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

# A Quadratic High-order Method for Triangular Mesh Generation Inspired by LBWARP

Mike Stees[a], Suzanne M. Shontz[b]

[a]Deparment of Electrical Engineering and Computer Science, Information and Telecommunication Technology Center, mstees@ku.edu, The University of Kansas, Lawrence, KS, 66045, USA
[b]Department of Electrical Engineering and Computer Science, Bioengineering Graduate Program, Information and Telecommunication Technology Center, shontz@ku.edu, The University of Kansas, Lawrence, KS, 66045, USA

**Abstract**

The ability to construct high-order meshes that conform to the boundary of curved geometries is a limiting factor in the adoption of high-order computational methods for solving partial differential equations. In this work, we propose a method for generating curved meshes of second degree. The approach consists of modifying an initial linear mesh by first, inserting nodes at the midpoint of each line segment, second, displacing the midpoints that fall along the boundary to ensure that they are on the prescribed boundary, and third, solving for the final positions of the internal nodes based on the boundary deformation. We present two numerical examples which demonstrate the viability of our method. We also discuss the quality of the elements generated in the examples.

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

*Keywords:* Mesh generation, high-order, second-order, triangular meshes

## 1. Introduction

In the last few decades, high-order computational methods for solving partial differential equations have piqued the interest of the computational methods community. The appeal of these methods lies in their potential to deliver solutions with a higher degree of accuracy, at a lower computational cost, than their low-order counterparts [1–3]. To realize the full potential of these high-order methods, they must be paired with an accurate discretization of the domain. In the presence of curved domain boundaries, this typically requires a high-order discretization of the domain. In other words, we need to pair these high-order methods with a high-order mesh that accurately captures the domain boundary. As a consequence of this, the adoption of high-order methods has been slowed by the lack of robust high-order mesh generators that are suitable for generating curved meshes for complex geometries.

*E-mail address:* mstees@ku.edu

1877-7058 
Peer-review under responsibility of the organizing committee of IMR 25.

Existing approaches for constructing high-order meshes can be divided into two categories [4]: *direct* methods, which build a curved high-order mesh directly from a suitable representation of the geometry, such as a CAD file, and the more popular *a posteriori* methods, which build a curved mesh by modifying an initial linear mesh [1,5–7]. In a high level sense, *a posteriori* methods consist of three steps: (1) high-order nodes are added to the linear mesh, either by enriching the linear elements uniformly or according to a suitable distribution designed to try and minimize interpolation error [8]; (2) the newly added boundary nodes are projected onto the exact boundary; (3) the interior nodes are moved to their final positions.

The goal of this work is to present a novel *a posteriori* algorithm for second-order mesh generation. The method is inspired by the Log Barrier-Based Mesh Warping (LBWARP) method of Shontz and Vavasis [9]. Different from LBWARP, our method starts with a piecewise linear triangular finite element mesh and adds the midpoints of each triangle to the set of mesh nodes to increase the order of the method. It then warps the resulting second-order mesh to curve the mesh elements.

## 2. The Log Barrier-Based Mesh Warping (LBWARP) Method

The LBWARP method is a mesh warping algorithm originally proposed by Shontz and Vavasis in [9]. At a high level, the algorithm follows these steps: (1) a strictly convex optimization problem is solved to calculate a set of weights for each interior node that represent the relative distances between the node and each of its neighbors; (2) a deformation is applied to the boundary nodes; (3) after moving the boundary nodes to their new positions, the final positions of the interior nodes are calculated by solving a system of linear equations using the optimal set of weights and the new boundary node positions.

## 3. A New Algorithm for Quadratic High-order Mesh Generation

Before explaining our new algorithm, let us define the following notation. First, let $(x_i, y_i)$ represent the x and y coordinates of the $i^{th}$ interior node in the initial mesh. Additionally, let the $x$ and $y$ coordinates of the nodes adjacent vertices be given by $\{(x_j, y_j) : j \in N_i\}$, where $N_i$ is the set of neighbors of node $i$. Unlike [9], our set of neighbors $N_i$ is now defined based on the type of interior node that we are considering. If $i$ is a midpoint of an interior mesh edge, then $N_i$ = {all nodes of the triangles to which $i$ belongs}. If $i$ is a regular interior node, then $N_i$ = {set of all midpoints that are adjacent to $i$}.

Much like [9], we will pose an optimization problem for calculating the set of weights, but first we need to manage an issue that arose as a result of changing our definition of neighbors. In [9], three of the internal nodes' non-adjacent neighbors were chosen to generate local weights to be used as an initial feasible point in the optimization problem. Since we added midpoints to each edge of our triangle and altered the way that we calculate neighbors, we cannot guarantee that the three nodes we chose as neighbors form a convex hull that includes the interior node we want to represent. To solve this problem, we explore a more generalized set of barycentric coordinates called mean value coordinates [10].

Mean value coordinates are a type of generalized barycentric coordinates based on the Mean Value Theorem for harmonic functions. They allow us to solve the following model problem. Suppose we have a set of nodes $v_1, v_2, \ldots, v_k$ that form a star-shaped polygon with $v_0$ as the kernel. The goal of these coordinates is to find a set of weights $\lambda_1, \lambda_2, \ldots, \lambda_k \geq 0$ such that the following constraints hold:

$$\sum_{i=1}^{k} \lambda_{ij} = 1 \tag{1}$$

$$\sum_{i=1}^{k} \lambda_i v_i = v_0. \tag{2}$$

Using mean value coordinates, we write each interior node as a convex combination of its neighbors using the following formulas [10]:

$$\lambda_i = \frac{w_i}{\sum_{j=1}^{k} w_j} \tag{3}$$

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{||v_i - v_0||}. \tag{4}$$

With these preliminaries out of the way, we can pose our optimization problem.

$$\max_{\lambda_{ij}, j \in N_i} \sum_{j \in N_i} \log(\lambda_{ij}) \tag{5}$$

$$\text{subject to } \lambda_{ij} > 0 \tag{6}$$

$$\sum_{j \in N_i} \lambda_{ij} = 1 \tag{7}$$

$$x_i = \sum_{j \in N_i} \lambda_{ij} x_j \tag{8}$$

$$y_i = \sum_{j \in N_i} \lambda_{ij} y_j. \tag{9}$$

As a result of using mean value coordinates to generate our initial feasible point, we must recall that $\lambda_{ij}$ could be 0, which violates (6). Fortunately, given an initial linear mesh of sufficient quality, no $\lambda_{ij}$ will equal 0 and thus $\lambda_{ij} > 0$. Now that we have calculated the initial feasible weights, we can proceed with solving the optimization problem given in (5) – (9) to find a set of weights that maximize the sum of the log of the weights. A closer examination of (5) – (9) reveals that our objective function, together with its constraints, forms a strictly convex optimization problem for which there exists a unique solution. After solving the optimization problem for the weights, the boundary deformation can be applied to move the boundary midpoint nodes to their final locations on the true boundary.

Finally, we use the new positions of the boundary nodes and the weights calculated in (5) – (9) to determine the final positions of the interior nodes by solving the set of linear equations given by (10) – (11)

$$\sum_{j \in N_i} \lambda_{ij} x_j = x_i \tag{10}$$

$$\sum_{j \in N_i} \lambda_{ij} y_j = y_i. \tag{11}$$

## 4. Numerical Results

In this section, we will show two numerical examples which demonstrate the viability of our method. The first example involves meshing an annulus. Figure 1(a,b) show an initial linear mesh and a final second-order curved mesh, respectively. The second example involves meshing the exterior of an airfoil. Figure 2(a,b) show an initial linear mesh

(a) (b)

Fig. 1. An example showing (a) an initial linear mesh of the annulus; (b) a curved second-order mesh

and a final second-order curved mesh, respectively. A closer view of the mesh near the leading edge of the airfoil is shown in 2(c). In order to assess the quality of our resulting meshes, we use the scaled Jacobian quality metric [4]. Figure 3 displays the histograms of the scaled Jacobian for the second-order meshes. In the annulus example, 95.28% of the elements have a scaled Jacobian $\geq 0.9$, and 4.72% have a scaled Jacobian in the range $[0.8, 0.9)$. The minimum value of the scaled Jacobian in this example was 0.85799. In the airfoil exterior example, 92.71% of the elements have a scaled Jacobian $\geq 0.9$, 6.27% have a scaled Jacobian that lies in the range $[0.8, 0.9)$, and the minimum value of the scaled Jacobian is 0.7435.

## 5. Conclusions

We have presented a new *a posteriori* method for generating curved, second-order conformal meshes with prescribed boundaries. The method is inspired by LBWARP. More specifically, the method starts by enriching the initial linear mesh. From there, an optimization problem is solved to calculate a set of weights for each interior node that represent the relative distances between that node and each of its neighbors. Next, a deformation is applied that moves the new boundary points from the first step onto the true boundary. Lastly, the final positions of the interior nodes are calculated by solving a system of linear equations using the optimal set of weights and the new boundary node positions. We demonstrated the performance of our method on two examples, namely an annulus and an airfoil exterior. Finally, we analyzed the quality of the resulting meshes using the scaled Jacobian.

## 6. Acknowledgments

The work of the first author was funded by the Madison and Lila Self Graduate Fellowship. The work of the second author was supported in part by NSF CAREER Award ACI-1500487 (formerly ACI-1330054 and OCI - 1054459).

## References

[1] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Defining quality measures for validation and generation of high-order tetrahedral meshes, in: Proceedings of the 22nd International Meshing Roundtable, Springer, 2014, pp. 109–126.

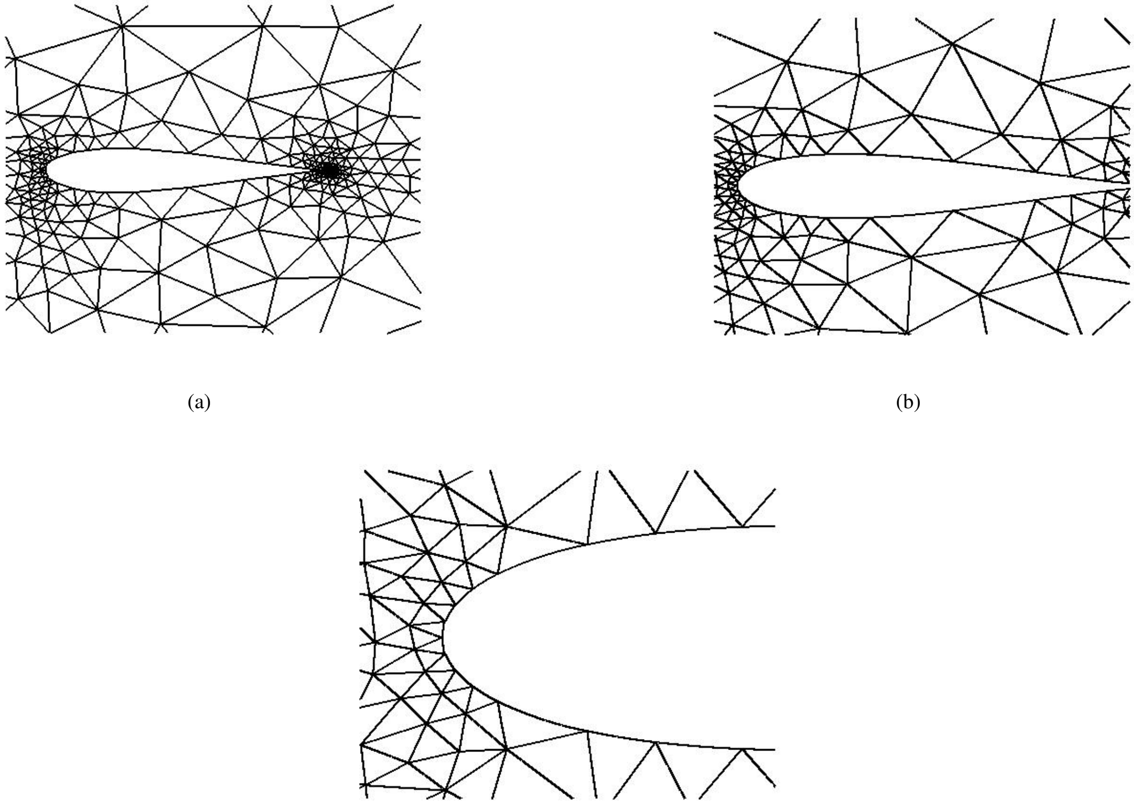
[2] W. F. Mitchell, How high a degree is high enough for high order finite elements?, Procedia Computer Science 51 (2015) 246–255.

[3] P. E. Vos, S. J. Sherwin, R. M. Kirby, From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low-and high-order discretisations, Journal of Computational Physics 229 (2010) 5161–5181.

[4] S. Dey, R. M. O'bara, M. S. Shephard, Curvilinear mesh generation in 3D., in: Proceedings of the 8th International Meshing Roundtable, 1999.

[5] X.-J. Luo, M. S. Shephard, R. M. Obara, R. Nastasia, M. W. Beall, Automatic p-version mesh generation for curved domains, Engineering with Computers 20 (2004) 273–285.

[6] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement using Lagrangian solid mechanics, in: Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, 2009.



(a) (b)

(c)

Fig. 2. An example showing (a) an initial mesh of the airfoil exterior with straight-sided triangular elements; (b) a curved second-order mesh; (c) a close-up view of (b)


(a) (b)

Fig. 3. Histograms of (a) the scaled Jacobian of the annulus example and (b) the scaled Jacobian of the airfoil exterior.

[7] Z. Q. Xie, R. Sevilla, O. Hassan, K. Morgan, The generation of arbitrary order curved meshes for 3D finite element analysis, Computational Mechanics 51 (2013) 361–374.
[8] T. Warburton, An explicit construction of interpolation nodes on the simplex, Journal of engineering mathematics 56 (2006) 247–262.
[9] S. M. Shontz, S. A. Vavasis, A mesh warping algorithm based on weighted Laplacian smoothing., in: Proceedings of the 12th International Meshing Roundtable, 2003, pp. 147–158.
[10] M. S. Floater, Mean value coordinates, Computer Aided Geometric Design 20 (2003) 19–27.