A Prismatic-Layer Advancing-Front Approach to Anisotropic Metric-Based Curved Mesh Generation*

Krzysztof J. Fidkowski†
University of Michigan, Ann Arbor, MI, 48188

We present a strategy for generating curved, high-order, hybrid meshes that consist of a combination of prismatic layers close to the body and unstructured elements in the rest of the domain. Such curved meshes are required for high-order discretizations, but they are difficult to generate and adapt, in particular when anisotropic elements are desired next to curved geometries. To address the problem of possible inversions in this region, the proposed strategy grows prismatic, anisotropic layers of elements close to the geometry, using a metric to dictate the element sizing and starting with a metric-conforming surface mesh. The curvature of the elements attenuates as the layers grow, and the prismatic regions end once linear faces are possible. A linear unstructured mesh fills the remaining portion of the domain, and it is generated using a simple, metric-based advancing-front algorithm. The strategy is implemented in an adaptive solution process by using an existing mesh as a scaffold for metric evaluation. Results are presented for two-dimensional problems in an output-based adaptive setting. Comparisons with global remeshing show similar performance in output convergence, mesh quality, and aspect ratio distributions, and improved robustness and efficiency due to lack of a separate mesh curving step.

I. Introduction

Current techniques used in generating and adapting curved meshes for high-order discretizations lack robustness when the elements become highly anisotropic, particularly in three dimensions. This is due to the possibility of obtaining negative mapping Jacobians, i.e. inverted elements or parts of elements, upon curving the elements to conform to the domain boundary. It is near the boundary that the majority of highly-anisotropic elements reside for computational fluid dynamics problems of engineering interest, which generally involve solutions of the Reynolds-averaged Navier-Stokes equations (RANS). Hence, the problem is of practical interest when simulating flow over aerodynamic bodies at high Reynolds number, using high-order methods.

One approach to curved-mesh generation and adaptation is hanging-node refinement of initially structured quadrilateral or hexahedral meshes [1, 2], and this has been the approach pursued in our work so far [3–5]. This is

*A preliminary version of this work appeared as paper 2024-1019 at the AIAA SciTech 2024 conference in Orlando, FL.
†Professor, Department of Aerospace Engineering, AIAA Associate Fellow.

a viable approach because curved meshes can be generated from linear ones through element agglomeration, and the subsequent refinement can be done in reference space of the quadrilaterals or hexahedra. However, it relies on the availability of structured meshes, the generation of which is still user-intensive for complex geometries, e.g. in creating a domain blocking. Such meshes can also be inefficient when anisotropic elements propagate to the farfield for point-matched blocks.

At the opposite end of the spectrum is fully-unstructured mesh generation and adaptation, with robust support for anisotropic curved elements. This area of research has seen much progress over the last 10-15 years [6–12], with various approaches ranging from linear elasticity to local mesh modification operations. Despite the progress, curved mesh generation remains a challenging task in three dimensions [13], particularly for cases that demand anisotropic elements near the surface.

We presently introduce an approach that bridges the gap between structured meshes and fully-unstructured mesh adaptation. This approach consists of two stages: the generation of prismatic layers (PL) close to the surface, followed by an advancing front (AF) simplex mesh generation for the rest of the domain. Local operations are also performed on the simplex mesh. Both stages are metric-based, meaning that they generate elements that conform to a prescribed Riemannian metric field. The implementation of the prismatic-layer advancing-front (PLAF) mesh generator is fairly straightforward, in that the first stage involves only growth of a surface mesh along outward-pointing normals, whereas the second stage deals only with linear elements.

In this paper, we outline PLAF and present results for two-dimensional problems. Section II overviews the approach, the equations, and the discretization. Section III reviews output-based error estimation and metric optimization, with new aspects relevant to hybrid meshes. The prismatic-layer advancing-front mesh generation method is presented in Section IV. Section V shows results for the proposed mesh generation approach used in an adaptive setting, with comparisons to an existing metric-based remeshing method, for several aerodynamic test cases. Section VI concludes with a summary and future directions.

II. Approach

A. Overview

Figure 1 illustrates the mesh generation and adaptation approach. The hybrid mesh is generated in two steps. In the outer, unstructured region, linear simplex elements are generated using a metric-driven advancing-front approach. The geometry for this region is the interface between the prismatic layer and the unstructured region, and this interface is sufficiently far away from the true geometry to avoid curved elements. The elements in this unstructured region may still be anisotropic, e.g. for the resolution of shocks, and this requirement poses little challenge as linear anisotropic mesh generation is a relatively mature technology, even in three dimensions [14–16].

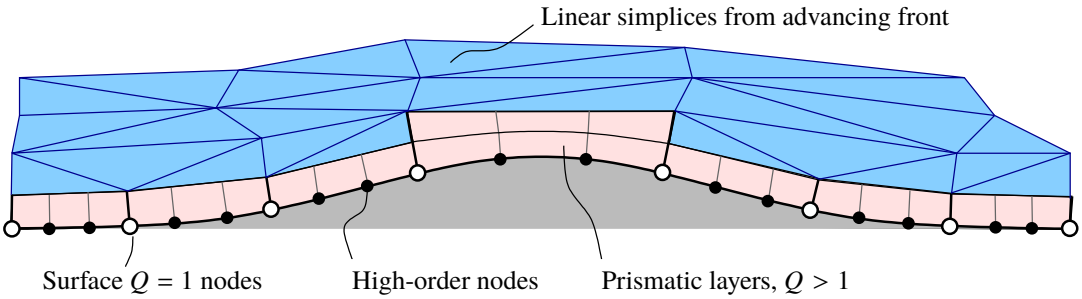


Fig. 1 Metric-driven mesh generation and adaptation for hybrid unstructured/prismatic meshes.

In the inner, prismatic region, the elements are generated by growing layers from a surface mesh. Sizing information for this growth comes from the desired metric field. The structured character of these layers prevents perfect conformity with the metric, although inefficiencies will not be excessive as large changes in the metric are not expected between the start and end of the thin prismatic region. This region ends once linear elements are possible, enabled by attenuation of the curvature through the layers.

For adaptation, meshes in both regions can be regenerated, or nodes can be moved in a metric-conforming manner [17]. The former is expected to be useful in the early stages of adaptation, when mesh changes are large, whereas the latter may be more efficient for small changes later in adaptation. In this work, we consider regeneration only, using prismatic layers and advancing fronts (PLAF).

B. Curved Elements

Curved elements are generally required for high-order discretizations around curved geometries [7, 10, 18]. Generating these elements is complicated by the non-local nature of the problem: curved elements can intrude on the space of adjacent elements, requiring them to be curved as well. For highly-anisotropic meshes, e.g. for high Reynolds-number flows near walls, this can lead to many layers of elements requiring curvature. The difficulty lies in keeping these elements valid, i.e. maintaining positive Jacobians, during the curving. Currently, there is no "native" curved mesh generator that remains robust in the presence of highly anisotropic elements in three dimensions. Instead, curved meshes for such problems are often generated by agglomerating structured linear meshes, at a loss of efficiency and adaptation flexibility, the latter restricted to small node movement or hanging nodes [3].

Once the domain is discretized with a mesh, the geometry of an object is represented by the shape of adjacent elements. When elements are *linear*, i.e. have straight lines for their edges, the geometry representation consists of panels. However, if the actual geometry is not piecewise linear, but instead curved, then the panel representation of the geometry may not be adequate. This occurs for high-order methods, which require more fidelity inside each element, and in such cases, the elements near the geometry need to be curved, as shown in Figure 2.

The underlying problem with a linear geometry in a high-order solver is that the geometric fidelity is inconsistent

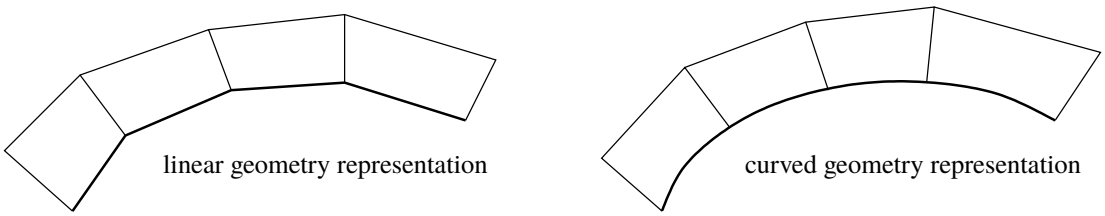


Fig. 2 Linear versus curved geometry representation.

with the fidelity of the solution approximation. Instead of solving for flow over a smooth surface, we instead solve for flow over a paneled geometry with sharp corners. This not only lowers the accuracy of the solution, negating the benefits of high order solution approximation, but can also cause solution convergence problems.

One way to represent curved elements is to use a nonlinear map from a reference element. We use a high-order polynomial to express the shape of the element in physical space. Let Q be the order of this polynomial. Then, choosing a Lagrange basis to express the map, the task reduces to specifying the global-space coordinates of the high-order geometry nodes. Figure 3 illustrates this mapping.

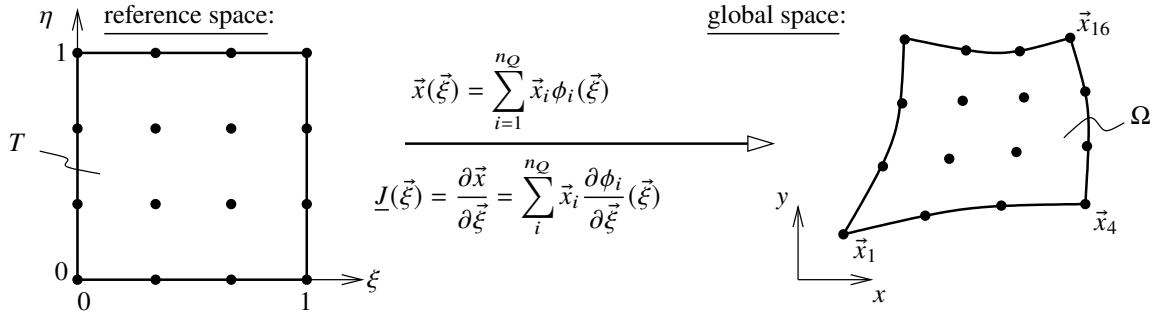


Fig. 3 An order Q geometry mapping for producing a curved element in global space.

The Lagrange basis provides a convenient way to represent the geometry because the Lagrange representation interpolates the geometry between specified points. In reference space, the n_Q nodes are equally spaced. For triangles, we have $n_Q = (Q+1)(Q+2)/2$ points, while for quadrilaterals $n_Q = (Q+1)^2$ points. The Lagrange basis functions are denoted by $\phi_i(\vec{\xi})$.

Note that $\vec{\xi} = (\xi, \eta)$ is the reference space coordinate, and $\vec{x} = (x, y)$ is the global coordinate. The mapping Jacobian matrix, $\underline{J}$, is no longer constant for $Q > 1$ elements, and the determinant J can vary inside the element. Ensuring that $J > 0$ everywhere inside the element is a challenging task for general unstructured elements because negative Jacobians can appear in localized regions of an element even when the high-order node distribution is not itself manifestly inverted.

In two dimensions, the prismatic layer consists of curved quadrilaterals, in which the Lagrange basis is formed from tensor product functions. In three dimensions, the elements become triangular or rectangular prisms, depending on the

surface mesh element type. The Lagrange basis for these prisms is a tensor product of the two-dimensional basis and a one-dimensional Lagrange basis along the normal direction.

C. Equations and Discretization

We apply the proposed prismatic-layer advancing-front mesh generation technique in two dimensions to a discretization of the compressible Navier-Stokes equations,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^s$ is the s-component state vector, consisting of the density, momentum per unit volume, and total energy per unit volume. $\vec{\mathbf{F}} \in \mathbb{R}^{\text{dim} \times s}$ is the flux vector, dim is the spatial dimension, and $\mathbf{S}$ is a source term, which is nonzero when modeling turbulence using Reynolds averaging, for which an additional equation is included [19, 20].

To solve Eqn. 1, we use a discontinuous Galerkin (DG) finite-element method [21–23] with the Roe [24] convective flux and the second form of Bassi and Rebay (BR2) [25] for the viscous flux. After choosing order p polynomial basis functions, the steady-state equations take the form $\mathbf{R}(\mathbf{U}) = \mathbf{0}$, where $\mathbf{U} \in \mathbb{R}^N$ is the discrete state vector, N is the total number of unknowns, and $\mathbf{R}$ is the residual. We solve this system using a Newton-Raphson method with pseudo-time continuation [26] and the generalized minimum residual (GMRES) [27] linear solver, preconditioned by an element-line Jacobi smoother with a coarse-level ($p = 1$) correction [23, 28].

III. Output-Based Metric Optimization

A. Error Estimation

We estimate numerical errors in outputs using an adjoint-weighted residual approach [29, 30]. The adjoint solution for a scalar output is the sensitivity of the output to residual source perturbations. Linearizing the residual and scalar output, $J(\mathbf{U})$, yields a linear system for the discrete adjoint vector, $\mathbf{\Psi} \in \mathbb{R}^N$,

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{\Psi} + \left(\frac{\partial J}{\partial \mathbf{U}}\right)^T = \mathbf{0}, \tag{2}$$

This equation is solved using a transposed version of the preconditioned-GMRES method used in the Newton-Raphson primal solver. To estimate numerical errors, we denote by H a coarse/current discretization space, and by h a fine one, here obtained by increasing the approximation order, $p \rightarrow p + 1$. Denoting by $\mathbf{U}_h^H$ the state prolongated from the coarse to the fine space, the output error can be estimated via

$$J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \approx -\delta \mathbf{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H), \tag{3}$$

where $\delta\boldsymbol{\Psi}_h$ is the fine-space adjoint with its coarse-space projection removed. The error estimate in Eqn. 3 is localized to elements,

$$J \approx \sum_{k=1}^{N_k} -\delta\boldsymbol{\Psi}_{h,k}^T \mathbf{R}_{h,k}(\mathbf{U}_h^H) \quad \Rightarrow \quad \mathcal{E}_{k0} \equiv \left|\delta\boldsymbol{\Psi}_{h,k}^T \mathbf{R}_{h,k}(\mathbf{U}_h^H)\right|, \tag{4}$$

where N_k is the number of elements, and the subscript k denotes components of the adjoint and residual associated with element k.

B. Metric Optimization

The elemental error indicator together with additional error indicators evaluated on sub-elements, via Eqn. 4 with projected adjoints, drive a metric optimization calculation based on MOESS: mesh optimization through error sampling and synthesis [31, 32]. This method computes the metric at linear ($Q = 1$) mesh nodes by equidistributing the marginal error-to-cost ratio over the domain. The elemental error indicator is assumed to vary with element size and orientation according to

$$\mathcal{E}_k = \mathcal{E}_{k0} \exp\left[\mathrm{tr}(\mathcal{R}_k \mathcal{S}_k)\right], \qquad \mathcal{M} = \mathcal{M}_0^{\frac{1}{2}} \exp(\mathcal{S}) \mathcal{M}_0^{\frac{1}{2}}, \tag{5}$$

where $\mathcal{E}_{k0}$ is the indicator on the current element, and $\mathcal{S}_k$ is the step matrix for adjusting the Riemannian metric [33] from the current value, $\mathcal{M}_0$ to the new value, $\mathcal{M}$.

The metric is a symmetric positive-definite tensor that serves as a yardstick for measuring distances in space, and here it is assumed to be a continuous field. At each point in physical space, $\vec{x}$, the metric tensor $\mathcal{M}(\vec{x})$ is used to measure the distance from $\vec{x}$ to another point infinitesimally far away, $\vec{x} + \delta\vec{x}$. The set of points at unit metric distance from $\vec{x}$ is an ellipse: eigenvectors of $\mathcal{M}$ give directions along the principal axes, while the length of each axis (stretching) is the inverse square root of the corresponding eigenvalue. The aspect ratio is the ratio of the largest stretching magnitude to the smalleset.

The rate tensor, $\mathcal{R}_k$, is determined separately for each element through the sampling procedure. Finally, the elemental cost model is the degrees of freedom per unit volume (area in 2D) of the element,

$$C_k = C_{k0} \exp\left[\frac{1}{2}\mathrm{tr}(\mathcal{S}_k)\right], \tag{6}$$

where $C_{k0} = \mathtt{dof}_{k0}$ is the current number of degrees of freedom on element k, e.g. $(p+1)(p+2)/2$ for triangles. Note that a step matrix that reduces an element size has positive trace, and hence increases the cost as expected, since a larger number of smaller elements is required to cover the same area.

A new aspect of MOESS in this work is its application to a hybrid mesh of prisms and simplices, i.e. quadrilaterals

and triangles in two dimensions. The error model rate tensor is determined through sampling local errors arising from refinements of an element in various directions. For a triangle, three edge splits and one uniform refinement comprise the four samples, so that a least-squares fit is needed to compute the three independent components of $\mathcal{R}_k$. For a quadrilateral, horizontal, vertical, and uniform refinements yield three samples, but it was found that the resulting linear system for $\mathcal{R}_k$ was often poorly conditioned. To improve conditioning, two additional samples are generated by splitting the quadrilateral along each of the two diagonals, resulting in two triangles, as illustrated in Figure 4. The

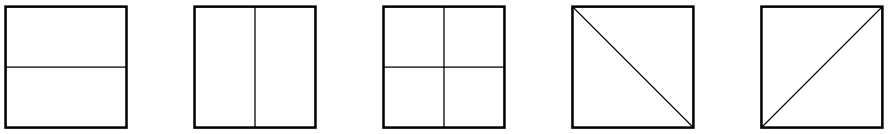


Fig. 4 Quadrilateral element splitting samples for regression of the error model rate tensor in MOESS.

mixing of element shapes during sampling poses no fundamental problem, as state transfer and mesh-implied metric calculations are well-defined for both triangles and quadrilaterals. In addition, the DG approximation space does not require conformity across element boundaries, so that an arbitrary basis can be used. Presently, we use a full-order basis on both triangles and quadrilaterals.

The flow/adjoint solution, mesh optimization, and remeshing are performed several times at either a fixed target cost or with a specified cost growth rate. In this work we use the former approach and compare the accuracy of outputs averaged over several adaptive iterations at a fixed number of degrees of freedom.

IV. Prismatic-Layer Advancing-Front Mesh Adaptation

A. Prismatic Layers

In two dimensions, the geometry is specified by clockwise loops of cubic splines, with corners allowed. The surface mesh consists of a high-order paneling of the loops. The location of linear ($Q = 1$) nodes along the spline is determined by equidistribution of metric-based length, as measured by a numerical integration, along the spline. High-order nodes are then added by uniformly subdividing the arc-length distance between adjacent linear nodes. Corner nodes are constrained to be $Q = 1$ nodes. The surface mesh defines the initial front from which the prismatic, i.e. quadrilateral, mesh is generated.

The first step in growing the prismatic layers is the calculation of outward-pointing normal vectors at every point, linear and high-order, on the front. These normals can be calculated from the spline geometry or, for simplicity in subsequent layers, by averaging panel normals computed from adjacent nodes, linear and high-order. Corner nodes are identified by the change in the normal between adjacent panels. Three different corner types are possible, as illustrated in Figure 5. Right-turn corners require additional normal vectors, computed by equally dividing the corner angle. No

normal vector needs to exist at left-turn corners, at which the two panels form adjacent element sides, as shown in the figure under "left turn 1". However, another left-turn option, "left turn 2" in the figure, is to add another normal vector bisecting the corner, thereby creating two quadrilateral elements. Both options were implemented and the latter was used in the present results due to its better metric conformity.

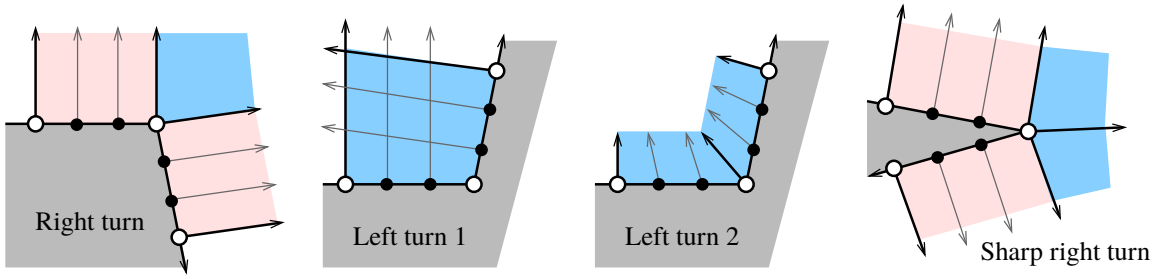


Fig. 5 Corner types in a two-dimensional advancing-front layer. Gray denotes the geometry, pink the standard prismatic elements, and blue the corner-specific elements.

The normal vectors are smoothed, via simple point Jacobi iterations, in direction and magnitude to improve the prism quality. New vertices are added along the normal vectors emanating from each point, linear and high-order, on the front. The layer distance is determined from the specified metric, and high-order vertices are uniformly distributed along this distance. Away from corners, the set of $(Q+1)^2$ vertices between and including those emanating from adjacent $Q = 1$ nodes defines a high-order element. At right-turn corners, as determined by a threshold angle, additional elements are placed in the space between normals, with adjacent normals forming the two sides of a quadrilateral. The remaining two edges for these elements are obtained by intersecting perpendiculars from the normal vector endpoints, and the remaining high-order nodes are uniformly distributed inside the bilinear quadrilateral. At left-turn corners, intersections of normals define the vertices for the single high-order element.

In a metric-conforming mesh, each edge length has unit measure under the metric. The metric length of an edge e is given by [15]

$$L_e^{\mathcal{M}} = \int_e \sqrt{\overrightarrow{dl}^T \mathcal{M} \overrightarrow{dl}}, \tag{7}$$

where $\mathcal{M}$ is the metric field, and $\overrightarrow{dl}$ is the infinitesimal vector along edge e. This integral can be done via quadrature or analytically if the variation of the metric along the edge is known or assumed. In this work, we compute the length twice, using the metric at the edge endpoints, resulting in d_0 and d_1, and then average them as $L_e^{\mathcal{M}} = (d_0 - d_1)/\ln(d_0 - d_1)$ [34].

If the metric is given on a background mesh, it is interpolated to the edge quadrature points using affine-invariant averaging [33]. In the prismatic layer, the metric is used primarily to measure the distance in the front-normal direction, i.e. along the normals. It is also used to measure the distance between points along the new front, which can be

optimized through small changes in the normal direction from each point.

Quadrilateral elements in the layer are formed by grouping $(Q+1)^2$ vertices. As the layers progress, the curvature of each new front is attenuated by adjusting the position of the high-order nodes on the new front. This is done by first computing the linear-element positions of the high-order nodes along the new front. Displacement vectors are then calculated from the normal-calculated positions to these "target" linear locations. If the magnitude of these displacements is smaller than a threshold, the displacement is applied to the curved vertices along the new front, and it is propagated to the interior nodes using linear interpolation. The threshold is a fraction, $f^{\text{att}} = 0.4$ of the local layer height. If the displacement exceeds the threshold, it is scaled to have a magnitude equal to the threshold distance. Figure 6 illustrates this procedure. The value of f^{att} was chosen in a compromise between quickly snapping to linear and preventing negative volumes. No negative volumes were observed with the chosen value in any of the numerical experiments.

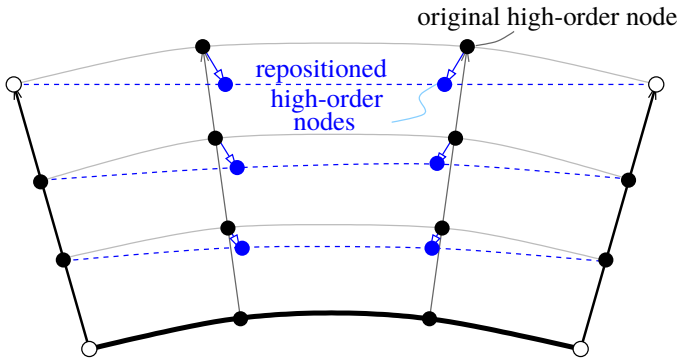


Fig. 6 Attenuation of curvature by node movement towards linear positions.

The active front of the prismatic layers shifts outwards as high-order quadrilaterals are added. To prevent front self-intersections, growth of the layers occurs only from the fronts that are still curved, as illustrated in Figure 1. Thus, the number of layers around a particular geometry is not constant but varies with location on the geometry. Note that the sides of growing prisms are linear, so that these edges can abut the linear, simplex elements. The prismatic layering stops when there are no more curved element tops, so that the entire front is linear. For the high-Reynolds number cases considered in this work, two to three layers were often sufficient to attain the linear front.

B. Simplex Advancing Front

Outside of the prismatic region, meshing is performed using unstructured mesh generation/adaptation, driven by the metric on a background mesh. Standard linear meshing techniques can be used here because the curved elements are confined to the prismatic layer. Presently, we use a simple metric-based advancing-front strategy to grow the unstructured mesh away from the prismatic layer into the domain.

In the two-dimensional problems considered here, the fronts consist of loops of nodes. For geometrical objects surrounded by prismatic layers, the initial fronts are the outside nodes of the layers. For other boundaries, the initial fronts are formed by distributing points along splines of the boundaries, using metric-based distances.

After initialization, the fronts are grown by placing new points and making connections to form new elements, until the entire domain is filled. Many ways exist to place points and grow fronts [35], some incorporating metric information [36], and the method chosen here is based on the following criteria:

- The resulting mesh should reasonably conform to the metric.
- Placement of new points should be robust, avoiding crossing of fronts.
- The resulting mesh should be efficient, minimizing slivers and short metric-based edge lengths.
- The process should be simple, to minimize errors, and computationally inexpensive.

Motivated by these criteria, we have developed a greedy algorithm that consists of the following steps:

Advancing front algorithm

1) Compute outer angles at all nodes on the fronts.
2) Choose the node with the minimum outer angle.
3) Compute directions for placement of new points based on the size of the outer angle.
4) Add zero or more new points using the metric to compute the distance from the original point.
5) Check for intersections with or proximity to other points; merge fronts if needed.
6) Form elements from new points or connections.
7) Return to step 1 if front nodes exist.

Figure 7 illustrates a step in this algorithm. Given a current state of the front, shown as a red line, a node is selected with the minimum outer angle. As shown, this is the minimum geometrical angle, which is appropriate for isotropic elements. The angle is split according to its size, in order to create n_{new} new nodes via the formula

$$n_{\text{new}} = \max\left(0, \lfloor 3\Delta\phi/\pi - 0.75 \rfloor\right),$$

where $\Delta\phi$ is the outer angle and $\lfloor \cdot \rfloor$ is the greatest integer function. This formula splits $\Delta\phi$ into sub-angles of approximately $\pi/3$ radians (60°). In these directions, new nodes are added at distances determined by the metric, computed at the selected node. Merging of fronts occurs if the new nodes run into an existing front. Note, using the minimum angle on the front to identify the next node promotes convexity of the front, which then reduces occurrences of crossings.

Using standard Euclidian measures to choose and split front outer angles is appropriate for isotropic metrics but becomes inefficient and inaccurate when anisotropy is present. This is because anisotropic elements are characterized

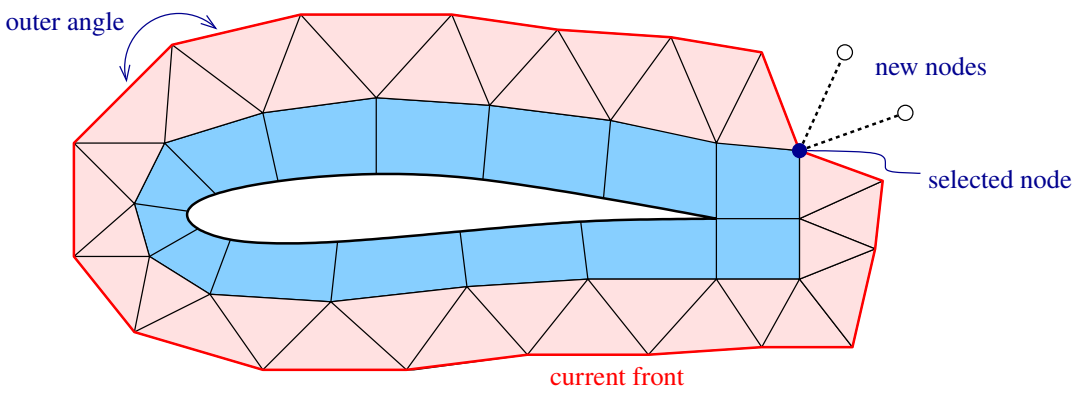


Fig. 7 Advancing-front mesh generation outside of the prismatic layer.

by small/large angles, not necessarily near $\pi/3$ radians. In this work, we therefore incorporate the metric into the outer angle selection and subdivision steps. Figure 8 illustrates this process, which consists of a transformation between the geometrical outer angle, $\Delta\phi$, and an ellipse outer angle, $\Delta\theta$, which approximately represents distance along the ellipse perimeter.

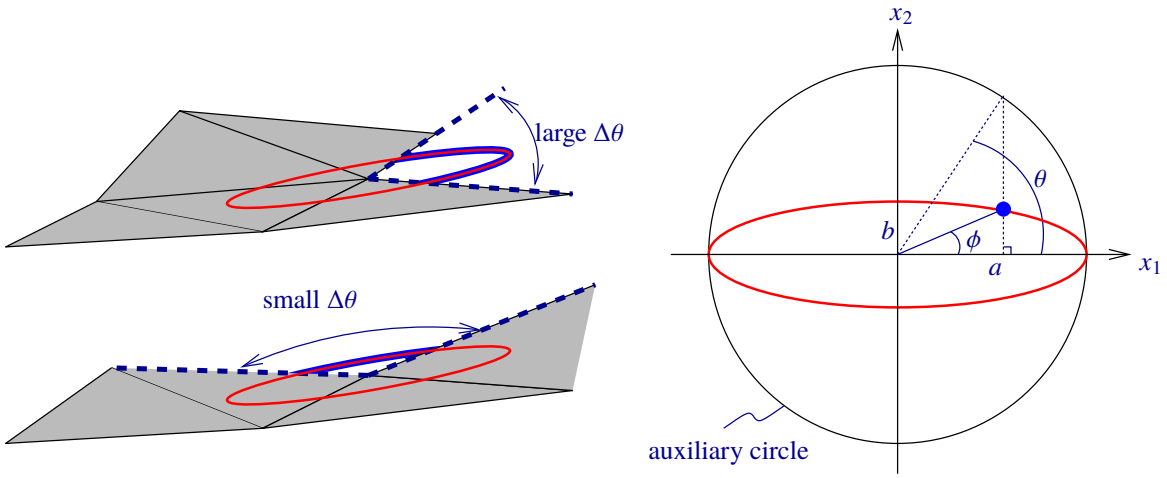


Fig. 8 Metric-based angle definition for advancing-front propagation.

The relationship between ϕ and θ is shown in Figure 8: ϕ measures the angle to a point on the ellipse, whereas θ measures the angle to a point on an auxiliary circle with the same projection onto the semi-major axis of the ellipse. The transformations between these angles are

$$\begin{aligned} x &= a\cos\theta \\ y &= b\sin\theta \end{aligned} \quad \Rightarrow \quad \phi = \tan^{-1}\left(\frac{y}{x}\right) = \tan^{-1}\left(\frac{b}{a}\tan\theta\right) \quad \Rightarrow \quad \theta = \tan^{-1}\left(\frac{a}{b}\tan\phi\right). \tag{8}$$

The eigenvectors and eigenvalues of the metric determine the directions and magnitudes of the major and minor axes.

From the geometry of the front, at a particular front node, ϕ measures the angle between the major axis and the ray emanating from the node. This angle is transformed to the ellipse angle, θ, using Eqn. 8. The difference of the angles between the two rays from the front node then yields $\Delta\theta$, which is used to choose and split the minimum angle.

Front connection and splitting occur when newly added nodes result in, respectively, crossing of two different fronts or a self-crossing of one front. Since the fronts are linear, the checks for intersections are straightforward during node addition. Specifically, a loop over the edges of all loop fronts is performed and intersections between edge segments are tested. In both cases, a new edge is added that is the minimum metric-based length edge between the selected node under consideration on the first front and all nodes on the second front, which is the same one for self intersections. In front connection, the two fronts become one, whereas in front splitting, a self intersection results in two fronts. Figure 9 illustrates this process. Note that all front loops remain clockwise.

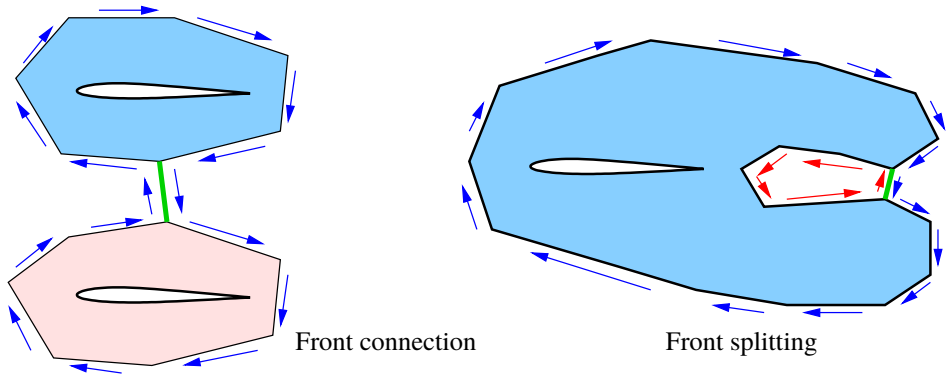


Fig. 9 Connection and splitting of fronts, resulting in one and two fronts, respectively. Arrows indicate the clockwise ordering of loops of nodes in each front.

C. Local Operators

Following the generation of the complete mesh, local mesh operations [34, 37–43] are applied to the simplex elements in order to improve metric conformity and quality. The local operations come in $n_{\text{local}} = 5$ iterations, each one consisting of collapsing, splitting, and swapping, as illustrated in Figure 10.

To guide the operations, we use the metric-based edge length measure, given in Eqn. 7. We also define a combined mesh element quality and conformity measure as

$$Q_k \equiv \frac{A_k^{\mathcal{M}}/A_0}{\frac{1}{3}\sum_{e=1}^{3}\left(L_e^{\mathcal{M}}\right)^2} Q_e, \quad Q_e \equiv \exp\left[-\sum_{e=1}^{3}\left(L_e^{\mathcal{M}}-1\right)^2\right], \tag{9}$$

where the sums are over the element edges. The metric-space area, $A_k^{\mathcal{M}}$, is computed from the physical-space area, A_k,

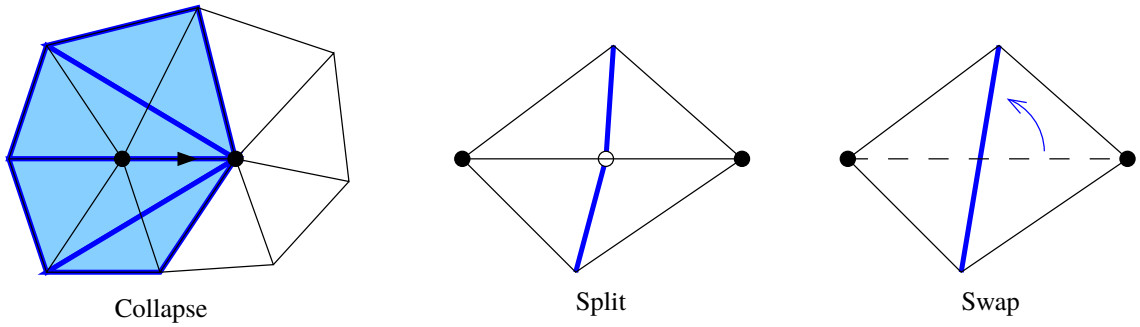


Fig. 10 Local mesh modification operators.

according to

$$A_k = \max\left(A \det \mathcal{M}^{\min}, 0\right), \tag{10}$$

where $\det \mathcal{M}^{\min}$ is the smallest metric determinant computed at the element's three nodes. $A_0 = \sqrt{3}/4$ is the area of a unit equilateral triangle in metric space. The zero bound ensures that the objective function, to be defined in Eqn. 11, remains bounded, while penalizing very small areas. The first fraction in the definition of Q_k is the standard quality measure used in previous works [34, 44], while the multiplication by Q_e incorporates a measure of metric conformity of the edges. Without Q_e, the quality Q_k would not distinguish between elements that are isotropic in metric space but of different sizes. Note that for a metric conforming element, $Q_e = 1$ and $Q_k = 1$.

For choosing between operations, we define an objective function, $J^{\mathcal{Q}}$, over the mesh as

$$J^{\mathcal{Q}} = \sum_k J_k^{\mathcal{Q}}, \quad J_k^{\mathcal{Q}} \equiv \frac{1}{Q_k + \epsilon}, \tag{11}$$

where ϵ is a small positive number on the order of machine precision. The inverse relationship is chosen to heavily penalize element areas close to zero. The goal of the local operations is then to minimize $J^{\mathcal{Q}}$.

We now briefly describe each of the operations. First, a collapse operation targets edges that are short in metric length, $L_e^{\mathcal{M}} < 1/\sqrt{2}$. For a flagged edge, we consider collapsing that edge to either of its two nodes, and also collapsing any edges adjacent to the these two nodes [34]. Only one collapse is chosen, the one that minimizes $J^{\mathcal{Q}}$ of the resulting mesh. In computing $J^{\mathcal{Q}}$ for a potential collapse, the $J_k^{\mathcal{Q}}$ values for the two elements that disappear are subtracted, while the $J_k^{\mathcal{Q}}$ values for all elements that change shape are modified. Following the collapse, all edges affected by the collapse are inactivated fro additional collapse operations in the current iteration.

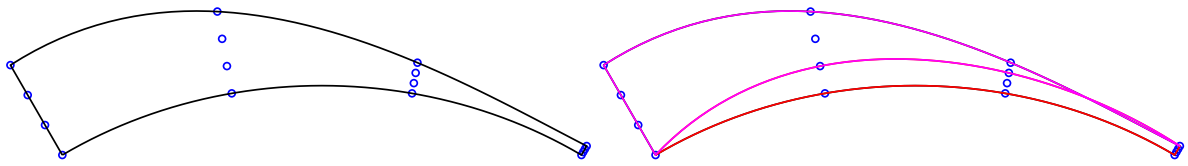
Split operations target edges of long metric length, $L_e^{\mathcal{M}} > \sqrt{2}$. A flagged edge is split in half, resulting in two new elements. The split and newly-added edges are inactivated for further splits in the current iteration.

Finally, swap operations target elements of low quality, $Q_k < 1/\sqrt{2}$ [16]. For each flagged element, three swaps are considered, one for each edge. The swap that results in the minimum J^Q is chosen, as long as this minimum is lower than the pre-swap value. Elements affected by the swap are inactivated for further swaps in the current iteration.

Node movement [17] could be applied as an additional operation, but it is not used in the present work as the meshes obtained without it were of suitable quality. Following mesh generation, if the number of resulting elements differs from the desired target cost by more than 5%, the metric is rescaled isotropically to meet the target cost and the mesh generation process is repeated.

D. The Need for Prisms at High Order

To avoid dealing with hybrid meshes, one option is to split quadrilaterals into triangles. If done in reference space for a high-order element, this approach often works and yields an all-triangular mesh. However, it is not fail-safe, and the resulting high-order triangles can become invalid, through the appearance of negative Jacobians, even when the quadrilateral is valid. Figure 11 illustrates an example of such a case.



(a) Quadrilateral (b) Subdivision into two triangles

Fig. 11 Demonstration of a valid $Q = 3$ quadrilateral resulting in invalid, negative-Jacobian, $Q = 3$ triangles upon a reference-space subdivision along one of its diagonals.

The reason for the appearance of the negative Jacobians is that the full-order space used to approximate the geometry of the triangular subelements differs from the tensor-product space of the quadrilateral geometry. A triangular subset of the high-order quadrilateral nodes therefore does not yield the same mapping, to the extent that negative Jacobians can appear on the triangles for reference-space coordinates where the quadrilateral was valid. In particular, as shown in the example in Figure 11, the tensor-product mapping of the diagonal of the quadrilateral, which remains inside the quadrilateral, does not correspond to the boundary between the two triangles, which is formed by a full-order map of the $Q + 1$ diagonal nodes.

When splitting highly-anisotropic quadrilaterals, such negative Jacobians will occasionally appear over multiple adaptation iterations, without a robust recourse to validity. This is the reason why in this work we do not split the quadrilaterals. Finally, we note that this situation does not occur when splitting a high-order triangle into triangles, or a quadrilateral into quadrilaterals [3]. In these cases, the approximation space for the reference-to-global mapping remains the same on the subelements, so that the global-space coordinates and Jacobian are unaffected.

E. Background Metric Evaluation

In an adaptive setting, the requested metric is given at nodes or elements of the background mesh, which is the mesh from the previous iteration. During the prismatic and advancing-front mesh-generation stages, this metric needs to be evaluated at arbitrary locations in space. To do this, we first determine the element containing the desired point, and the element reference coordinates of the point. To simplify this calculation, the background mesh is snapped to linear, although keeping the mesh curved is also possible and only marginally more expensive when the number of curved elements is small. Linear basis functions are then evaluated at the reference point in the element, and the metric is interpolated from the $Q = 1$ nodes. The interpolation is done using a weighted affine averaging algorithm [33]. To accelerate the search for the containing element, a quad-tree structure is built for the elements of the background mesh. Then, given an arbitrary point, the tree is efficiently traversed starting from the root until a leaf is reached. The few elements with overlap in the leaf are tested until the containing element is found.

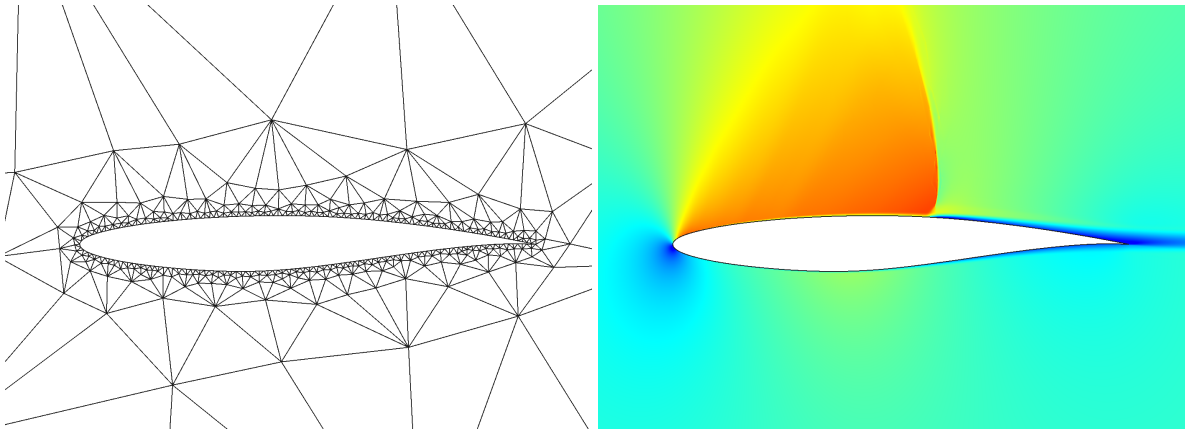
V. Results

This section presents results of tests of the combined prismatic-layer advancing-front (PLAF) mesh generation strategy in an adaptive context. All cases use the Reynolds-averaged Navier-Stokes equations. Mesh optimization is performed using MOESS, with ten iterations at a prescribed number of degrees of freedom. For comparison, we also consider the relatively standard approach of regenerating the entire mesh in an unstructured linear manner, using the bi-dimensional anisotropic mesh generator (BAMG) [45], and then curving it via linear elasticity. Comparisons between BAMG and PLAF are made using final meshes and outputs averaged over the last five adaptive iterations at a fixed target cost.

A. Transonic Flow

As a first test case, we consider turbulent transonic flow over an RAE 2822 airfoil. The conditions are $M = 0.734$, $Re = 6.5 \times 10^6$, and $\alpha = 2.59°$. Figure 12 shows the initial mesh used for adaptation and the Mach number contours. The flow solution exhibits a shock on the upper surface and a thin boundary layer that thickens after passing through the shock. The case is challenging for curved mesh generation because of the presence of highly-anisotropic elements near the curved boundary. We consider $Q = 3$ curved geometry on the airfoil surface, and $p = 2, 3$ approximation orders for the solution.

We are interested in the drag coefficient on the airfoil as the output functional for error estimation and adaptation, and Figure 13 presents the convergence of this output versus degrees of freedom (`dof`). Three target values for `dof` were used: 12k, 24k, and 48k. Shown are results for the two approximation orders, $p = 2$ and $p = 3$, and two meshing strategies: the present prismatic-layer advancing-front (PLAF) method, and unstructured remeshing using the bi-dimensional anisotropic mesh generator (BAMG) and linear-elasticity mesh curving. A "truth" value is also given,



(a) Initial mesh (b) Mach contours, (0 - 1.6)

Fig. 12 Initial mesh and Mach contours for the transonic case.

computed using a $p = 4$ solution on a uniform refinement of one of the finest adapted $p = 3$ meshes. This truth solution had approximately 280k `dof`. We see that after the coarsest degree-of-freedom meshes, both meshing strategies yield outputs that agree well with each other and settle near the truth value.

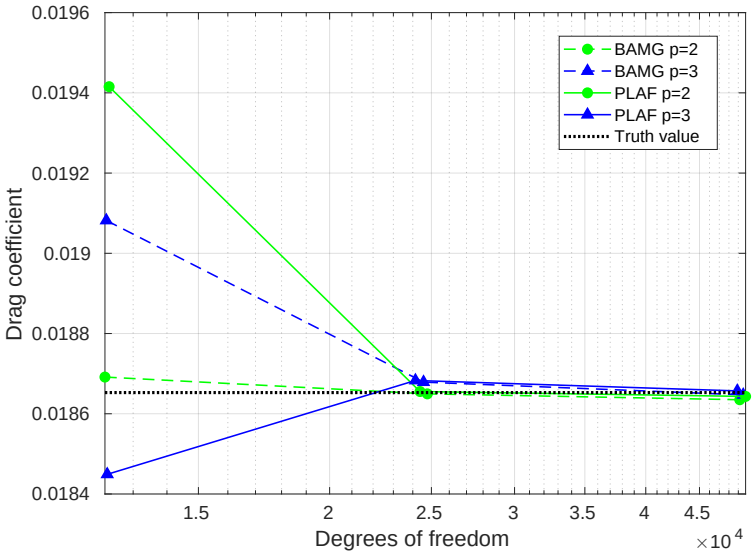
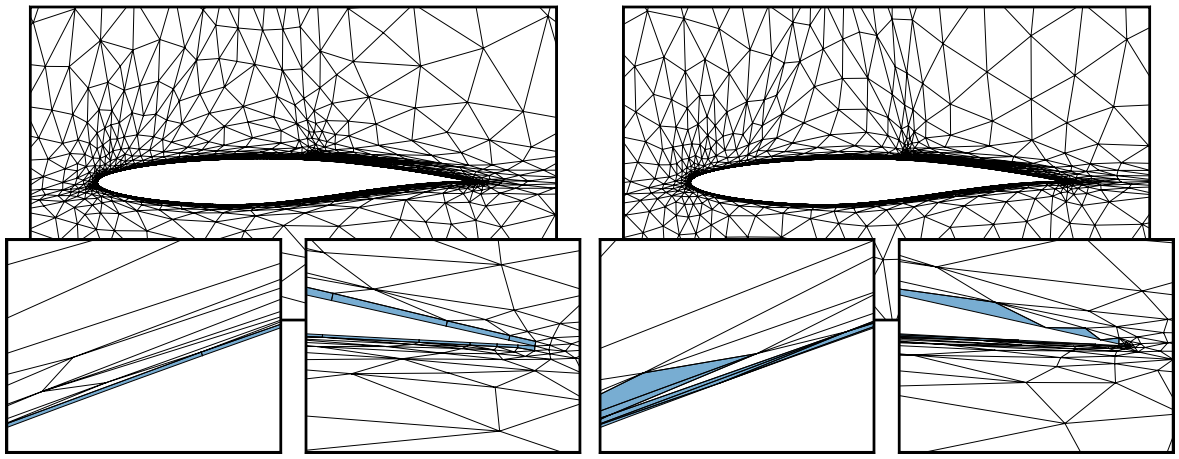


Fig. 13 Drag convergence for the transonic case.

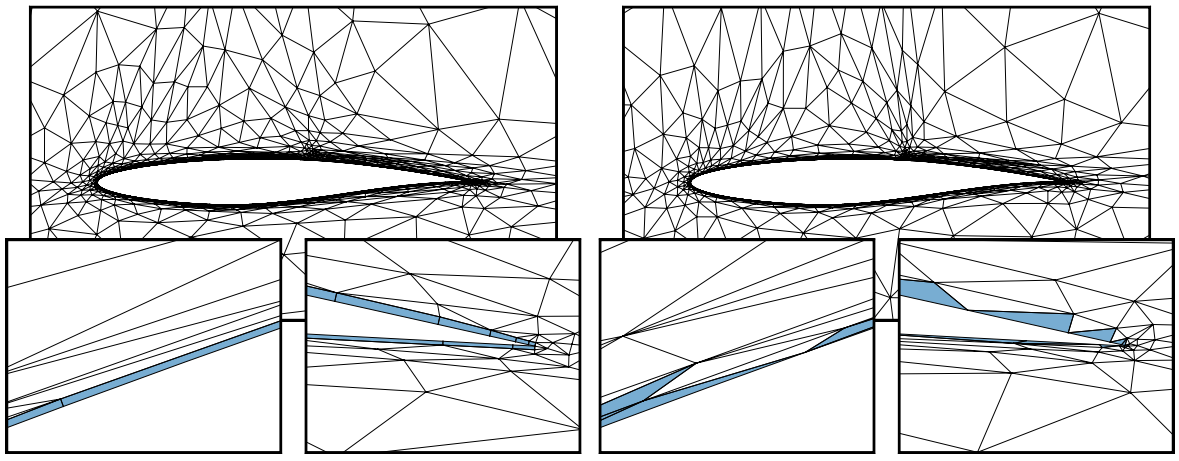
Figures 14 and 15 show the adapted meshes at 24k `dof`. The shaded blue elements denote ones at high geometry order, $Q = 3$, while all other elements are linear. For PLAF, only the quadrilateral elements in the prismatic layer(s) are at high order. For BAMG, linear elasticity is used with all elements at high order, after which as many elements as possible are converted to linear. This results in mostly triangles near the boundary at high order, with some exceptions away from the boundary.

Overall, the meshes at each p value look similar between PLAF and BAMG. The shock is not resolved extensively, except at the root, and this is consistent with previous output-based studies [46]: resolution of the entire shock is not necessary to obtain accurate near-field drag predictions. The boundary layers are resolved with elements of similar anisotropy, with a small difference at the trailing edge, where the anisotropy on the upper surface is larger for PLAF compared to BAMG.



(a) PLAF, $p = 2$ (b) BAMG, $p = 2$

Fig. 14 Adapted meshes for the transonic case, at 24k `dof`, $p = 2$.

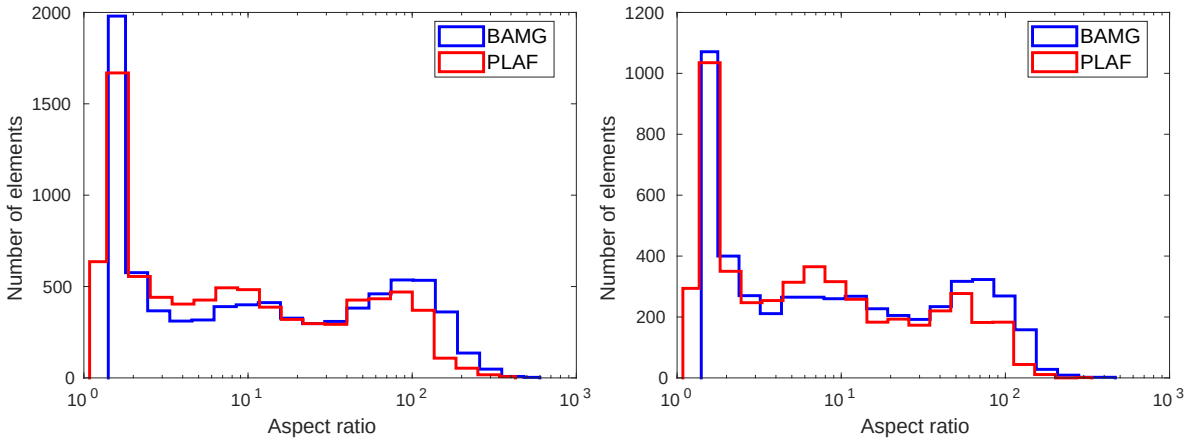


(a) PLAF, $p = 3$ (b) BAMG, $p = 3$

Fig. 15 Adapted meshes for the transonic case, at 24k `dof`, $p = 3$.

Finally, Figure 16 presents histograms of the aspect ratios of the elements for the final adapted meshes at 48k `dof`. A logarithmic scaling is used for the aspect ratio bins. PLAF and BAMG have similar distributions at both approximation orders. The largest aspect ratios at $p = 2$ are slightly higher than those for $p = 3$, and BAMG has slightly more elements at the largest aspect ratios. This is due to lack of node smoothing in PLAF and to performing edge splits last at each

local operator iteration, resulting in slight suboptimality of the PLAF meshes.



(a) $p = 2$ (b) $p = 3$

Fig. 16 Aspect ratio statistics for the transonic case at 48k `dof`.

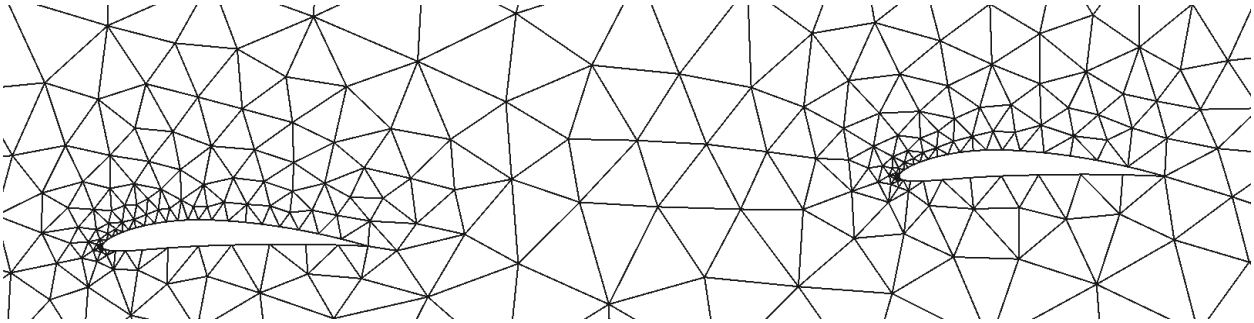
B. Tandem Airfoils

As a next example, we simulate two airfoils flying in tandem. This case further tests the front merging capability of PLAF, since now three fronts exist: one from each airfoil, and one from the farfield. Both airfoils are NACA 5410, and the rear airfoil is about two chords behind the front airfoil. Both are at an angle of attack of $\alpha = 5°$, flying at a Mach number of $M = 0.3$ and Reynolds number $Re = 2 \times 10^6$. The initial mesh for the airfoils is shown in Figure 17(a). This mesh is quite coarse, with no boundary layer or wake resolution. These critical regions are resolved through adaptation.

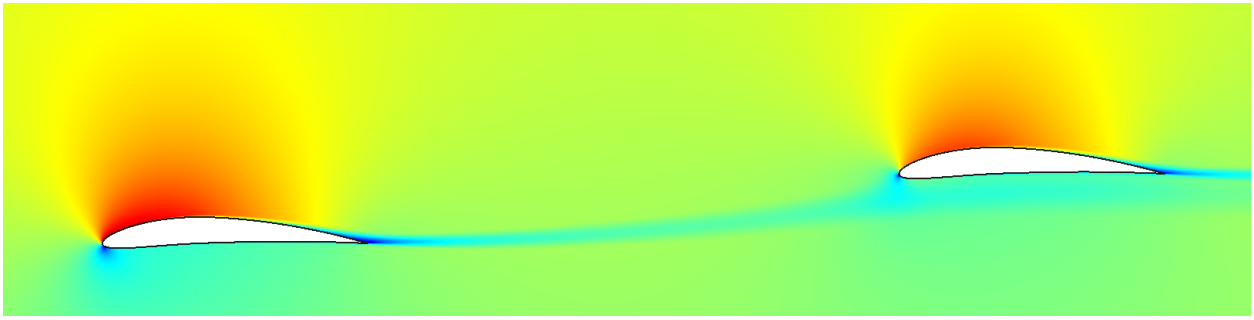
Figure 17(b) shows the Mach number contours, which demonstrate that the wake of the front airfoil hits slightly below the second airfoil. The output of interest in this case is the lift on the second airfoil. A component of the corresponding lift adjoint is shown in Figure 17(c). This is the conservation of x-momentum component, so that we can interpret the figure as the sensitivity of second-airfoil lift to source terms in the x-momentum conservation equation. According to our chosen sign convention, these source terms appear on the left-hand-side of the conservation equations, which is then consistent with the blue (negative value) streak above and in front of the second airfoil. An externally-imposed increase in the x-momentum along this line will increase the lift on the second airfoil.

In PLAF, only 1-2 prismatic layers are typically needed around each airfoil, and then the linear simplex algorithm takes over. The fronts advance from each airfoil, and at some point they meet and connect. Figure 18 shows the resulting front after the connection is made. The regions of high metric-space concavity are quickly filled in after this connection.

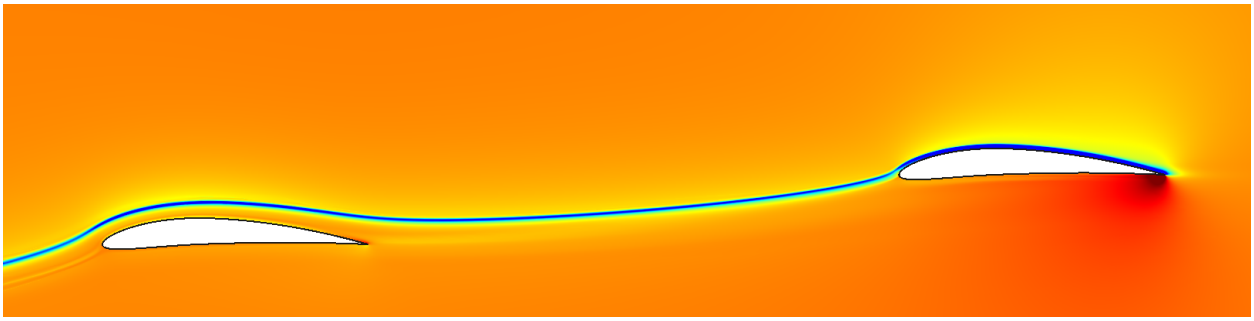
Figure 19 shows the convergence of the output, the lift on the second airfoil, with degrees of freedom. As before, three `dof` targets are used: 12k, 24k, and 48k. At each target, 15 adaptations are run and the results are averaged over the last 5 iterations. The truth value is computed from a uniformly-refined finest adapted mesh, with $p = 4$. For both



(a) Initial mesh



(b) Mach contours, (0 - 0.5)



(c) Lift adjoint contours, conservation of x-momentum component

Fig. 17 Initial mesh and Mach/adjoint contours for the tandem airfoil case.

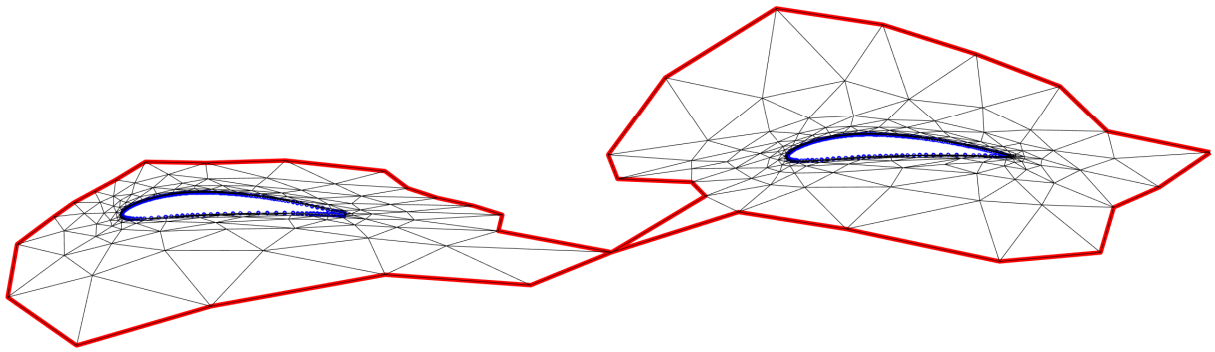


Fig. 18 Intermediate stage of the advancing-front algorithm for the tandem airfoil case, just after connection of two fronts.

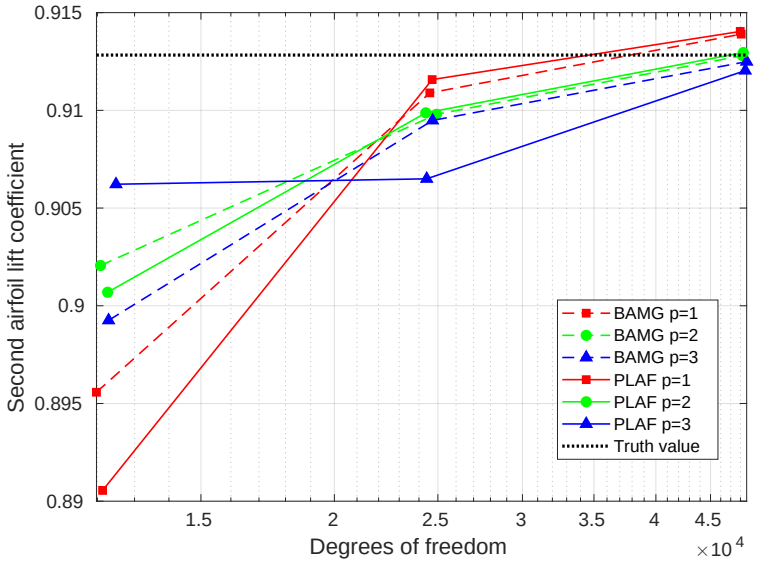


Fig. 19 Second-airfoil lift convergence for the tandem airfoil case.

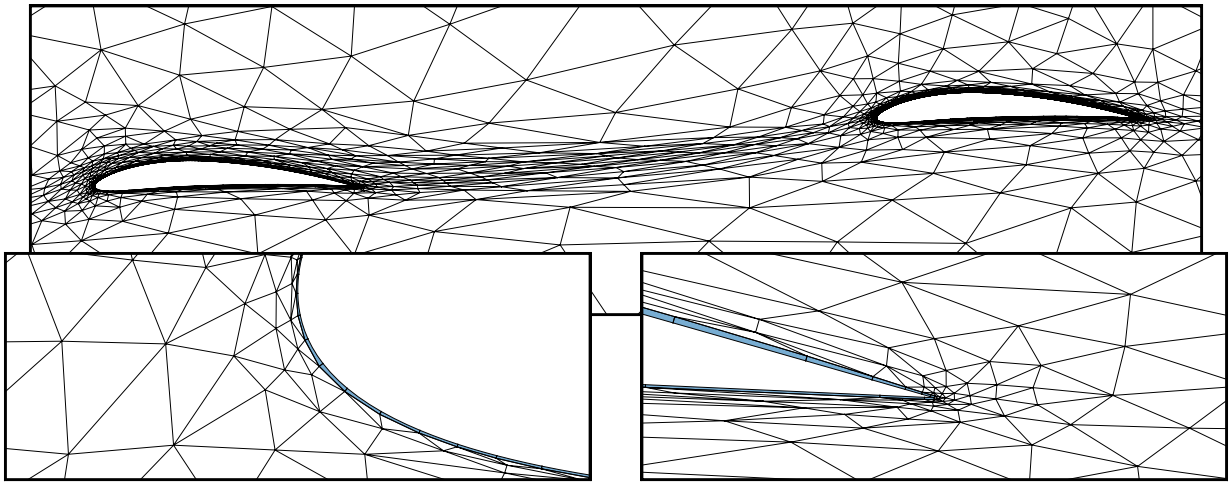
$p = 1$ and $p = 2$, the convergence between PLAF and BAMG is very similar. The lift is under-predicted on the coarse meshes and increases with degrees of freedom, eventually overshooting the truth value, at least for $p = 1$. For $p = 3$, the differences between PLAF and BAMG are larger, even at 24k `dof`. However, by 48k `dof`, both PLAF and BAMG again agree well with each other.

Figures 20 and 21 show the adapted meshes at $p = 2$ and $p = 3$, respectively. High-order curved elements are again highlighted in blue, and zoom boxes pertain to the second airfoil. We see similar refinement patterns overall between PLAF and BAMG: leading/trailing edges, boundary layers, and the wake of the first airfoil up to the second airfoil are all targeted. The anisotropy near the boundary is more consistent for PLAF, which employs prismatic elements grown from the surface, resulting in high-quality quadrilaterals. BAMG relies on the unstructured mesher to conform to the metric, and due to inexactness in satisfying metric conformity, variations appear in the size and stretching of the surface elements.

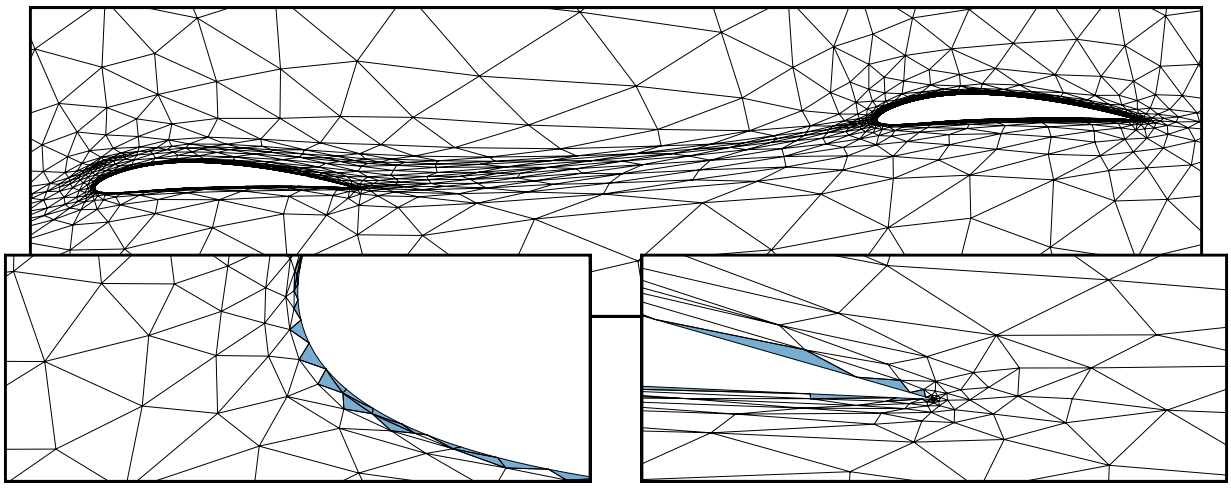
Figure 22 shows aspect-ratio statistics at $p = 2$ and $p = 3$ for the finest adapted meshes generated using PLAF and BAMG. As in the previous example, the histograms are similar between PLAF and BAMG, with minor variations until the largest anisotropy values. There, BAMG generally has slightly more elements at the highest anisotropy compared to PLAF.

C. Three-Element Airfoil

As a final test case, we consider the MDA 30P/30N airfoil at $M = 0.2$, $\alpha = 16°$, $Re = 10^6$. Figure 23 shows the initial mesh, Mach contours, and a component of the lift adjoint. The flow is subsonic and attached, and it is characterized by an interaction between the three elements: the flap, main airfoil, and slat, as illustrated in the adjoint

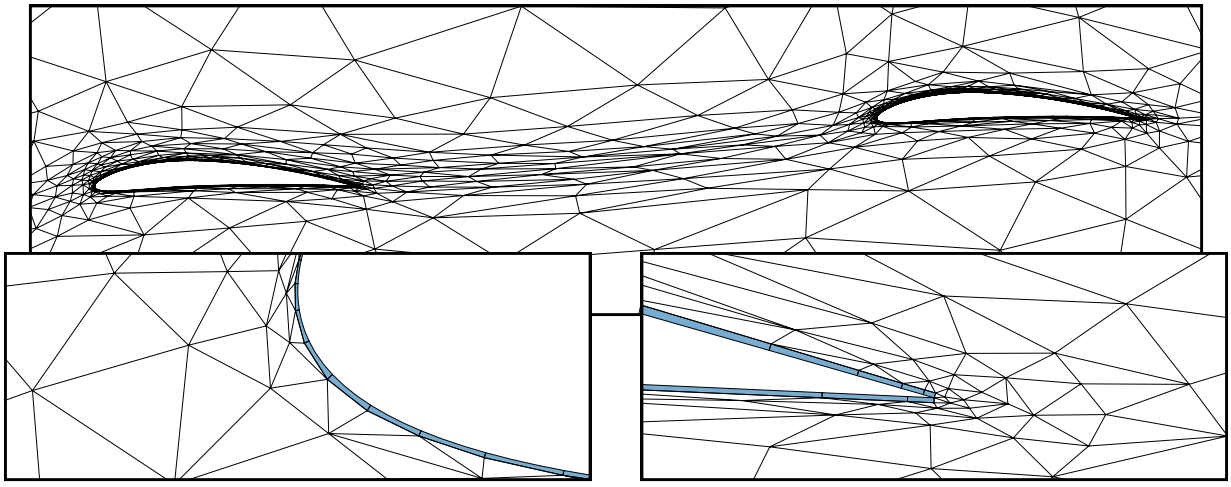


(a) PLAF, $p = 2$

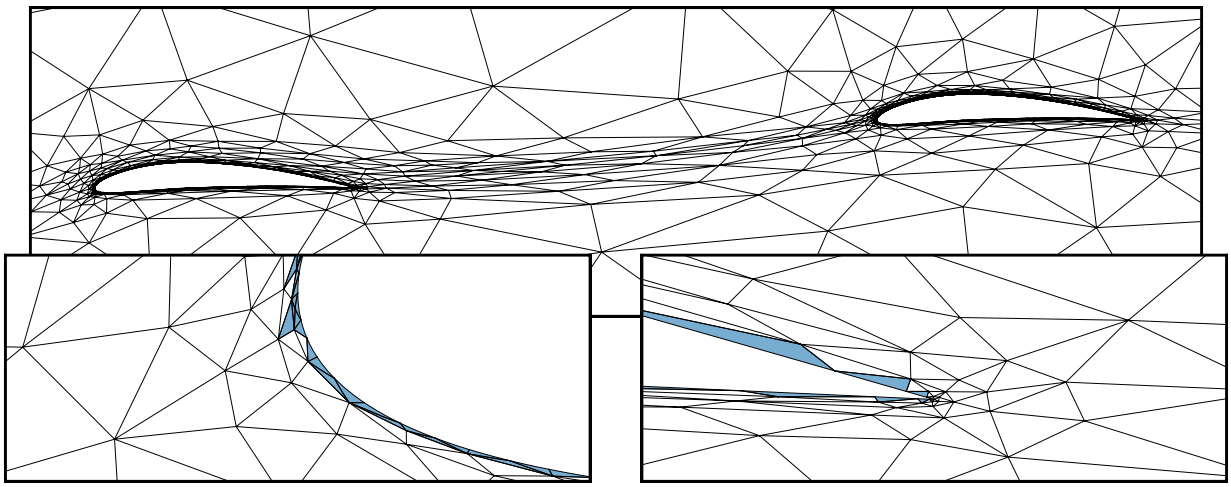


(b) BAMG, $p = 2$

Fig. 20 Adapted meshes for the tandem airfoil case at 24k `dof`, $p = 2$. Zoom boxes are on the second airfoil's leading and trailing edges.

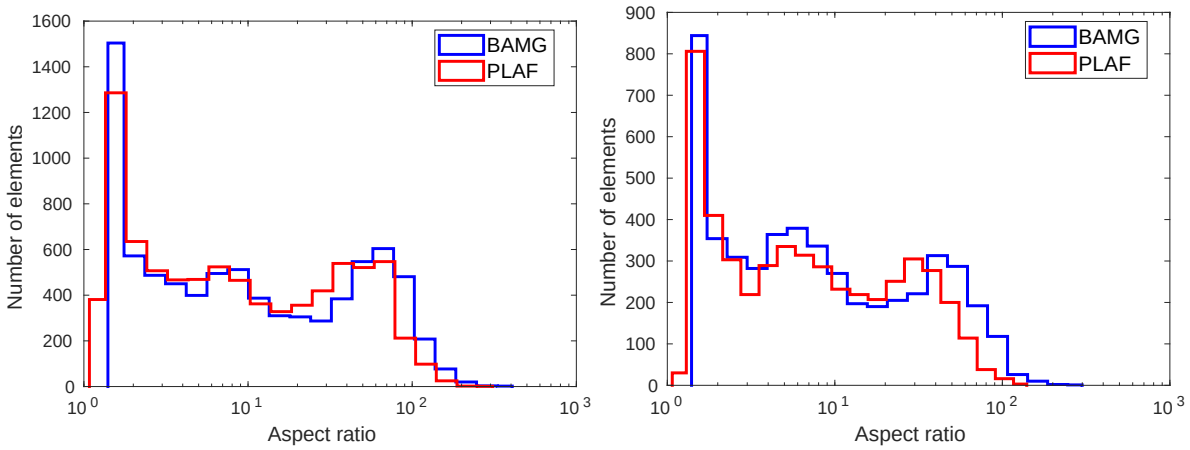


(a) PLAF, $p = 3$



(b) BAMG, $p = 3$

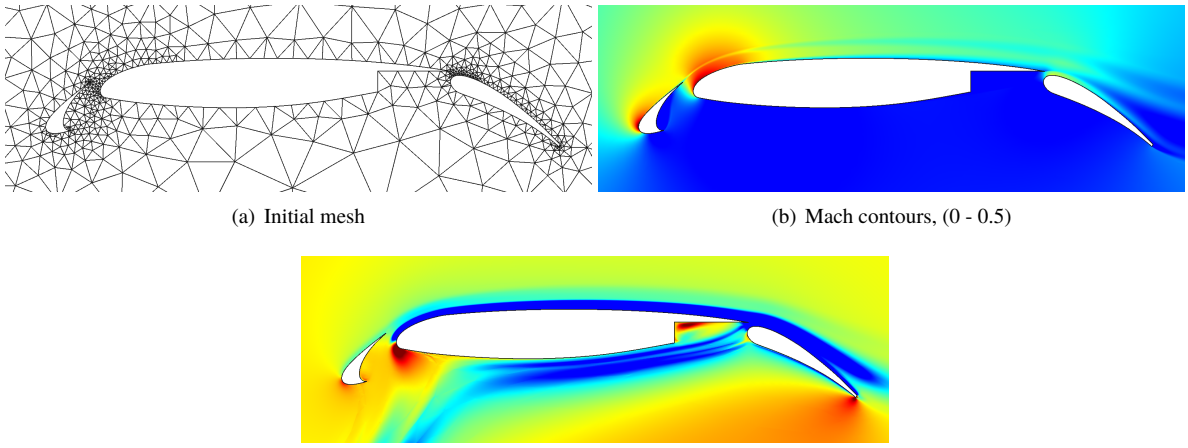
Fig. 21 Adapted meshes for the tandem airfoil case at 24k `dof`, $p = 3$. Zoom boxes are on the second airfoil's leading and trailing edges.



(a) $p = 2$ (b) $p = 3$

Fig. 22 Aspect ratio statistics for the tandem airfoil case at 48k `dof`.

sensitivity field. The geometry is more intricate than the previous examples, with blunt trailing edges, a cove on the main element, and proximity of the surfaces. No robustness problems were observed in the PLAF mesh generator, however, even for the moderate Reynolds number of 1 million, due to the use of linear attenuation and growth only from prisms that were still curved.



(a) Initial mesh (b) Mach contours, (0 - 0.5)

(c) Lift adjoint contours, conservation of x-momentum component

Fig. 23 Initial mesh and Mach/adjoint contours for the three-element airfoil case.

Figure 24 shows convergence of the output of interest, the combined lift of all three elements, versus target degrees of freedom: 12k, 24k, and 48k. BAMG and PLAF are similar, with the largest difference at $p = 1$, for which PLAF converges more quickly to the truth value computed on a refined mesh and order.

Figures 25 and 26 show the adapted meshes at $p = 2$ and $p = 3$, respectively. Important for the output prediction in this case are the leading/trailing edges of each element as well as the boundary layers, particularly on the upper surface

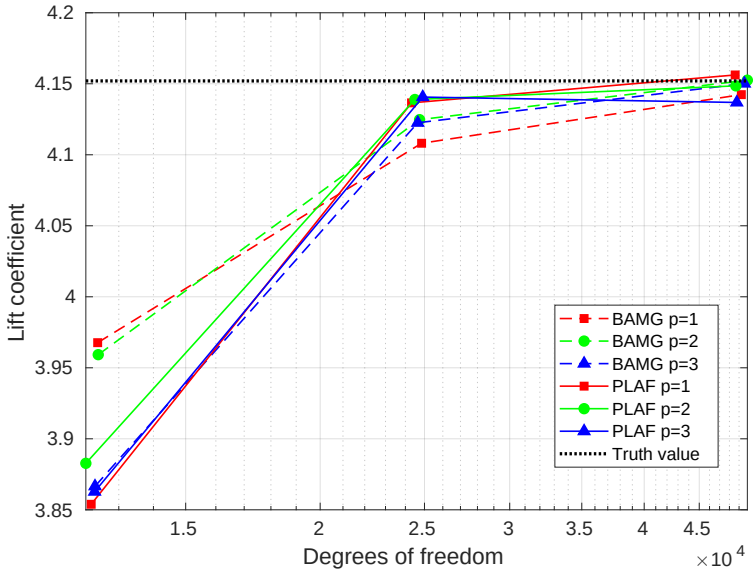
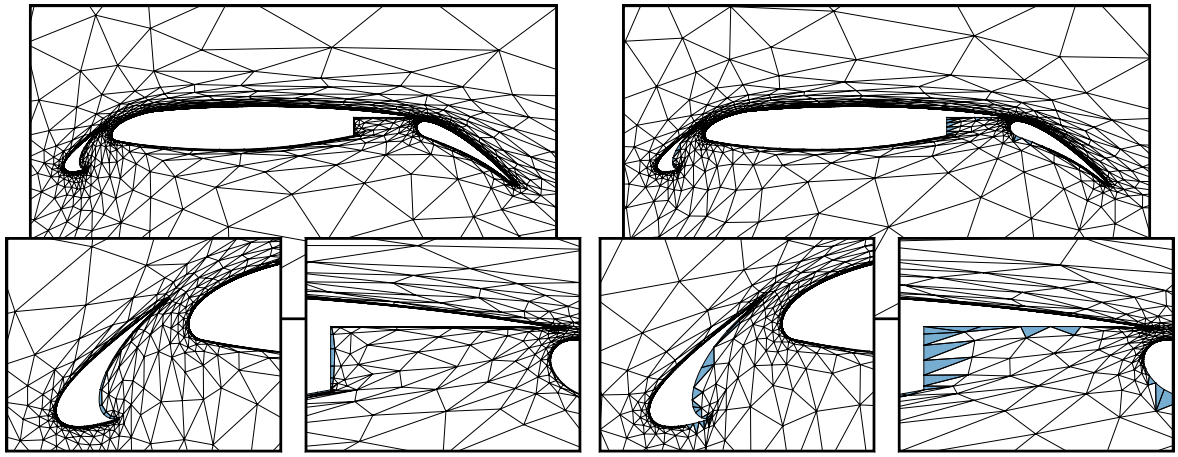


Fig. 24 Lift convergence for the three-element airfoil case.

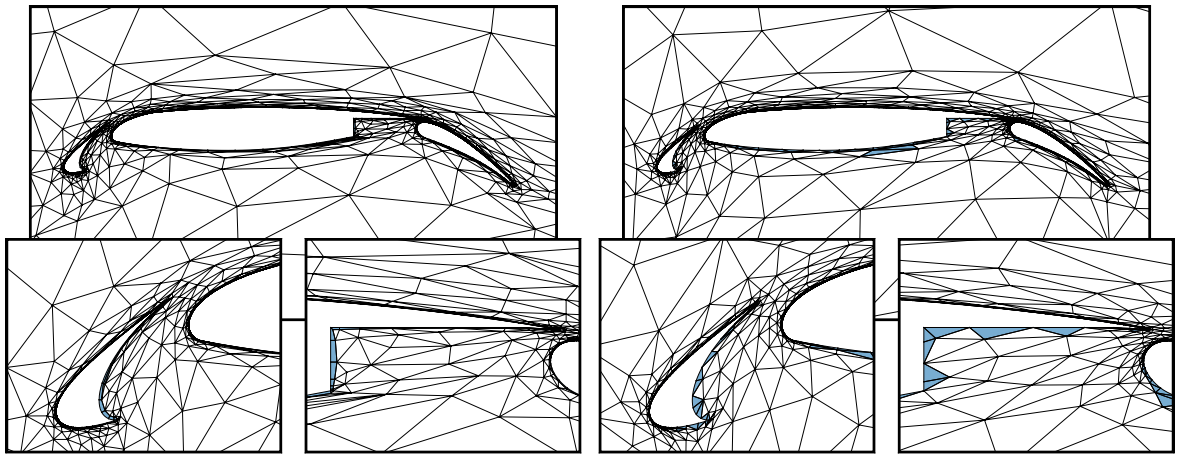
of the main element. The cove contains a low-speed recirculation and does not require much refinement. For both orders, the PLAF and BAMG meshes are nearly indistinguishable. The boundary-layer resolution and anisotropy are similar, as are regions of more isotropic refinement ahead of the slat. A small difference is in the cove, where PLAF places smaller, more anisotropic elements near the boundary compared to the larger, more isotropic elements generated by BAMG.



(a) PLAF, $p = 2$ (b) BAMG, $p = 2$

Fig. 25 Adapted meshes for the transonic case, at 24k `dof`, $p = 2$.

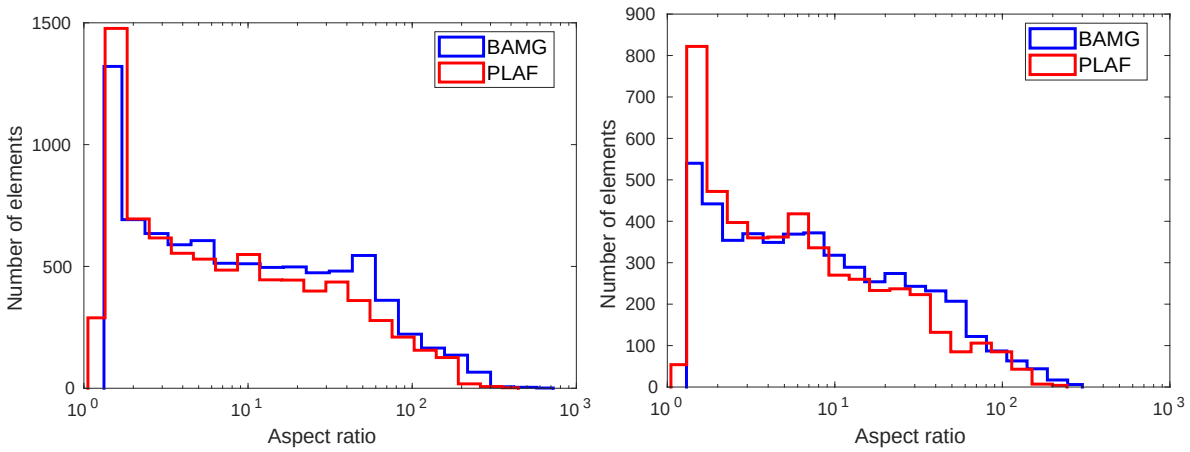
Finally, Figure 27 shows the aspect ratio distributions for $p = 2$ and $p = 3$ on the final adapted meshes at 48k `dof`. As in the previous examples, the distributions are similar between BAMG and PLAF. PLAF has relatively more elements in the lowest aspect-ratio bin, whereas BAMG has more in some of the intermediate/high aspect-ratio bins.



(a) PLAF, $p = 3$ (b) BAMG, $p = 3$

Fig. 26 Adapted meshes for the transonic case, at 24k `dof`, $p = 3$.

However, these differences are minor and vary with adaptive iteration.



(a) $p = 2$ (b) $p = 3$

Fig. 27 Aspect ratio statistics for the three-element airfoil case at 48k `dof`.

VI. Conclusions

This paper presents the formulation and results of a new method for generating meshes with curved anisotropic elements. Such meshes are required for high-order discretizations of many aerodynamics problems, particularly those at high Reynolds numbers, but they are difficult to generate due to the propensity for element inversions near curved boundaries when the anisotropy is high. The method introduced generates a mesh in two stages: the first consists of a layering of prismatic elements from the curved surface to a distance away from the boundary that enables the use of linear faces; the second is an unstructured meshing of the remainder of the domain using only linear elements. The first

stage is robust against inversions due to the orthogonality of the prismatic layers, which grow normally outward from the geometry. The second stage is robust due to the use of linear elements.

Both stages use metric information to size the elements, so that they are suitable for mesh adaptation using error estimation and metric optimization. The unstructured stage consists of an advancing-front strategy that uses the metric for angle and distance measurements. The prismatic growth stage attenuates the curved surface information to linear in a small number of prismatic layers, depending on the anisotropy, and growth only continues from elements that are curved. This greatly improves the robustness by preventing unnecessary growth of linear prisms, and it improves efficiency by reducing the number of curved elements. Local operations, using novel optimization metrics, are performed on the linear simplices to further improve metric conformity and mesh quality.

Results for several high-Reynolds number turbulent-flow cases, including in transonic conditions and over multiple-element airfoils, compare the proposed approach, PLAF, to unstructured remeshing using BAMG and high-order curving via linear elasticity. These comparisons are performed in an output-based adaptive setting with metric optimization. The observed output convergence histories are similar between the two methods, at least for the orders and cases tested. An advantage of PLAF is that it does not require a separate curving step due to its use of curved prisms, which also more easily maintain high quality. In addition, the outer advancing-front mesher is simple to implement and surprisingly robust.

An extension of this approach is to three dimensions, for which the complexity of each operation will increase, but for which the robustness of the overall process in maintaining positive-Jacobian elements is expected to remain. The approach for three dimensions is to first generate a metric-conforming surface mesh of high-order triangles and/or quadrilaterals. As in the two-dimensional case, normal vectors will be defined at each linear and high-order vertex on the surface, and points will be placed along these normals. Corners and edges will be specified by the geometry or obtained by measuring changes in the normal across surface elements. Elements will be added or taken away from the edge/corner cases as needed to keep a valid front. Curvature of the elements will attenuate with increasing layer until linear facets are possible on the entire front, at which point linear unstructured meshing, e.g. via a three-dimensional advancing front strategy, will take over. This direction is the subject of future work.

Acknowledgement

This work was supported in part by a Phase II STTR project, under contract N68335-22-C-0171.

References

[1] Leicht, T., and Hartmann, R., "Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations," *International Journal for Numerical Methods in Fluids*, Vol. 56, 2008, pp. 2111–2138. https://doi.org/https://doi.org/10.1002/fld.1608.

[2] Leicht, T., and Hartmann, R., "Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations," *Journal of Computational Physics*, Vol. 229, 2010, pp. 7344–7360. https://doi.org/https://doi.org/10.1016/j.jcp.2010.06.019.

[3] Ceze, M. A., and Fidkowski, K. J., "An anisotropic hp-adaptation framework for functional prediction," *AIAA Journal*, Vol. 51, 2013, pp. 492–509. https://doi.org/10.2514/1.J051845.

[4] Ceze, M. A., and Fidkowski, K. J., "Drag Prediction Using Adaptive Discontinuous Finite Elements," AIAA Paper 2013-0051, 2013. https://doi.org/10.2514/6.2013-51.

[5] Fidkowski, K. J., "Three-Dimensional Benchmark RANS Computations Using Discontinuous Finite Elements on Solution-Adapted Meshes," AIAA Paper 2018–1104, 2018. https://doi.org/10.2514/6.2018-1104.

[6] Li, X., Shephard, M. S., and Beall, M. W., "Accounting for curved domains in mesh adaptation," *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 247–276. https://doi.org/https://doi.org/10.1002/nme.772.

[7] Persson, P.-O., and Peraire, J., "Curved mesh generation and mesh refinement using Lagrangian solid mechanics," AIAA Paper 2009-0949, 2009. https://doi.org/https://doi.org/10.2514/6.2009-949.

[8] Johnen, A., Remacle, J.-F., and Geuzaine, C., "Geometrical validity of curvilinear finite elements," *Journal of Computational Physics*, Vol. 233, 2013, pp. 359–372. https://doi.org/https://doi.org/10.1016/j.jcp.2012.08.051.

[9] Abgrall, R., Dobrzynski, C., and Froehly, A., "A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems," *International Journal for Numerical Methods in Fluids*, Vol. 76, No. 4, 2014, pp. 246–266. https://doi.org/10.1002/fld.3932.

[10] Ruiz-Gironés, E., Sarrate, J., and Roca, X., "Generation of curved high-order meshes with optimal quality and geometric accuracy," *Procedia engineering*, Vol. 163, 2016, pp. 315–327. https://doi.org/https://doi.org/10.1016/j.proeng.2016.11.108.

[11] Karman, S. L., Ervin, J., Glasby, R. S., and Stefanski, D. L., "High-order mesh curving using WCN mesh optimization," AIAA Paper 2016-3178, 2016. https://doi.org/10.2514/6.2016-3178.

[12] Feuillet, R., Loseille, A., and Alauzet, F., "Optimization of P2 meshes and applications," *Computer-Aided Design*, Vol. 124, 2020, p. 102846. https://doi.org/https://doi.org/10.1016/j.cad.2020.102846, URL https://www.sciencedirect.com/science/article/pii/S0010448520300397.

[13] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA Contractor Report 218178, 2014.

[14] Park, M. A., and Darmofal, D. L., "Parallel Anisotropic Tetrahedral Adaptation," AIAA Paper 2008-917, 2008. https://doi.org/https://doi.org/10.2514/6.2008-917.

[15] Ibanez, D., Barral, N., Krakos, J., Loseille, A., Michal, T., and Park, M., "First benchmark of the Unstructured Grid Adaptation Working Group," *Procedia Engineering*, Vol. 203, 2017, pp. 154–166. https://doi.org/https://doi.org/10.1016/j.proeng.2017.09.

800, URL https://www.sciencedirect.com/science/article/pii/S1877705817343618, 26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain.

[16] Galbraith, M. C., Caplan, P. C., Carson, H. A., Park, M. A., Balan, A., Anderson, W. K., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., "Verification of Unstructured Grid Adaptation Components," *AIAA Journal*, Vol. 58, No. 9, 2020, pp. 3947–3962. https://doi.org/10.2514/1.J058783.

[17] Fidkowski, K. J., "Output-Based Mesh Optimization Using Metric-Conforming Node Movement," AIAA Paper 2023–2369, 2023. https://doi.org/10.2514/6.2023-2369.

[18] Bassi, F., and Rebay, S., "High–order accurate discontinuous finite element solution of the 2-D Euler equations," *Journal of Computational Physics*, Vol. 138, 1997, pp. 251–285. https://doi.org/https://doi.org/10.1006/jcph.1997.5454.

[19] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902, 2012.

[20] Ceze, M. A., and Fidkowski, K. J., "High-Order Output-Based Adaptive Simulations of Turbulent Flow in Two Dimensions," AIAA Paper 2015–1532, 2015. https://doi.org/10.2514/6.2015-1532.

[21] Reed, W., and Hill, T., "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos Scientific Laboratory Technical Report LA-UR-73-479, 1973.

[22] Cockburn, B., and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261. https://doi.org/https://doi.org/10.1023/A:1012873910884.

[23] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "*p*-Multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113. https://doi.org/10.1016/j.jcp.2005.01.005.

[24] Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372. https://doi.org/https://doi.org/10.1016/0021-9991(81)90128-5.

[25] Bassi, F., and Rebay, S., "Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 197–207. https://doi.org/https://doi.org/10.1002/fld.338.

[26] Ceze, M. A., and Fidkowski, K. J., "Constrained pseudo-transient continuation," *International Journal for Numerical Methods in Engineering*, Vol. 102, 2015, pp. 1683–1703. https://doi.org/10.1002/nme.4858.

[27] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific Computing*, Vol. 7, No. 3, 1986, pp. 856–869. https://doi.org/https://doi.org/10.1137/0907058.

[28] Persson, P.-O., and Peraire, J., "Newton-GMRES Preconditioning for Discontinuous Galerkin Discretizations of the Navier-Stokes Equations," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 2709–2733. https://doi.org/https://doi.org/10.1137/070692108.

[29] Becker, R., and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, edited by A. Iserles, Cambridge University Press, 2001, pp. 1–102. https://doi.org/https://doi.org/10.1017/S0962492901000010.

[30] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. https://doi.org/10.2514/1.J050073.

[31] Yano, M., and Darmofal, D., "An optimization-based framework for anisotropic simplex mesh adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, p. 7626–7649. https://doi.org/10.1016/j.jcp.2012.06.040.

[32] Fidkowski, K. J., "A Local Sampling Approach to Anisotropic Metric-Based Mesh Optimization," AIAA Paper 2016–0835, 2016. https://doi.org/10.2514/6.2016-0835.

[33] Pennec, X., Fillard, P., and Ayache, N., "A Riemannian framework for tensor computing," *International Journal of Computer Vision*, Vol. 66, No. 1, 2006, pp. 41–66. https://doi.org/https://doi.org/10.1007/s11263-005-3222-z.

[34] Michal, T., and Krakos, J., "Anisotropic Mesh Adaptation Through Edge Primitive Operations," AIAA Paper 2012-0159, 2012. https://doi.org/https://doi.org/10.2514/6.2012-159.

[35] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive remeshing for compressible flow computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466. https://doi.org/https://doi.org/10.1016/0021-9991(87)90093-3.

[36] Marcum, D., and Alauzet, F., "Aligned Metric-Based Anisotropic Solution Adaptive Mesh Generation," *Procedia Engineering*, Vol. 82, 2014, pp. 428–444. https://doi.org/https://doi.org/10.1016/j.proeng.2014.10.402.

[37] Freitag, L. A., and Ollivier-Gooch, C., "Tetrahedral Mesh Improvement Using Swapping and Smoothing," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 21, 1997, pp. 3979–4002. https://doi.org/https://doi.org/10.1002/(SICI)1097-0207(19971115)40:21%3C3979::AID-NME251%3E3.0.CO;2-9.

[38] Castro-Diaz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., "Anisotropic unstructured mesh adaptation for flow simulations," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, 1997, pp. 475–491. https://doi.org/https://doi.org/10.1002/(SICI)1097-0363(19970830)25:4%3C475::AID-FLD575%3E3.0.CO;2-6.

[39] Wood, W. A., and Kleb, W. L., "On Multi-dimensional Unstructured Mesh Adaptation," AIAA Paper 99-3254, 1999. https://doi.org/https://doi.org/10.2514/6.1999-3254.

[40] Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M., and Vallet, M.-G., "Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles," *International Journal*

for Numerical Methods in Fluids, Vol. 32, No. 6, 2000, pp. 725–744. https://doi.org/https://doi.org/10.1002/(SICI)1097-0363(20000330)32:6%3C725::AID-FLD935%3E3.0.CO;2-4.

[41] Baker, T. J., "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3-4, 1997, pp. 243–273. https://doi.org/https://doi.org/10.1016/S0168-874X(96)00032-7.

[42] Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003-3849, 2003. https://doi.org/https://doi.org/10.2514/6.2003-3849.

[43] Caplan, P. C., Haimes, R., Darmofal, D. L., and Galbraith, M. C., "Four-Dimensional Anisotropic Mesh Adaptation," *Computer-Aided Design*, Vol. 129, 2020, p. 102915. https://doi.org/https://doi.org/10.1016/j.cad.2020.102915, URL https://www.sciencedirect.com/science/article/pii/S0010448520301081.

[44] Knupp, P. M., "Algebraic mesh quality metrics for unstructured initial meshes," *Finite Elements in Analysis and Design*, Vol. 39, No. 3, 2003, pp. 217–241. https://doi.org/https://doi.org/10.1016/S0168-874X(02)00070-7.

[45] Borouchaki, H., George, P., Hecht, F., Laug, P., and Saltel, E., "Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes," INRIA-Rocquencourt, France. Tech Report No. 2741, 1995.

[46] Venditti, D. A., and Darmofal, D. L., "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46. https://doi.org/https://doi.org/10.1016/S0021-9991(03)00074-3.