Anisotropic advancing front quadrangulation with optimization-based smoothing

Philip Caplan, Robert Haimes, David L. Darmofal and Steven R. Allmaras

Massachusetts Institute of Technology, Aerospace Computational Design Laboratory

1 Introduction

Recent research in high-order discretisations of partial differential equations suggests the need for mesh adaptivity to reduce solution error at a reasonable computational cost [8]. Simplices are often used because of their ability to easily mesh complex domains; however, hexahedra offer better alignment with solution features, such as shocks or boundary layers. The goal of this report is to describe some work in developing an anisotropic quadrilateral mesh generator with the intent of extending the framework to higher dimensions.

Anisotropy is specified here through a continuous field of metric tensors, following the framework of [6], [3] and [4]. Since we expect our metric to be described discretely at the nodes of a background tesselation, the tensor at any point $\mathbf{x}$ is evaluated by computing the weighted affine-invariant mean,

$$\mathcal{M}(\mathbf{x}) = \underset{\mathcal{M}}{\operatorname{argmin}} \sum_{\nu \in \mathcal{V}(\kappa)} w_\nu(\mathbf{x}) || \log \left(\mathcal{M}_\nu^{-1/2} \mathcal{M} \mathcal{M}_\nu^{-1/2} \right) ||_F^2, \quad \mathbf{x} \in \kappa \tag{1}$$

where κ is the enclosing element of $\mathbf{x}$ in the background tesselation, $\mathcal{V}(\kappa)$ is the set of vertices of κ and $w_\nu(\mathbf{x})$ is the barycentric coordinate of $\mathbf{x}$ corresponding to vertex ν. This mean is computed iteratively with the gradient descent algorithm of [6] which usually converges within 10 iterations.

Our problem can now be stated as follows: generate a quad-dominant mesh such that all edges are nearly of unit length under the specified metric,

$$\sqrt{2}/2 \leq \ell_{\mathcal{M}}(\mathbf{e}) \leq \sqrt{2}, \qquad \text{where } \ell_{\mathcal{M}}(\mathbf{e}) = \int_0^1 \sqrt{\mathbf{e}^T \mathcal{M}(\mathbf{e}_0 + \mathbf{e}s)\mathbf{e}} \mathrm{d}s \tag{2}$$

and all elements are of sufficient quality in the metric space,

$$c_\kappa \leq q(\kappa) \leq \beta, \qquad \text{where } q(\kappa) = \frac{\sum\limits_{e \in \partial\kappa} \ell^2_{\mathcal{M}}(e)}{\int\limits_\kappa \sqrt{\det \mathcal{M}(\mathbf{x})}\, \mathrm{d}\mathbf{x}} \tag{3}$$

where c_κ is the quality of a perfect element: $c_\kappa = 4\sqrt{3}$ for triangles and 4 for quadrilaterals; β provides a bound on the poorest desired element quality.

2 Approach

Our overall approach is similar to the Q-Morph algorithm [5]. That is, we generate an initial triangulation, set up fronts along the boundaries and advance towards the interior using a suitable method for defining quadrilaterals. In contrast to [5] and [2], we do not use edges of the existing triangulation for defining quadrilaterals but, instead, recover all the necessary edges through edge swapping [5]. The primary advantage of marching into an existing triangulation instead of free space as in [1] is that front collisions are detected at a lower cost during edge recoveries or point-in-triangle searches.

2.1 Boundary discretisation

The fronts are initialized to the geometry boundaries and must first be discretised according to the requested metric. Each boundary is represented by a set of primitive entities (lines, arcs, splines). The total length of the boundary entity in the metric space $\ell(\Gamma)$ gives an estimate of the number of edges to be inserted as $N = \text{round}(\ell(\Gamma))$; each edge then has a target length of $\overline{\ell(\gamma)} = \ell(\Gamma)/N$ and are discretised by recursively bisecting the remaining arclength until a root of $f(s) = \ell(\gamma) - \overline{\ell(\gamma)}$ is found (Fig. 1(a)).

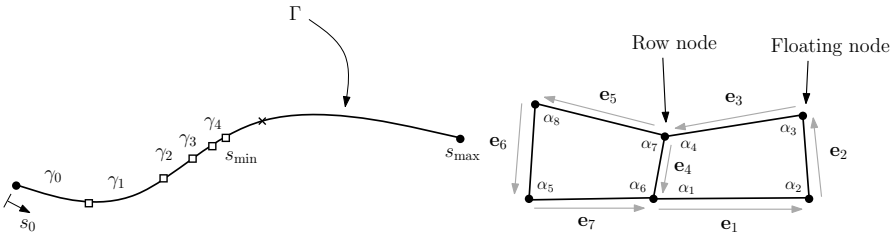
2.2 Quadrilateral definition

We explored a variety of methods for defining quadrilaterals. One interesting method was to locally streamline the eigenvectors of the continuous metric field with a four-stage Runge-Kutta integration method [7]. Elements were defined by intersecting these streamlines to create unit length edges; however, this had no guarantee on the resulting element quality. We also studied objective functions to locally optimize the coordinates of the inserted quad. One of these involved the minimization of the affine-invariant distance between the element-implied ($\mathcal{M}_\kappa$) and target ($\mathcal{M}_t$) metric,

$$\kappa_{\text{opt}} = \underset{\kappa}{\text{argmin}} \int\limits_\kappa || \log \left(\mathcal{M}_t^{-1/2}(\mathbf{x}) \mathcal{M}_\kappa(\mathbf{x}) \mathcal{M}_t(\mathbf{x})^{-1/2} \right) ||^2_F \, \mathrm{d}\mathbf{x} \tag{4}$$

where the element-implied metric is computed from the Jacobian as $\mathcal{M}_\kappa(\mathbf{x}) = (\mathcal{J}(\mathbf{x})^T \mathcal{J}(\mathbf{x}))^{-1}$.

The aforementioned methods are computationally expensive and, although they solve the local quadrilateral generation problem, they ignore the global one. One single insertion has a significant effect on the topology of the growing mesh. Thus, we also developed a more heuristic approach, locally analyzing the requested sizes of the metric at the front edge and spend additional time smoothing the surrounding front edges.



(a) Boundary discretisation (b) Smoothing

Fig. 1. Boundary-metric discretisation and smoothing algorithms

2.3 Front smoothing

Front nodes are smoothed based on whether they are *row nodes* which are connected to two quadrilaterals or *floating nodes* which are connected to only one quadrilateral [1], [5], [2]. In contrast to previous approaches, nodes are smoothed with a nonlinear optimizer. To avoid additional computational cost, only five nodes surrounding an insertion are smoothed. For a floating node ν, we minimize

$$f(\nu) = (\ell(\mathbf{e}_2) - 1)^2 + (\ell(\mathbf{e}_3) - 1)^2 + \cos^2(\alpha_2) + \cos^2(\alpha_3) + \cos^2(\alpha_4) \qquad (5)$$

where the angles are measured in the metric space. For example, $\cos(\alpha_3) = (\mathbf{e}_2 \cdot \mathbf{e}_3)_{\mathcal{M}} / (\ell(\mathbf{e}_2)_{\mathcal{M}} \ell(\mathbf{e}_3)_{\mathcal{M}})$. This function drives edges $\mathbf{e}_2$ and $\mathbf{e}_3$ to unit length under $\mathcal{M}$ and angles α_2, α_3, α_4 to 90° in the metric space (Fig. 1(b)). A similar approach is taken for row nodes, where additional terms accounting for the lengths of edges $\mathbf{e}_5$ and $\mathbf{e}_6$ and the remaining angles are added to Eq.(5).

3 Results and Discussion

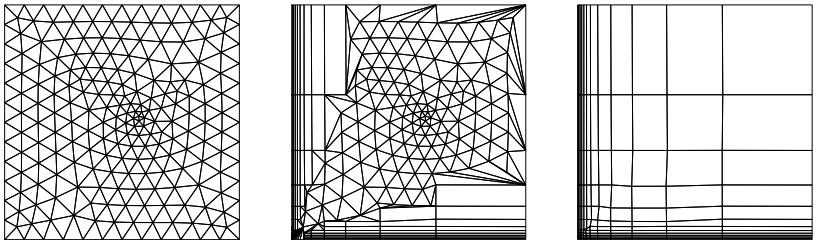
Preliminary tests

Our approach was first tested in a $[-10, 10]^2$ square with an analytic shock-boundary layer type metric,

$$\mathcal{M}(x,y) = \begin{bmatrix} (.045x + 5.05)^{-2} & 0 \\ 0 & (.045y + 5.05)^{-2} \end{bmatrix} \tag{6}$$

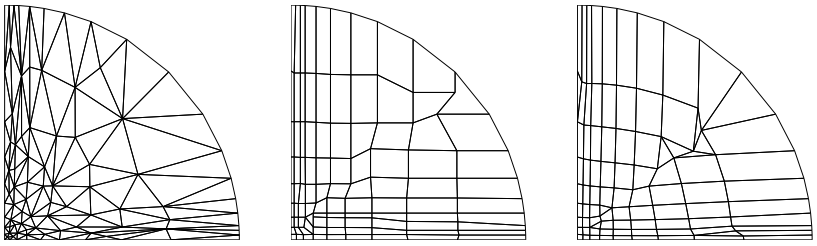
defined at the vertices of the triangulation of Fig. 2(a). The resulting mesh contains 165 vertices, 139 quads and 2 triangles (lower left corner of Fig. 2(c)); the average edge length in $\mathcal{M}$ is $\bar{\ell}_{\mathcal{M}} = 0.934$ with a standard deviation of 0.110.

A similar, less anisotropic metric was then tested in a quarter-circle to study the effects of a boundary which does not align with the eigevenvectors of $\mathcal{M}$. The meshes obtained by streamlining the eigenvectors (Fig. 3(b): $\bar{\ell}_{\mathcal{M}} = 0.939 \pm 0.163$) and the normal-based method (Fig. 3(c): $\bar{\ell}_{\mathcal{M}} = 0.996 \pm 0.150$). It appears streamlining the eigenvectors produced higher quality elements at front collisions; however, boundary alignment and overall mesh quality are poorer than the normal-based method. Note there is still some topological cleanup to be done and we have yet to implement a global anisotropic smoothing scheme. Both methods, nonetheless, exhibit the desired directionality in contrast to simplex adaptation algorithms, as shown in Fig. 3(a).



(a) Initial (b) Intermediate (c) Final

Fig. 2. Meshes generated for the square test case (normal-based insertion)



(a) Simplex-based (b) Streamlining (c) Normal-based

Fig. 3. Comparison of different methods in the quarter-circle test case

RAE 2822 transonic airfoil with boundary layer

The algorithm is currently being tested on a transonic RAE 2822 airfoil where the requested metric is optimized to reduce the drag error estimate at a specified computational cost [9]. Fig. 4(c) suggests our mesh generation algorithm is appropriately capturing boundary alignment while conforming to the metric; however, there is still work to be done in refining the smoothing algorithm since many irregular elements are formed as the fronts advance (Fig. 4(b)).

(a) Initial (b) Intermediate (c) Intermediate (zoom)

Fig. 4. Meshes generated for the RAE 2822 test case (normal-based insertion)

References

1. Blacker, T. D. and Stephenson, M. B. (1991) Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32:811-847.
2. Lee, Y. K. and Lee, C. K. (2003) A new indirect quadrilateral mesh generation scheme with enhanced local mesh smoothing procedures. *International Journal for Numerical Methods in Engineering*, 58:277-300.
3. Loseille, A. and Alauzet, F. (2011) Continuous mesh framework I: Well-posed continuous interpolation error. *Society for Industrial and Applied Methematics*, 49(1):38-60.
4. Loseille, A. and Alauzet, F. (2011) Continuous mesh framework II: Validations and applications. *Society for Industrial and Applied Mathematics*, 49(1):61-86.
5. Owen, S. J., et al. (1999) Q-Morph: An indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 44:1317-1340.
6. Pennec, X., Fillard, P. and Ayache, N. (2006) A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision*, 66(1):41-66.
7. Vyas, V. and Shimada, K. (2009) Tensor-guided hex-dominant mesh generation with targeted all-hex regions. In *Proc. of 18th International Meshing Roundtable*, 377-396.
8. Yano, M., Modisette, J. M., and Darmofal, D. L. (2011) The importance of mesh adaptation for higher-order discretizations of aerodynamic flows. In: AIAA-2011-2852.
9. Yano, M. (2012) An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes. *PhD Thesis*, Massachusetts Institute of Technology.