Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

αMST: A Robust Unified Algorithm for Quadrilateral Mesh Adaptation

Chaman Singh Verma[a], Krishnan Suresh[b]

[a]*Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA*
[b]*Department of Mechanical Engineering, University of Wisconsin, Madison, WI 53706, USA*

Abstract

Mesh adaptation plays a critical role in balancing computational efficiency and numerical accuracy. Three types of mesh adaptation techniques exist today, namely, mesh improvement, mesh refinement and mesh simplification, and for each of these, several strategies have been proposed. Most of these strategies yield acceptable geometric mesh quality but provide limited control over topological quality.

In this paper, we introduce a unified algorithm for all three types of mesh adaptation for quadrilateral meshes. The algorithm builds upon the *Minimum Singularity Templates* (MST) idea proposed by the authors for improving the topological quality of a quadrilateral mesh. The MST is extended here to define the concept of an αMST where a single parameter α controls mesh adaptation: $\alpha = 1$ for mesh improvement, $\alpha > 1$ for mesh refinement, and $\alpha < 1$ for mesh simplification.The proposed algorithm generates mesh with high geometric and topological qualities. Further, it is non-hierarchical and stateless, and yet it provides an arbitrary level of mesh adaptation. Finally, since cyclic chords can play an important role in quadrilateral mesh adaptation, we provide a simple constructive algorithm to insert such chords using the αMST. Several examples are presented that demonstrate the robustness, efficiency, and versatility of the proposed concept and algorithm.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: Quadrilateral mesh adaptation; improvement, refinement, simplification; singularities, templates

1. Introduction

Since inception, automatic mesh generating algorithms have been continuously evolving (see two surveys by Bommes [3] and Owen [14]). These algorithms typically accept user's requirements at a high level of abstraction and produce a mesh with high geometric fidelity for simulation. These mesh generators have greatly simplified finite element simulations. A complete automation provided by these methods significantly reduce the most time-consuming phase of simulation, i.e. preparing a model from the underlying geometry. However, many problems facing engineers

* Corresponding author. Tel.: +1-608-698-4729
E-mail address: csverma.wise.edu

Peer-review under responsibility of the organizing committee of IMR 25.

and designers are dynamic in nature. Therefore, to balance computational efficiency and numerical accuracy, even a high-quality mesh must be adapted. For example, in hypersonic flow simulation, the mesh must be refined near shock-waves, while in structural analysis, meshes must similarly be refined and the quality improved near stress raisers; these critical regions are typically not known a priori. A naive and inefficient strategy would be to refine and improve the quality of the mesh everywhere, but this is impractical. It will lead to finite element models with large degrees of freedom, slowing down the simulation. Instead, the meshes must be refined and improved in critical regions, and coarsened elsewhere, a process called *mesh adaptation*. Mesh adaptation ensures a balance between computational efficiency and numerical accuracy.

For mesh adaptation to be effective, it must be fully automated, efficient, and versatile. Several such adaptation strategies have been proposed for both simplicial (triangular and tetrahedral) and non-simplicial (quadrilateral and hexahedral) meshes; the latter being significantly more challenging [1]. The focus of this paper is on quadrilateral mesh adaptation.

Once one or more regions have been identified within a mesh, the overall goal of mesh adaptation is to improve, refine, or coarsen the mesh in these regions, while respecting both geometric and topological quality constraints. Geometric quality constraints include aspect ratio, skew, distortion, shear, etc.; current adaptation strategies are typically capable of respecting such geometric constraints. The topological quality, on the other hand, is determined by the number of nodal *singularities* in the mesh; for a quadrilateral mesh, a mesh node is *regular* if it has four incident edges, otherwise it is *singular* (or irregular) node. Existing mesh adaptation strategies provide limited control over topological quality since it is considered hard to optimize and manipulate topology of a quad mesh, resulting in a large number of singularities. Excessive singularities can,unfortunately,lead to (1) numerical instability in CFD applications [20], (2) wrinkles in subdivision surfaces [11], (3) irrecoverable element inversions near concave boundaries, (4) helical patterns [2], (5) produce visible seams in texture maps, and (6) breakdown of structured patterns on manifolds.

A second limitation of current mesh adaptation strategies is that they are specific to the type of mesh adaptation, i.e., different strategies are needed for mesh improvement, mesh refinement, and mesh simplification, and several such strategies must be combined in practice.

In this paper we describe a unified and robust algorithm for quadrilateral mesh adaptation, with control over both geometric and topological qualities. The algorithm is based on the *Minimum Singularity Templates* (MST) proposed in [22]. While the MST was used to remove singularities in a mesh, it is extended here to define the concept of αMST where a single parameter α controls mesh adaptation: $\alpha = 1$ for mesh improvement, $\alpha > 1$ for mesh refinement, and $\alpha < 1$ for mesh simplification. A second salient feature of the proposed algorithm is that it is non-hierarchical and stateless, making it easy to implement.

2. Basic Definitions and Proposition

In this paper, we use standard meshing terminology. However, for clarity of exposition, we reiterate few of them.

Definition 1. *The valence of a vertex v_i is the number of edges incident on it. A vertex with "n" valence is denoted by $\mathcal{V}n$. An internal vertex with valence 4 is considered* regular*, otherwise it is an* irregular *or a* singular *vertex. An internal vertex with valence 2 is a called* doublet*.*

In this paper, we consider only $\mathcal{V}3$ and $\mathcal{V}5$ singular nodes as all other high valence nodes can be converted into $\mathcal{V}3$ and $\mathcal{V}5$ nodes using standard atomic **face open** *or* **face close** *operation [1].*

Definition 2. *A patch is a sub-mesh with disc topology (Figure 1). Furthermore, we assume that the boundary nodes of the patch are ordered counter-clockwise. We designate some boundary equally spaced nodes $N \in \{3, 4, 5, 6\}$ of the patch as corner nodes and call the patch as N-sided patch. A* side *of the patch is defined as the mesh boundary between two consecutive corner nodes (Figure 2).*

Definition 3. *A Chord in a quadrilateral mesh is a set of quadrilateral elements formed by traversing opposite edges of a quadrilateral starting from an edge. There are two types of chords in a topological valid quadrilateral mesh:*

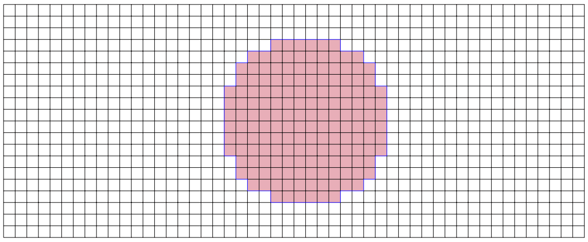
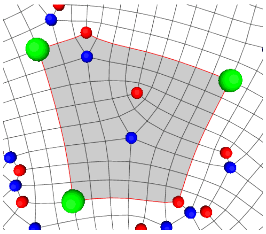
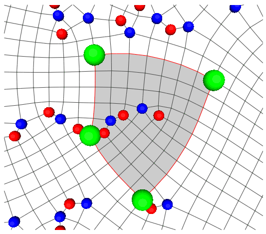


Fig. 1: Shaded quadrilateral elements define a patch.

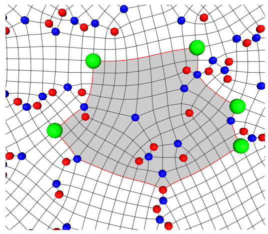
1. **Boundary Chord:** *A chord which contains two boundary edges is called a boundary chord. In fact, in any topological valid quadrilateral mesh, any chord starting from a boundary edge must end at some other boundary edge (Figure 3a).*
2. **Cyclic Chord:** *If starting from an internal edge, traversal completes with the starting edge, then such a chord is called a Cyclic chord (Figure 3b).*



(a) A 3-sided patch.

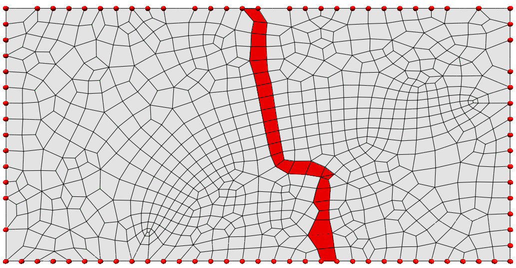


(b) A 4-sided patch.

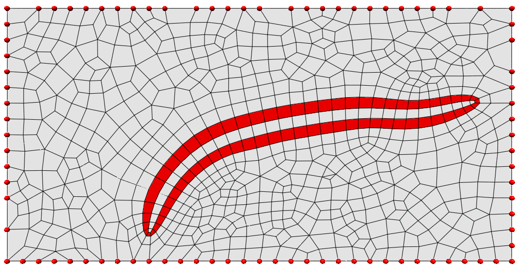


(c) A 5-sided patch.

Fig. 2: Examples of 3-5 and 5 sided patches.



(a) Boundary Chord.



(b) Cyclic Chord.

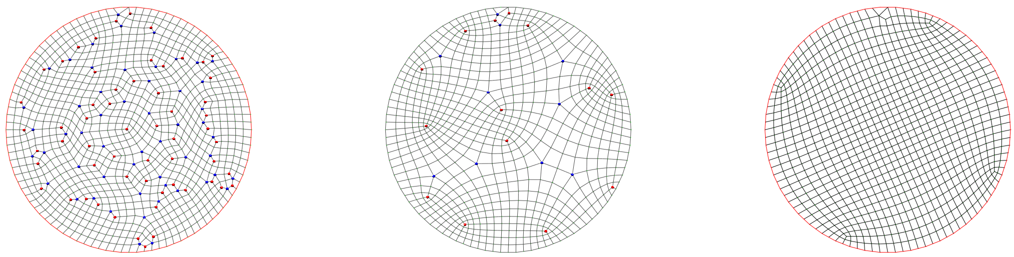
Fig. 3: Chords in a quadrilateral mesh.

3. Related Work

Mesh adaptation has been extensively studied since the beginning of mesh generation algorithms. Here we give a brief overview of the related work which may help understand major contributions of the present work. Since our work unifies all the three adaptation techniques, i.e. improvement, refinement, and simplification, we cover all of them in this brief survey.

- **Mesh improvement:** A high quality mesh is characterized by both geometric and topological qualities. Geometric qualities include element aspect ratio, area, min/max angle, etc. A complete list of various quality metrics is provided in the *Verdict* [15] software and a thorough analysis of various metrics is presented by Shewchuck [19]. In geometric improvement, mesh nodes are repositioned to locations which optimize user-specified objective functions. Since, the literature on mesh geometric optimization is vast, we refer reader to *Mesquite* [6]

For topological quality, we consider the degree of each node and topological improvements involve modifying edges (through swapping, collapsing etc.) so that a mesh achieves better configuration. In [22], we proposed an algorithm based on *Minimum Singularity Template* (MST) to reduce the number of singularities in a quad mesh in localized regions while maintaining geometric quality. Figure4a illustrates an example of a quadrilateral mesh with large number of singularities; after applying the standard MST algorithm, a mesh with significantly less number of singularities is obtained (Figure 4b). Although MST is effective in reducing singularities, both the number and placements of singularities may be suboptimal. A significantly improved patch with our new method is shown in Figure 4c.



(a) Input model (b) MST result (c) Ideal placementl

Fig. 4: With the standard MST, quad mesh improvement is suboptimal.

- **Mesh refinement:** Mesh refinement involves adding new elements in specified regions. As shown in Figure 5 a quadrilateral element can be refined into any number of smaller quadrilateral elements using recursion, but such a subdivision will always have many singularities unless the boundary is also refined. Schneider [18] proposed 2-refinement and 3-refinement templates as shown in Figure 6; an improved version was proposed by Garmella [10]. These templates are applied to the elements identified (grey elements in Figure 6) and tagged for refinement. In order to keep the mesh consistent, neighbouring elements must also be refined; and to avoid refining the entire mesh, singularities are inserted as illustrated. Although these templates provide high geometric qualities, they are hierarchical and produce many singularities in the region adjacent to the selected regions. In addition, application of some of these templates may create unstable refinement [18].

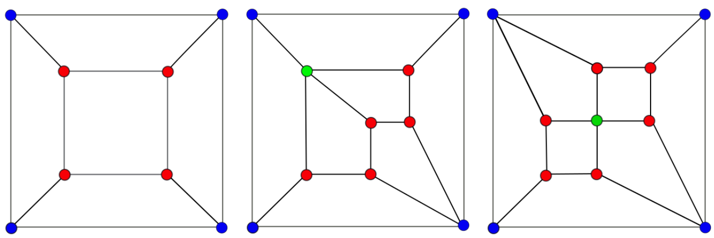


Fig. 5: Local refinement of a quadrilateral element.

- **Mesh simplification:** Simplification involves deletion of elements until a prescribed threshold is achieved. For a quadmesh, simplification is far more challenging than improvement and refinement. For such mesh simplification, many local operations such as quad-close [12], quad-collapse [7], edge split, vertex rotation [21], edge-flips, and quad-vertex merge [8] have been developed. Unfortunately, all these operations increase singularities when applied to a patch containing a single singularity. To be effective, these operations must be applied to large regions [5] or some higher level structures in a quad mesh. *Poly-chord* is one of the structures,

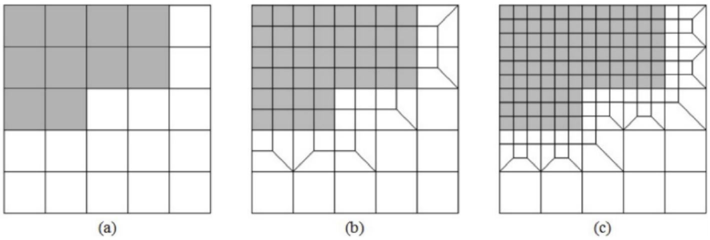


Fig. 6: Schneider's templates can not refine arbitrarily without producing large number of singularities (Image from Schneider's paper).

which has been exploited for quad simplification. In a poly-chord collapse, an entire line of side-to-side quads are removed [4] (see Figure 7). Staten et.al. [9] showed that removal of cyclic chords produces localized coarsening. They also showed a way to create cyclic chords by stitching partial chords using local operation. Dewey et.al [9] later developed coarsening rings (within the coarsening region) and simplified the mesh by collapsing them. Although, it is simple to extract all the poly-chords passing through a region, applying them for the simplifications is usually non-trivial (1) since a poly-chord encapsulates global structure and it can be arbitrarily complex; it can be self-touching, self-intersecting, and can span significant number of elements of the mesh, (2) if there are multitude of chords passing through a region, each one must be incrementally extracted and collapsed, (3) since a chord can extend beyond the localized region, it must be split into smaller independent parts, and (4) collapsing a chord may increase the degree of some nodes, therefore, local face-open operations must be applied after the chord operation.

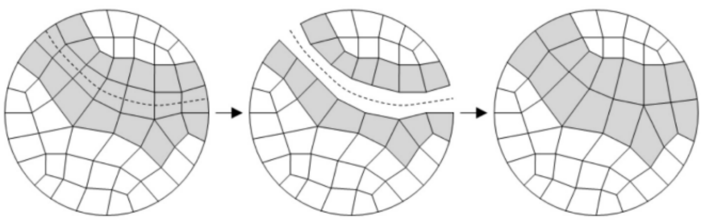
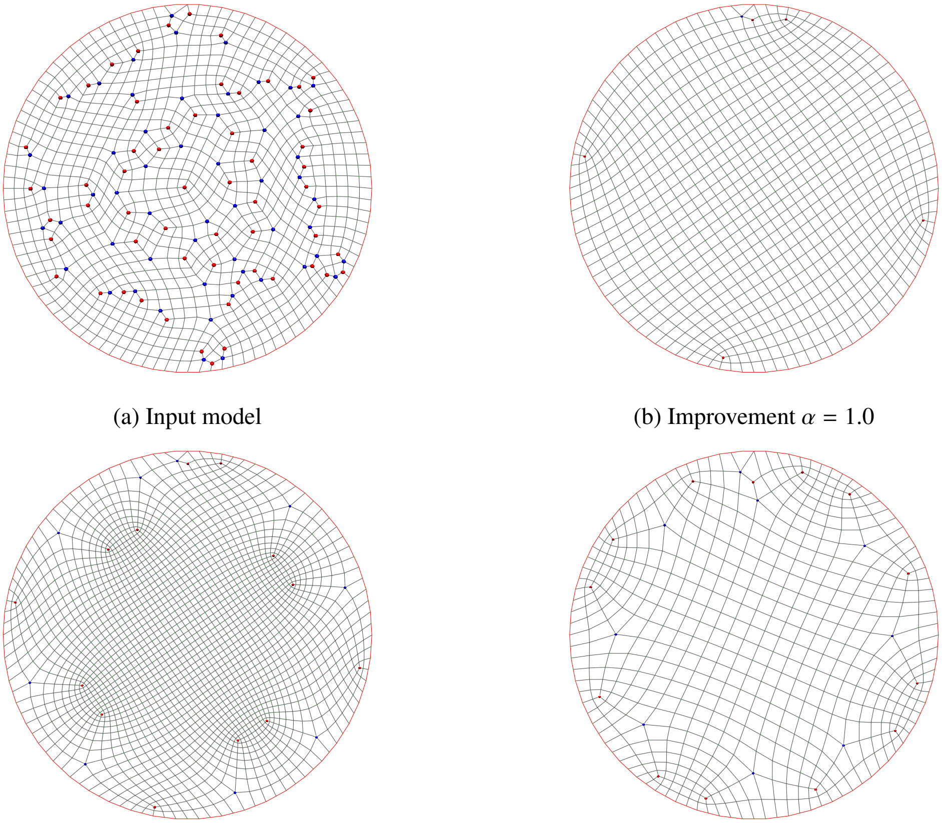


Fig. 7: Quad simplification using chord removal (Image from Anderson's paper).

To the best of our knowledge, only Tarini et.al [21] considered all three adaptation techniques in their work. They defined three kinds of local operations: *coarsening operations,* to simplify the mesh; *optimizing operations*, which change local connectivity without affecting the number of elements; and *cleaning operation*, which resolve invalid configuration. Similarly Kinney [12] provided large number of templates (more than 1000) for quadmesh clean-up.

4. Synopsis and Contributions

In this paper, we are proposing a method to adapt specified regions in a quadrilateral mesh. Typically, these regions are automatically identified and then tagged for refinement, simplification, or improvement. We also assume that these regions are disjoint, almost convex [13] and have disk topology. We refer such a region as a *Patch*. Figure 8a shows one synthetic example of a patch. In addition, we also provide a single parameter α which controls the expected number of quad elements in the region. A value of $\alpha = 2.0$ indicates that the user expects double the number of elements in the patch after adaptation. Our proposed method adapts a patch with few singularities while maintaining high geometric quality. Figure 8 shows all three examples of mesh adaptation. While there are many approaches for each of these tasks, we will show that all these can be done very efficiently with one unified algorithm i.e. αMST.



(a) Input model

(b) Improvement $\alpha = 1.0$

(c) Refinement $\alpha = 1.2$

(d) Simplification $\alpha = 0.80$

Fig. 8: In α MST a single operation can perform improvement, refinement, and simplification.

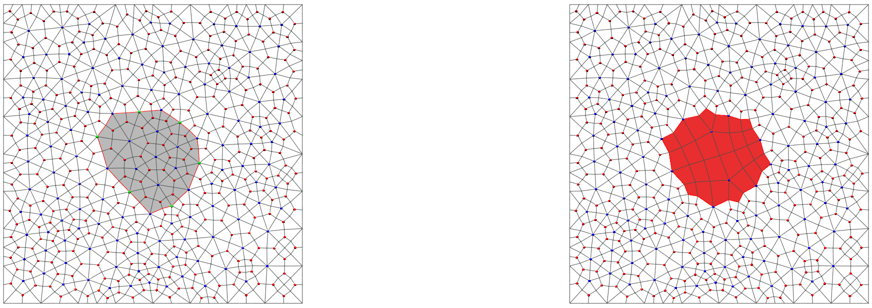
5. Minimum Singularity Templates

In this section, we briefly describe our previous work [22] on the *Minimum Singularity Templates* (MST) and present the idea behind αMST templates.

The MSTs rely on the following four theoretical results from *Combinatorial Topology*:

(I) Every topological disk, with even number of boundary segments admits a quadrilateral mesh.
(II) Every polygon with k-sides has at-least $|k - 4|$ singularities [17].
(III) A single singularity can not be moved, or removed from a quadrilateral mesh [16]
(IV) Minimum number of singularity in domain is decided by the Euler characteristic, and it is invariant with respect to geometric shapes.

Statement II implies that 3-sided and 5-sided patches will always have at-least one interior singularity and only a 4-sided patch can have zero interior singularity. Statement III implies that singularities modifications require at least two singularities. Statement IV places restriction on minimum number of singularities in a model. Based on these results, we presented a constructive algorithm [22] to generate low singularities templates for 3,4,5, and 6 sided polygons. These templates are called *Minimum Singularity Templates (MST)*. We refer readers to [22] for the complete analysis, generation, procedure for applying these templates on any quadrilateral mesh.

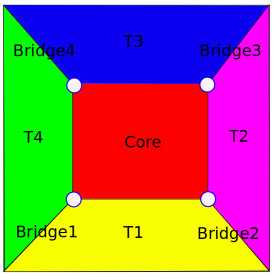


(a) A patch containing singularities. (b) Patch after MST operation and smoothing.

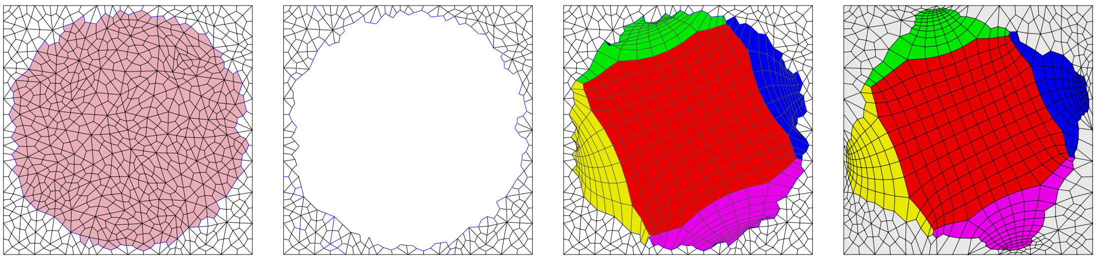
Fig. 9: An example of standard MST operation applied to a patch.

5.1. Genesis of αMST

The standard MSTs are very effective in reducing singularities in a given patch (Figure 9). However, they have limitations: for a given boundary segments on a patch, MST generates a new mesh with fewer number of singularities and quadrilateral elements. Moreover, these mesh templates do not have cyclic chords which we can exploit for mesh adaptation. Therefore, it is very challenging to adapt a MST patch without introducing few more singularities. These new singularities allow adaptation and control the maximum distortion of elements. We extend N-sided MST by subdividing into $N + 1$ sub-patches which will also introduce N singularities within the patch.



(a) A Canonical patch.



(b) A patch in a quad mesh. (c) An empty patch. (d) Remesh the patch. (e) Optimize the patch.

Fig. 10: The idea behind α MST.

Below we present the idea behind αMST with 4-sided patch, but exactly same steps are applicable to other templates.

Figure 10a shows the abstract patch of the 4-sided αMST. Figure 10b shows a patch which we need to adapt. First, we extract its boundary as shown in Figure 10c and arrange the boundary edges in a counter-clockwise direction. Thereafter, we select four nodes uniformly from the boundary. These four nodes correspond to the four corners of the abstract patch. Now we move the four corners inside the domain and create one *core* and four 4-sided transitional patches surrounding the core. In addition, there are also four bridges which connect the core with the boundary of the patch. In this manner, a patch is subdivided into five sub-patches. Improvement, refinement, and simplification are controlled by how the core is discretized. Each sub-patch is remeshed with the standard MST, therefore, we have certain lower bound on the number of singularities in the patch. Figure 10d shows the results of mapping the template in the physical domain. To improve the quality of elements near the patch boundary, we apply *Mesquite* optimization and the result is shown in Figure 10e. Any modification in the core influences elements in the core and the transitional patches and it has no effect outside the patch. Figure 11 shows edge flows in both standard and αMST templates. An edge flow diagram indicates how the boundaries are split for generating quad mesh topology.

- **Improvement:** The boundary of the core is given the same number of nodes as the patch boundary (Figure12a).
- **Refinement:** The boundary of the core is discretized with more nodes than the patch boundary (Figure 12b).
- **Simplification:** The boundary of the core is discretized with less nodes than the patch boundary (Figure 12c).

With this method, the core is discretized with expected number of quadrilateral elements and transitional sub-patches are discretized with a fewer number of singularities to accommodate all-quads elements. It should be noted that the number of singularities in each transitional patch is close to the theoretical least number,however,the patch itself may have more singularities. In practice, after applying MST on each $N + 1$ sub-patches, we also apply the standard MST over the entire patch and it usually collapses many singularities present in the transitional patches.

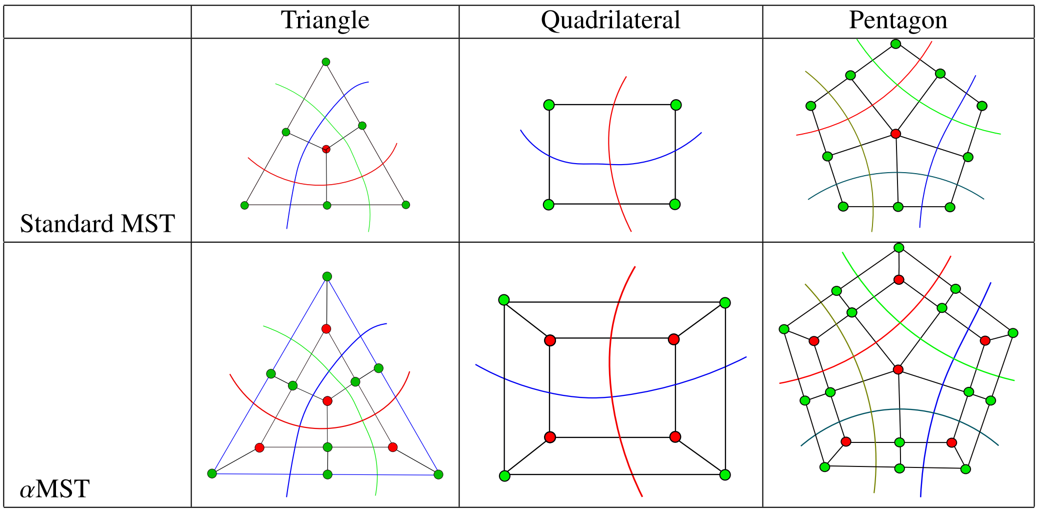
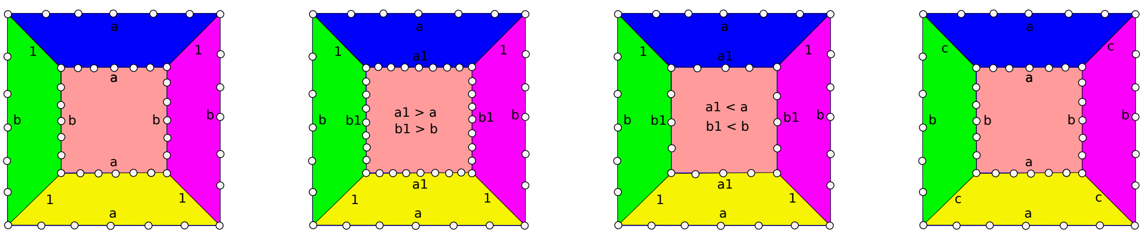


Fig. 11: Edge flows in standard and αMST templates.

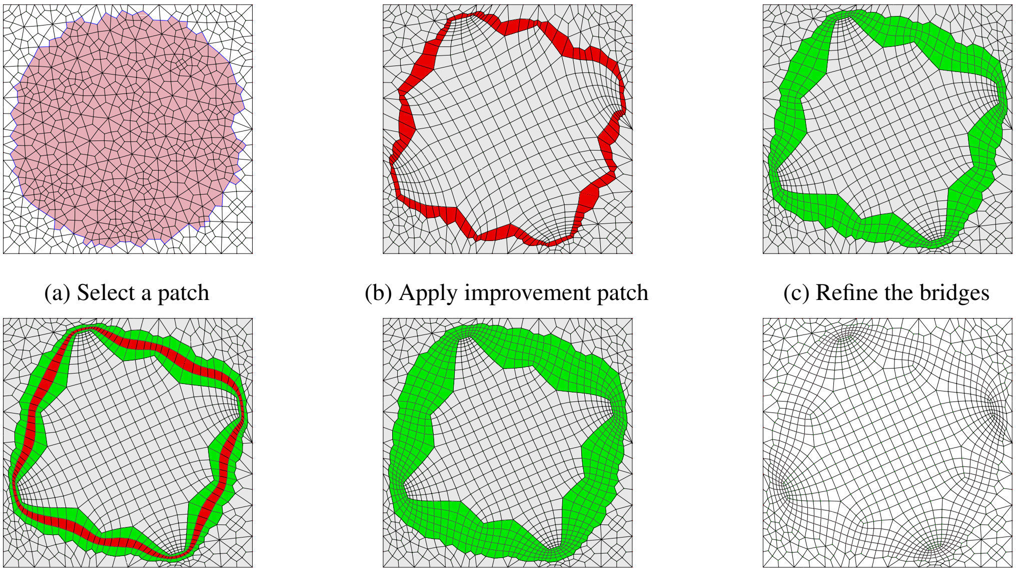


(a) Improvement. (b) Refinement. (c) Simplification. (d) Cyclic chords generation.

Fig. 12: A single template for improving, refining, and simplifying a patch.

5.2. Generating cyclic chord

A cyclic chord provides many advantages in quadmesh adaptation [9]. When these chords are refined, they do not introduce new singularities and if they are simple, removing them is easy. Moreover, removal of a simple cyclic chord which is sandwiched between two simple cyclic chords also does not introduce new singularities. Unfortunately, such cyclic chords are rare in meshes generated with automatic quad mesh generators. With αMST we can generate them easily. Instead of refining the core of a patch, if we keep the number of nodes on the core boundary equal to the patch boundary (Figure 12d) and refine the bridges, then none of the sub-patches will have interior singularities. Such refinement of bridges will create cyclic chords in the transitional patches. Moreover, specifying number of nodes on the bridges is a free parameter, therefore, an unbounded number of cyclic chords can be generated with the αMST. Figure 13 shows the steps in generating cyclic chords. Figure 13f shows the mesh after Mesquite optimization. Such chords are the prime candidate for mesh refinement and simplification.



(a) Select a patch (b) Apply improvement patch (c) Refine the bridges

(d) Refining a cyclic chord (e) Multiple cyclic chords (f) Mesh after optimization

Fig. 13: Creating cyclic chords in a patch.

6. Applying templates

The αMST templates are flexible and can be applied automatically or interactively. During the simulation, the user specifies regions where high density of elements may improve the accuracy. Currently, our algorithm expects the region to be almost convex and have disk topology. The convexity allows using aggressive geometric optimization methods to improve the qualities of interior elements.

From a given patch our algorithm uniformly selects four corners on the boundary. As shown in Figure 14 these four corners influence the position of singularities. Optimal selection of these four corners is a complex problem and requires some global optimization methods. For simplicity and without loss of generality, we just pick four nodes uniformly and apply αMST.

If the objective is to improve, refine, or simplify the entire mesh, then Figure 15b shows one way in which large circular patches are automatically selected using the medial axis of the domain. However, patches can also be created using Voronoi, or convex mesh decomposition algorithms.

After applying αMST to a patch, there may be many singularities on the periphery of the patch. Therefore, after adapting all the patches, we also apply standard MST operations to the final mesh to reduce the singularities and then perform global optimization using Mesquite software.

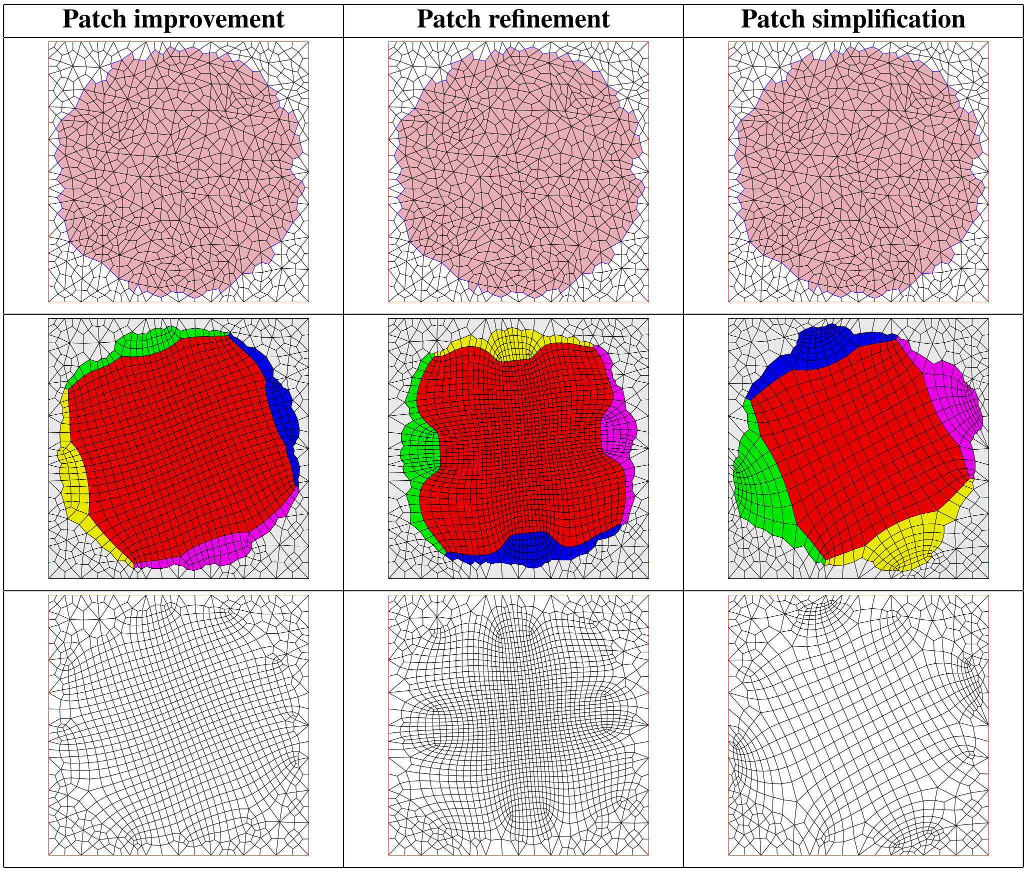


Fig. 14: A simple example of αMST applied to a disk.

Algorithm-I: MeshAdapt($\mathcal{M}$, $listPatch$, $alphaVals$)

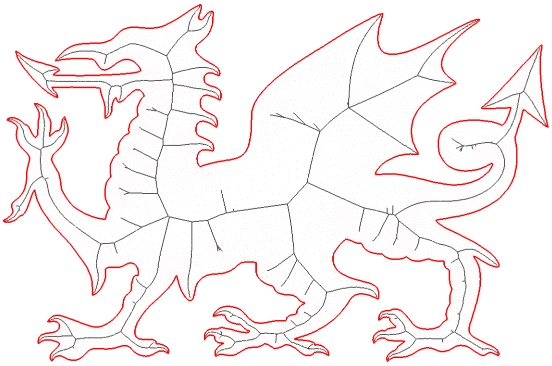
Input: A quad mesh $\mathcal{M}$, list of patches to adapt and corresponding alpha values

Output: The adapted mesh with few singularities and high geometric quality.

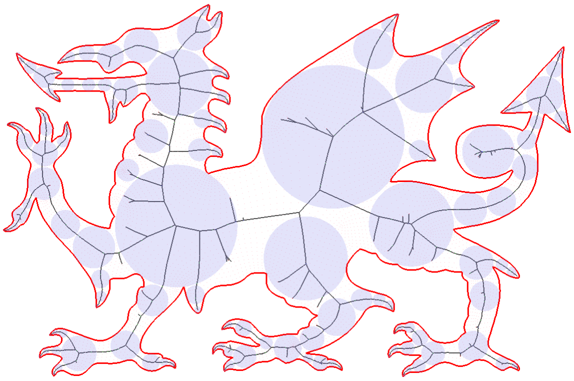
for each patch $\in$ listPatch
    applyAlphaMST(patch, alphaVal[patch.id])
endfor
stdMST($\mathcal{M}$) // To remove singularities on patches periphery
meshOpt($\mathcal{M}$) // Global shape optimization using Mesquite

Algorithm-II: applyAlphaMST($\mathcal{M}_p$, α)

{ boundNodes } = **getBoundary**($\mathcal{M}_p$) // Boundary nodes of the patch in counter-clockwise order
{corners,Nsides} = **getCorners**(boundNodes)
{core, Tpatches} = **createSubPatches**(α, boundNodes, cornerNodes)
$\mathcal{M}_s$ = **stdMST**(core) // Always structured, no singularities
for i = 1, Nsides
    $\mathcal{M}_s = \mathcal{M}_s$ + **stdMST** (TPatches[i])
meshLaplaceSmooth($\mathcal{M}_s$) // Standard Laplacian Smoothing
$\mathcal{M}_p = \mathcal{M}_s$



(a) The medial axis



(b) Medial Circles

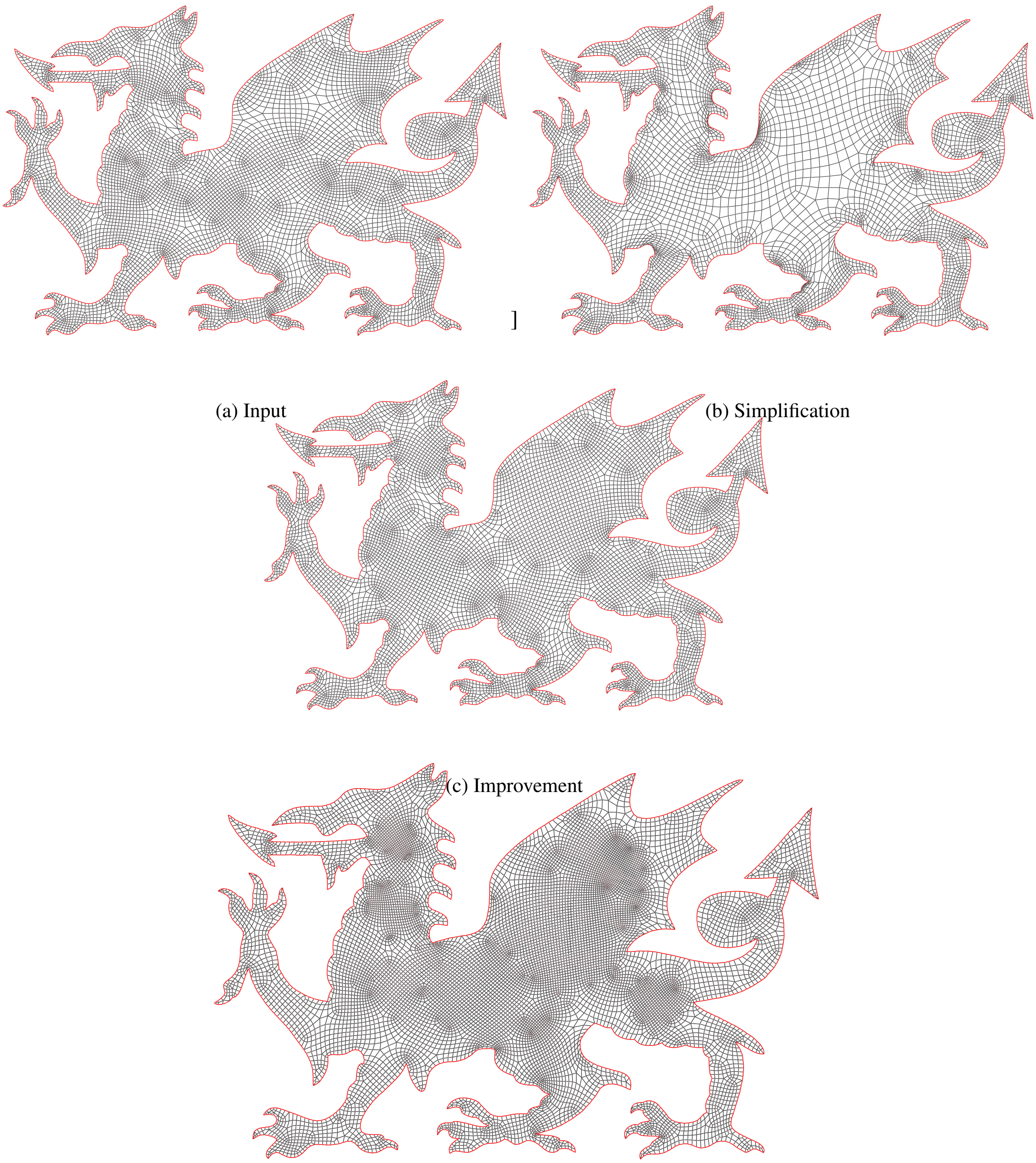
Fig. 15: Medial circles can be used as patches.

7. Results

Because of space limitation, we present only one result with automatic refinement, improvement and simplification. Figure 16a shows the example in which the input mesh has large number of singularities. This input model was discretized with Frontal algorithm of *Gmsh* software. As we can see, mesh improvement significantly reduced the singularities. Table 1 shows that *Verdict* mesh quality after each operation. From the result, it is clear that refinement and simplification do not increase the number of singularities and geometric qualities remain close to the input mesh. Figure 16c shows the refinement with $\alpha = 1.5$. In this example, we refined only the large patches. Figure 16b shows the simplification with $\alpha = 0.80$. In the simplification, we notice that few elements have high distortion. Such distortion can be minimized with better geometric optimization methods. Both skew and taper metric have poor values even for the input mesh and such behaviour was noticed with other examples too.In all these examples, we can notice that the adapted mesh has a smooth transition from the adapted region to non-adapted region and singularities are well-spaced indicating that our method achieves all our goals.

	Welsh			
	Input	**Improve**	**Refine**	**Simplify**
# Faces	7601	7789	12087	5612
# Singularities	572	278	602	400
Aspect Ratio	92.79%	96.20%	97.12%	83.48%
Condition Number	99.00%	99.75%	99.96%	99.56%
Distortion	98.10%	99.10%	99.19%	99.28%
MinAngle	98.80%	99.50%	99.82%	97.82%
MaxAngle	99.20%	99.80%	99.75%	96.00%
Jacobian	99.50%	99.92%	99.99%	99.91%
Scaled Jacobian	100.0%	100.0%	99.97%	99.26%
Shape	98.70%	99.10%	99.96%	99.64%
Shear	97.75%	99.25%	99.98%	99.57%
Skew	21.20%	23.78%	16.25%	41.66%
Taper	9.0%	10.0%	0.0%	0.0%
Warpage	99.90%	100.0%	99.99%	99.91%

Table 1: Mesh Qualities reported by Verdict Software



]

(a) Input

(b) Simplification

(c) Improvement

(d) Refinement

Fig. 16: Results

8. Conclusion and future work

There are different tools for localized quadrilateral mesh improvement, refinement, and simplifications. In this work, we have shown that all these can be done very efficiently with αMST. Unlike some methods, this method is non-hierarchical and stateless, yet it can arbitrarily adapt any number of quadrilateral elements in a patch while maintaining the geometric quality. Moreover, our refinement and simplification are stable as applying these operations do not deteriorate the geometric quality of elements. The simplicity of implementation and generality are two key advantages of our approach. We also present one way to automate the mesh adaptation with a single parameter α. One additional advantage of our approach is that all these operations are easy to unroll if the application of these templates does not match the users expectation. Simplification, in particular, is very attractive, as it does not use the dual-chords concept, which although simple, is not very intuitive from end users perspective. Our method also provides deterministic control over the number of quadrilaterals and singularities. However, our method requires improvement in handling narrow regions. Currently, in very narrow regions few singularities could cause high distortion. Anisotropic refinement and simplification are important in many applications and in near future, we plan to explore similar simplicity using these template.

References

[1] Bret D. Anderson, StevenE. Benzley, and StevenJ. Owen. Automatic all quadrilateral mesh adaption through refinement and coarsening. In *Proceedings of the 18th International Meshing Roundtable*, pages 557–574. Springer Berlin Heidelberg, 2009.

[2] David Bommes, Timm Lempfer, and Leif Kobbelt. Global structure optimization of quadrilateral meshes. In *Computer Graphics Forum*, volume 30, pages 375–384. Wiley Online Library, 2011.

[3] David Bommes, Bruno Lvy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 32(6):51–76, 2013.

[4] Michael J. Borden, Steven E. Benzley, and Jason F. Shepherd. Hexahedral sheet extraction. In *Proceedings of the 11th International Meshing Roundtable, IMR 2002, Ithaca, New York, USA, September 15-18, 2002*, pages 147–152, 2002.

[5] Agostino Bozzo, Daniele Panozzo, Enrico Puppo, Nico Pietroni, and Luigi Rocca. Adaptive quad mesh simplification, 2010.

[6] Michael L. Brewer, Lori Freitag Diachin, Patrick M. Knupp, Thomas Leurent, and Darryl J. Melander. The mesquite mesh quality improvement toolkit. In *IMR*, 2003.

[7] Joel Daniels, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. Quadrilateral mesh simplification. *ACM Trans. Graph.*, 27(5):148:1–148:9, December 2008.

[8] Joel Daniels, II, Claudio T. Silva, and Elaine Cohen. Localized quadrilateral coarsening. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1437–1444, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[9] Mark W. Dewey, Steven E. Benzley, Jason F. Shepherd, and Matthew L. Staten. *Proceedings of the 17th International Meshing Roundtable*, chapter Automated Quadrilateral Coarsening by Ring Collapse, pages 93–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[10] Rao Garimella. *Proceedings of the 18th International Meshing Roundtable*, chapter Conformal Refinement of Unstructured Quadrilateral Meshes, pages 31–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[11] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using catmull-clark surfaces. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 35–44, New York, NY, USA, 1993. ACM.

[12] Paul Kinney. Cleanup: Improving quadrilateral finite element meshes. In *6th International Meshing Roundtable*, pages 449–461, 1997.

[13] Jyh-Ming Lien and Nancy M. Amato. Approximate convex decomposition of polygons. *Computational Geometry*, 35(12):100 – 123, 2006.

[14] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal. Q-morph: an indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 44(9):1317–1340, 1999.

[15] Philippe P. Pebay, David Thompson, Jason Shepherd, Patrick Knupp, Curtis Lisle, Vincent A. Magnotta, and NicoleM. Grosland. New applications of the verdict library for standardized mesh verification pre, post, and end-to-end processing. In MichaelL. Brewer and David Marcum, editors, *Proceedings of the 16th International Meshing Roundtable*, pages 535–552. Springer Berlin Heidelberg, 2008.

[16] Chi-Han Peng, Eugene Zhang, Yoshihiro Kobayashi, and Peter Wonka. Connectivity editing for quadrilateral meshes. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 141:1–141:12, New York, NY, USA, 2011. ACM.

[17] S. Schaefer, J. Warren, and D. Zorin. Lofting curve networks using subdivision surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 103–114, New York, NY, USA, 2004. ACM.

[18] Robert Schneiders. Algorithms for quadrilateral and hexahedral mesh generation. *Proceedings of the VKI Lecture Series on Computational Fluid Dynamic, VKI-LS*, 4, 2000.

[19] Jonathan Richard Shewchuk. What is a good linear element? - interpolation, conditioning, and quality measures. In *In 11th International Meshing Roundtable*, pages 115–126, 2002.

[20] Jos Stam. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, 22(3):724–731, July 2003.

[21] Marco Tarini, Nico Pietroni, Paolo Cignoni, and Daniele Panozzo. Practical quad mesh simplification. *CG Forum (Eurographics*, pages 407–418, 2010.

[22] Chaman Singh Verma and Krishnan Suresh. 24th international meshing roundtable: A robust combinatorial approach to reduce singularities in quadrilateral meshes. *Procedia Engineering*, 124:252 – 264, 2015.