LOCAL DELAUNAY REFINEMENT IN 2D

The first key step of the algorithm is the construction of the initial local mesh. In this paper the symmetry cell of the CDT of the input model is taken as the initial local mesh. Denote CDT of the input model X as CDT(X). As X is symmetric, so does CDT(X). Theoretically, any connected symmetry cells of CDT(X) can be taken as the initial local mesh because they all lead to the same result. But, symmetry cells with fewer symmetry edges are preferred as they will reduce the frequency of moving triangles and result in fewer vertices. The initial local mesh is constructed by cutting the CDT of X with two symmetric edge paths, which are found using shortest path algorithms (Fig. 3).

There are three kinds of edges in the local mesh. Symmetry edges are boundary edges that do not belong to X. Fixed edges are boundary edges that belong to X. Interior edges are edges that totally lie within the local mesh. There are two groups of symmetry edges, respectively form the lower symmetry bound and the higher symmetry bound. Symmetry edges on the lower bound can be rotated counterclockwise θ degrees about the centroid of X to coincide with symmetry edges on the higher bound, where θ is the symmetry angle. Reversely, the higher bound can be rotated counterclockwise (360-θ) degrees to meet the lower bound.

If a skinny triangle exists, a Steiner point should be inserted to remove this triangle. Before insertion, the triangle enclosing the Steiner point should be located first. If the point lies outside of the local mesh but one of its symmetric point lies inside of the local mesh, the Steiner point should be replaced with the symmetric point (Fig. 4.). If any sub-segments are encroached by the Steiner point or any of its symmetric points, the Steiner point should not be inserted.

After inserting a Steiner point, triangles enclosing the point are split, and the mesh is updated by edge flipping. Symmetry edges are also allowed to be flipped, simply turn them into interior edges by moving corresponding triangles (Fig. 5.). Suppose e is a symmetry edge, e' is the symmetric edge of e, t is

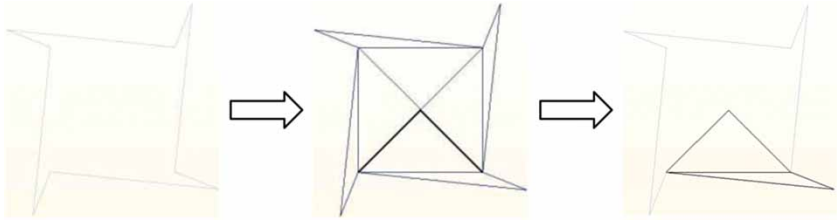

Fig. 3: Construction of initial local mesh (from left to right): (a) input model; (b) CDT; (c) initial local mesh.

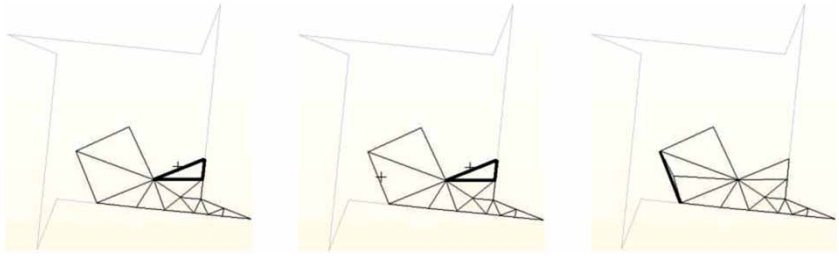

Fig. 4: Steiner point location: (a) Steiner point outside of the local mesh; (b) symmetric point inside of the local mesh; (c) insert symmetric point and split triangle.

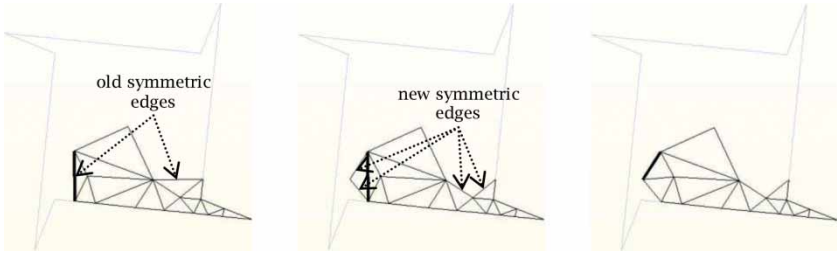

Fig. 5: Flip of symmetry edge: (a) symmetry edge to be flipped; (b) turn the symmetry edge into an interior edge by moving triangle; (c) the edge is flipped.

the adjoining triangle of e, and t' is the adjoining triangle of e'. Compute the rotational transformation T around the centroid of X, which transforms e' to e. e should be flipped if the circumcircle of t' transformed by T contains the vertex of t opposed to e. If e should be flipped, e and e' are merged into one interior edge by transforming vertices of t' by T and moving t' onto e. Adjacency relations of the moved triangle are also adjusted and new symmetry edges are formed.

The symmetry boundary changes as triangles on symmetry edges are moved. The lower bound and the higher bound always keep consistent. The local mesh after the refinement may have bad shaped symmetry boundary, which is adjusted by moving triangles having more than one symmetry edge.

5. SYMMETRY CONSTRAINED LOCAL DELAUNAY REFINEMENT IN 3D

The algorithm in 3D proceeds basically in the same way as its 2D case, but with more complexity. As it is difficult to find smallest symmetry bounds which split the tetrahedron apart, approaches constructing such symmetry bounds have to be developed. Moreover, moving of tetrahedrons is more likely to cause invalid topologies, which should be handled with care. Details are elaborated below.

For clarity, tetrahedron faces in the local mesh are classified into three categories: fixed faces (boundary faces that belong to the input model), symmetry faces (boundary faces that do not belong to the input model) and interior faces. Fixed faces cannot be flipped, while other faces can be flipped. Edges/vertices on symmetry faces are symmetry edges/vertices.

5.1. Construct Initial Symmetry Cell Mesh

The key to the construction of the initial local mesh is to find a pair of symmetry bounds with as fewer faces as possible. The procedure FindSymmetryBounds is used to find such symmetry bounds.

FindSymmetryBounds(X, M)
  //X is the input model, M is the CDT of X
1. Let O be the centorid of X, A be the rotational axis of X, and L be one of the perpendicular axis of A
2. Create cylindrical coordinate system (O,A,L)
   //O is the origin, A is the polar axis, L is the longitudinal axis
3. Compute the cylindrical coordinates (ρ, φ, z) for each point of M
4. Create half-planes P1($\varphi = 0$) and P2($\varphi = \theta$)
5. Find groups of tetrahedrons intersected with $P1$ and $P2$, denoted $G1$ and $G2$
6. Mark adjacent faces between tetrahedrons in $G1$ and tetrahedrons lie totally on the forward side of $P1$ as the lower symmetry bound
7. Mark adjacent faces between tetrahedrons in G2 and tetrahedrons lie totally on the forward side of $P2$ as the higher symmetry bound

---

Each symmetry face on the lower bound is related to its symmetric face on the higher bound. Two faces are symmetric if their vertices are symmetric. Two vertices are symmetric if they have the same radial coordinates and height, and the difference of their angular coordinates equals the symmetry angle. Tetrahedrons between the lower and higher symmetry boundaries form the initial local mesh. Fig. 6 shows the CDT of a rotationally symmetric model and its initial local mesh. The lower symmetry bound are painted in red and the higher symmetry bound are painted in blue.

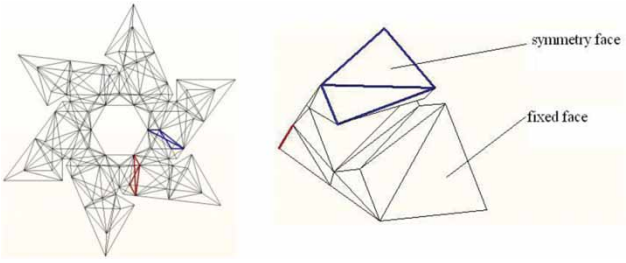

Fig. 6: Construction of 3D initial local mesh: (a) CDT in front view with lower bound in red and higher bound in blue, (b) initial local mesh.

5.2. Move Tetrahedrons

Sometimes, the mesh becomes topologically invalid after a tetrahedron is moved: one exterior edge connected with more than two exterior faces (Fig. 7(a)); one exterior vertex connected with more than one exterior face caps (Fig. 7(b)); even disconnected tetrahedrons. Normally, faces are connected by edges, and tetrahedrons are connected by faces. Faces connected only by vertices and tetrahedrons connected only by vertices or edges are all invalid topologies. Invalid meshes will hinder following up refining operations. These potential invalid topologies must be identified and repaired. The procedure FindMoveTets is used to find and fix potential invalid move.

Information about faces around boundary edges and vertices are used to identify invalid topologies in this paper. Faces that separate tetrahedrons to be moved and the remaining tetrahedrons are called split faces. Denote the group of tetrahedrons to be moved as G. Denote e a boundary edge and v a boundary vertex in G. Six types of invalid topologies are identified according to the following rules. (1) If there are two split faces adjacent to e, then e will become invalid after move (Fig. 8(a)). (2) If e is a symmetry edge, but none of its adjacent symmetry faces belongs to G, then the symmetric edge of e will become invalid after move. (3) If split faces around v form a closed cap, then v will become an invalid vertex after move

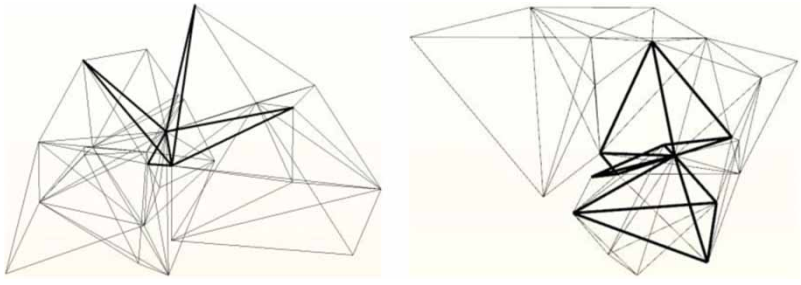

Fig. 7: Invalid topology: (a) invalid edge connection, (b) invalid vertex connection.

```
FindMoveTets(M, G)
  //M is the mesh, G is the group of tetrahedrons to be moved
1. Let b= IsValidMove(G)
2. While b!=VALIDMOVE
3.    If b==INVALIDFACE
4.       FixInvalidFace(M,G)
5.       b=IsValidMove(G)
6.    EndIf
7.    If b==INVALIDEDGE
8.       FixInvalidEdge(M, G)
9.       b=IsValidMove(G)
10.   EndIf
11.   If b==INVALIDVERTEX
12.      FixInvalidVertex(M,G)
13.      b=IsValidMove(G)
14.   EndIf
15. EndWhile
```

(Fig. 8(b)). (4) If spit faces around v fall into at least two groups of edge-connected face patches, then v will become invalid after move. (5) If v is a symmetry vertex, but none of its adjacent symmetry faces belongs to G, then the symmetric vertex of v will become invalid after move. (6) If split faces of G fall into at least two disconnected patches, then G will be split into several disconnected parts.

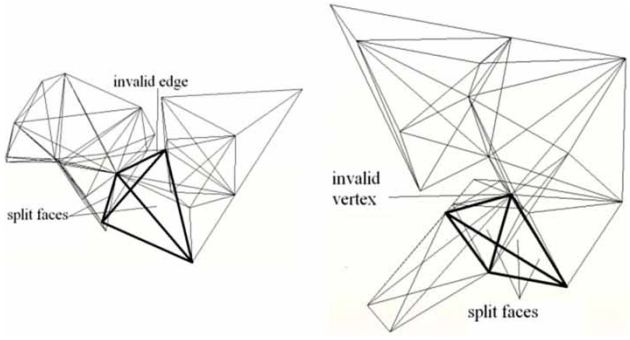

Fig. 8: Configurations leading to invalid topology: (a) two split faces around a boundary edge, (b) split faces form a cap at a boundary vertex.

Potential invalid topologies are repaired by expanding the tetrahedron group. For case (1), G is expanded by including minimal adjacent tetrahedrons of e from one split face to one boundary face. For case (2), G is expanded by including minimal adjacent tetrahedrons of e from one split face to one symmetry face. For case (3), G is expanded by including minimal adjacent tetrahedrons of v from one split face to one boundary face. For case (4), remaining tetrahedrons at v are separated into several disconnected groups. All groups except the largest one are added into G. For case (5), G is expanded by including minimal adjacent tetrahedrons of v from one split face to one symmetry face. For case (6), the remaining tetrahedrons are separated by the split faces into several disconnected groups. All groups except the one containing the docking faces are included into G. Docking faces are object symmetry faces where tetrahedrons in G will be moved to. There may be multiple choices for the combination of split face and symmetry face or split face and boundary face. In such case, the combination which leads to minimal number of new symmetry faces is chosen.

6. SIZE OPTIMALITY

In this paper, the 2D symmetry cell mesh constructed by the proposed method is proved to be size optimal. A symmetry cell mesh is said to be size optimal if the full mesh restored from the symmetry pattern is size optimal, which means that the number of elements of the full mesh is within a constant factor of the minimum number under fixed quality specification. The proof procedure consists of two major steps. The first step is to prove that the full mesh recovered from the symmetry cell mesh can be generated by the orbit insertion method in [8]. The second step is to prove that the mesh generated by the orbit insertion method is size optimal.

Suppose the rotational symmetry fold of an input PSLG X is n, and the symmetry angle is θ ($\theta = 360^\circ/n$). Denote the centroid of X as o. If o lies inside of X, then it is also included in the PSLG. The symmetry group of X is $C_n = \{R_0, R_1, \ldots, R_{n-1}\}$, where $R_i (i \in \mathrm{N}, 0 \le i \le n)$ is a rotational transformation of $i^*\theta$ degrees about o. $lfs()$ is a function associated with X and defined on the plane of X [2]. For any two points p and q on the plane, $lfs(q) \le lfs(p) + |pq|$.

Construct symmetric mesh T and symmetry cell mesh PT synchronously in the following way: (1) Compute initial full mesh $T_0 = \mathrm{CDT}(X)$ and initial symmetry cell mesh PT_0. (2) Refine both meshes by inserting Steiner points. Denote T_i the full mesh and PT_i the symmetry cell mesh after the ith refinement. (3) Denote p the Steiner point of PT_{i-1} in the ith refinement. Apply transformations in C_n to p to get n symmetric points. Insert these points into T_{i-1}. Update both meshes. (4) If all triangles in PT_i satisfy the required quality criterion, stop refinement. Let $PT = PT_i$ and $T = T_i$.

THEOREM 1 *PT is a symmetry cell of T.*

Proof This theorem is proved by induction. (a) By construction, PT_0 is a symmetry cell of T_0. (b) If PT_i is a symmetry cell of T_i, then PT_{i+1} is a symmetry cell of T_{i+1}. Let $T' = \bigcup_{k=0}^{n} R_k(PT_{i+1})$. Firstly, according to the point insertion rule for T_{i+1}, T' and T_{i+1} have the same set of points. Secondly, transformed PT_{i+1} only intersect at symmetry edges. Thirdly, every edge of T' is locally Delaunay, so T' is Delaunay. Therefore, $T' = T_{i+1}$, PT_{i+1} is a symmetry cell of T_{i+1}. (c) By (a) and (b), the theorem holds. ■

LEMMA 1 *Let p and q be two of the n symmetric Steiner points inserted into some T_i, and $q = R_1(p)$. Denote the radius of the largest vertex free circle at p as r. If $|pq| < r$, then $\theta < 60^\circ$, and $|pq| > 2r\sin\frac{\theta}{2}$.*

Proof In Fig. 9, the two circles in solid lines are vertex free circles of p and q respectively. A vertex free circle is a circle that does not enclose any vertices of T_i. Because T_i is rotationally symmetric, hence the two circles are congruent. As o lies in the outside of both circles, therefore $\beta \ge \theta$. When $|pq| < r$, $|pq| = 2r\sin\frac{\beta}{2} \ge 2r\sin\frac{\theta}{2}$, which gives $|pq| > 2r\sin\frac{\theta}{2}$, and $\theta < 60^\circ$. ■

LEMMA 2 *For fixed constants C_T and C_S, the following statements hold:*

(1) For each vertex p in T_0, the distance to its nearest neighbor vertex is at least $lfs(p)$.

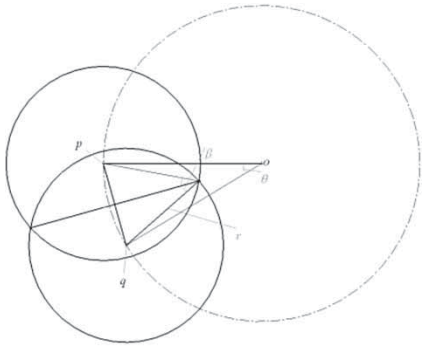

Fig. 9: Two symmetric Steiner points and their vertex free circles.

(2) *When an interior vertex p is added to T_i, the distance to the nearest neighbor vertex is at least* $\frac{lfs(p)}{C_T}$.

(3) *When a boundary vertex p is added to T_i, the distance to the nearest neighbor vertex is at least* $\frac{lfs(p)}{C_S}$.

Proof Case (1) holds by definition of the *lfs*() function. When $\theta \geq 60^\circ$, the lemma can be proved using arguments in [2]. When $\theta < 60^\circ$, the nearest neighbor vertex of p may be its symmetric vertices, in which case C_T and C_S take different values. Denote such a symmetric vertex as q.

(1) p is the circumcenter of a skinny triangle. The radius of the circumcircle is r. According to the derivation in [2], when $C_S \geq C_T \geq 1$, inequality $r \geq \frac{lfs(p)}{(1+2C_S \sin\alpha)}$ holds, where α is the specified minimum angle bound. When $|pq| \geq r$, nearest neighbors of p are vertices of the skinny triangle, at a distance of r. When $|pq| < r$, nearest neighbors of p are its symmetry vertices, at a distance of $|pq|$. By Lemma 1, $|pq| > 2r\sin(\frac{\theta}{2}) \geq \frac{2\sin(\frac{\theta}{2})lfs(p)}{(1+2C_S \sin\alpha)}$. Therefore, case (2) holds when $\boxed{C_T \geq \frac{(1+2C_S \sin\alpha)}{2\sin(\frac{\theta}{2})}}$.

(2) p is the midpoint of an encroached subsegment whose length is $2r$. According to the derivation in [2], when $C_S \geq (1+\sqrt{2}C_T)$, inequality $r \geq \frac{lfs(p)}{(1+\sqrt{2}+2\sqrt{2}C_S \sin\alpha)}$ holds. When $|pq| \geq r$, nearest neighbors of p are vertices of the encroached sub-segment, at a distance of r. When $|pq| < r$, nearest neighbors of p are its symmetry vertices, at a distance of $|pq|$. By Lemma 1, $|pq| > 2r\sin\frac{\theta}{2} \geq \frac{2\sin\frac{\theta}{2}lfs(p)}{(1+\sqrt{2}+2\sqrt{2}C_S \sin\alpha)}$. Therefore, case (3) holds when $\boxed{C_S \geq \frac{(1+\sqrt{2}+2\sqrt{2}C_S \sin\alpha)}{2\sin\frac{\theta}{2}}}$.

When $\sin\alpha < \frac{\sqrt{2}}{2}\sin\frac{\theta}{2}$, the boxed conditions can be simultaneously satisfied by choosing $C_S = \frac{1+\sqrt{2}}{2\sin\frac{\theta}{2}-2\sqrt{2}\sin\alpha}$, $C_T = \frac{1+2C_S \sin\alpha}{2\sin\frac{\theta}{2}}$.

LEMMA 3 *For each vertex p of T, its nearest neighbor vertex q is at a distance at least of* $\frac{lfs(p)}{C_S+1}$.

Proof If p is added after q, or p and q are added at the same time, by Lemma 2, $|pq| \geq \frac{lfs(p)}{C_S}$. If p is added before q, then $|pq| \geq \frac{lfs(q)}{C_S}$, by Lipschitiz condition, $|pq| \geq \frac{lfs(p)-|pq|}{C_S}$, therefore, $|pq| \geq \frac{lfs(p)}{C_S+1}$. ■

THEOREM 2 *T is size optimal.*

Proof According to arguments in [2], if Lemma 3 holds for T, then the size of T is no larger than C times the size of any triangulation of X which has the same radius-edge ratio bound, where $C = O((C_S+1)^2 A)$, where A is the radius-edge ratio bound of T. Thus, T is size optimal. ■

THEOREM 3 *PT is an optimal symmetry cell mesh of X.*

Proof This theorem holds by Theorem1 and Theorem 2. ■

7. RESULTS

The proposed approach is implemented with the open source library Triangle [6] and Tetgen [5], and its performance is tested both on 2D and 3D examples.

Two 2D rotationally symmetric models are tested. For each model, the full mesh and the symmetry cell mesh are generated respectively using traditional Delaunay refinement method and method proposed in this paper. A minimum angle bound of 35 degrees is imposed. Comparative results are shown in Fig. 10 and Tab. 1. It can be seen that the quality of the symmetry cell meshes approaches the quality of the full

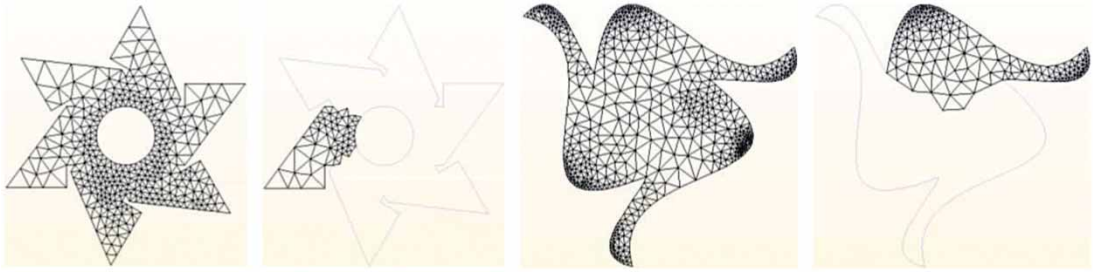

Fig. 10: Symmetry cell meshes for 2D rotationally symmetric models: from left to right, C6 full mesh, C6 cell mesh, C3 full mesh, C3 cell mesh.

Statistics	C6 full mesh	C6 symmetry cell	C3 full mesh	C3 symmetry cell
Mesh triangles	647	90	1095	332
Smallest angle	35.02	35.272	35.005	35.048
Largest angle	109.82	108.49	109.37	106.2

Tab. 1: Mesh statistics for model C6 and C3

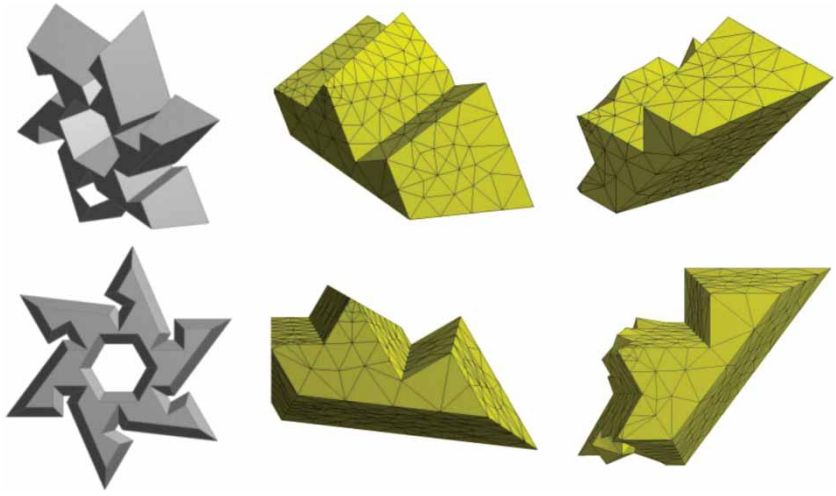

Fig. 11: Symmetry cell meshes for a 3D rotationally symmetric model: from left column to right column, input model, mesh for predefined symmetry cell M1, symmetry cell mesh M2.

meshes, while element numbers of the cell meshes are slightly smaller than 1/n of the element numbers of the full meshes.

Fig. 11 shows two symmetry cell meshes of an 3D rotationally symmetric model. The first column is the input model. The middle column is the mesh generated by Tetgen with periodic boundary condition for a predefined symmetry cell. The rightmost column is the symmetry cell mesh generated using the proposed method. A radius edge ratio of 1.414 and a volume constraint of 10 are imposed. Fig. 12 shows the higher and lower symmetry bounds of our mesh, which are highlighted in red and blue colors respectively. Other information on mesh qualities is listed in Tab. 2, where a tetrahedron's aspect ratio (AR) is its longest edge length divided by its smallest side height.

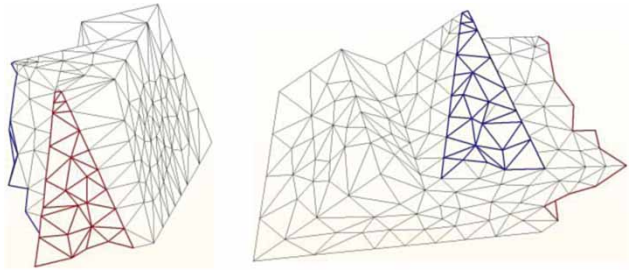

Fig. 12: Symmetry bounds of M2.

Modal analysis is conducted for the two meshes in Fig. 12. Mesh M3 is a mesh generated by the commercial software ANSYS Workbench for the same predefined symmetry cell as M2. M3 has 12461

Mesh	Mesh tetrahedra	AR (<1.5)	AR (1.5-2)	AR (2-2.5)	AR (2.5-3)	AR (3-4)	AR (4-6)	AR (6-10)	AR (10-15)
M1	1162	36 (3.10%)	359 (30.90%)	422 (36.32%)	170 (14.63%)	114 (9.81%)	44 (3.79%)	16 (1.38%)	1 (0.09%)
M2	1115	34 (3.05%)	421 (37.76%)	329 (29.51%)	152 (13.63%)	101 (9.06%)	47 (4.21%)	27 (2.42%)	4 (0.36%)

Tab. 2: Mesh statistics for model in Fig. 12.

Mesh	Time (sec)	Mode 1 Freq.	Mode 2 Freq.	Mode 3 Freq.	Mode 4 Freq.	Mode 5 Freq.	Mode 6 Freq.	Accuracy
M1	2.10	0	6.3077e-6	18.974	25.786	33.175	42.230	81.5%
M2	2.10	0	2.1228e-6	18.822	25.745	32.726	41.575	82.7%
M3	143.25	9.5928e-7	6.0285e-6	16.366	21.673	26.642	36.913	

Tab. 3: Comparison of analysis time and accuracy.

elements. Frequencies of the first six modes are computed. Analysis results are listed in Tab. 3. For brevity, only frequencies of the first harmonic index are given.

Results for 3D models show that the symmetry cell mesh generated by the proposed method has fewer elements than the mesh of the predefined symmetry cell. The analysis takes the same seconds for both meshes. But the accuracy of M2 is slightly higher than M1.

8. CONCLUSIONS

An automated method for constructing symmetry cell meshes of rotationally symmetric models is proposed in this paper. The proposed method has the following advantages: (1) It can lead to as fewer mesh elements as possible without lowering specified mesh quality. (2) Efficiency is achieved by only maintaining necessary minimal meshes. (3) The generated meshes can be taken directly as input to the downstream finite element procedure.

The future work includes: (1) dealing with models with curved boundaries; (2) proof of optimality in 3D; (3) enforcing symmetry constraints for other meshing methods; (4) efficiency comparison with previous automated symmetry cell mesh generation methods.

REFERENCES

[1] Bossavit, A.: Boundary Value Problems with Symmetry and Their Approximation by Finite Elements, SIAM Journal on Applied Mathematics, 53(5), 1993, 1352–1380. http://dx.doi.org/10.1137/0153064

[2] Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation, Journal of Algorithms, 18(3), 1995, 548–585. http://dx.doi.org/10.1006/jagm.1995.1021

[3] Shewchuk, J. R.: Delaunay Refinement Mesh Generation, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

[4] Suresh, K.; Sirpotdar, A.: Automated symmetry exploitation in engineering analysis, Engineering with Computers, 21(4), 2006, 304–311. http://dx.doi.org/10.1007/s00366-006-0021-2

[5] Tetgen, http://wias-berlin.de/software/tetgen/.

[6] Triangle, https://www.cs.cmu.edu/~quake/triangle.html.

[7] Ungor, A.: Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations, Computational Geometry: Theory and Applications, 42(2), 2009, 109–118. http://dx.doi.org/10.1016/j.comgeo.2008.06.002

[8] Zeng, W.; Shi, R.; Gu, X. D.: Global Surface Remeshing Using Symmetric Delaunay Triangulation in Uniformization Spaces, Proceedings of the 2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering, Washington, DC: IEEE Computer Society, 2011, 160–169. http://dx.doi.org/10.1109/ISVD.2011.29