A closed advancing-layer method for generating curved boundary layer mesh

Rémi Feuillet *, David Marcum † and Frédéric Alauzet‡
GAMMA3 Team, INRIA Saclay Ile-de-France, Palaiseau, France
Center for Advanced Vehicular Systems (CAVS), Mississippi State University, MS 39762 USA

A closed advancing-layer method for generating high-order (P2) high-aspect-ratio elements in the boundary layer (BL) region is presented. This approach efficiently and naturally produces a smooth anisotropic blending between colliding BL fronts. It is also able to guarantee mesh generation validity and robustness as it starts from an existing valid mesh and always considers mesh validity and quality prior to mesh modifications. In addition, it provides a robust strategy to couple unstructured anisotropic mesh adaptation and high-aspect-ratio element pseudo-structured BL meshes. Contrary to classical high-order approaches that deform an existing linear BL mesh into a curved BL mesh, the proposed method is directly generating the curved BL mesh. This strategy provides the ability to check validity and quality of the high-order elements prior to including them in the BL mesh. In the same manner the mesh deformation approach for including the BL mesh in the overall mesh allows checking of validity and quality of the outer region high-order elements. This approach utilizes a recently developed connectivity-change based P2 moving mesh strategy for deforming the curved volume mesh as the BL is inflated. In regards to the high-order BL mesh generation, it features state-of-art capabilities, including; optimal normal evaluation, normal smoothing, blended BL termination, mixed-elements BL, varying growth rate, and BL imprinting on curved surfaces. First results obtained on simple geometries are presented to assess the proposed strategy.

I. Introduction

Unstructured meshes are widely used in large-scale computational field simulation, and in particular, computational fluid dynamic (CFD), applications to help solve real world problems found in industry and government. In CFD applications, viscous boundary-layer (BL) regions are typically prominent near the vehicle or component surfaces and must be resolved to capture the relevant physics. BL regions often have very stringent numerical requirements as they involve high-gradient and non-linear physics and usually include turbulence. These regions are in general known a priori and are ideally suited to a pseudo-structured approach. Optimal meshes are highly aligned, precisely spaced and very structured in at least the normal direction.

However, in the last decade high-order resolution methods (continuous Galerkin, 12, 19 discontinuous Galerkin, 9 spectral differences, 38 ...) have been used. To preserve the high-order of convergence of these methods, it is required to have a high-order representation of the geometry in the mesh. These meshes are curved in order to fit at best with the boundary of the studied geometric shape. If Euler meshes (without BL mesh) are considered, we generally obtain valid high-order elements almost everywhere in the domain when the surface mesh is curved using simple mesh deformation algorithm. The mesh deformation is coupled with a mesh optimization algorithm to correct the few invalid elements. But, if such mesh contains a BL mesh (viscous mesh) in geometry region with curvature, then the new curved boundary faces are crossing many layer of elements in the BL mesh. Maintaining the high-order mesh validity requires to generated highly curved and highly stretched elements in the BL region.

*PhD student, Remi.Feuillet@inria.fr
†Professor, marcum@cavs.msstate.edu
‡Senior Researcher, Frederic.Alauzet@inria.fr
Overall methodologies used to generate the high-aspect-ratio elements in the BL mesh region commonly rely on an advancing-layer/normal type process in which the boundary surface mesh is inflated one layer at a time. There are two primary branches of techniques that can be classified as open- or closed-advancing-layer methods for generating valid non-intersecting BL meshes.

With an open-type advancing-layer method a boundary surface mesh is the starting point and it is inflated or advanced one or more layers at a time. Proximity and intersection checks are performed as each layer is generated. If the criteria for these checks are not met then BL advancement is terminated locally. The key to a successful open-type method is a robust proximity and intersection checking process. Typically the layers are advanced until a termination criterion is reached, such as a set normal spacing or number of layers. After all layers are advanced as far as possible, the inflated boundary surface is used to generate a tetrahedral volume mesh outside of the BL region. The primary advantage of this method is efficiency. BL regions with tens or even hundreds of millions of elements can be generated on common computers including laptops. The key weakness is potential intersections of merging layers. To deal with this robustly requires trade-offs in efficiency and/or early termination of the BL advancement. Intersection checks of varying sophistication can be used at the expense of efficiency. A compromise of limited intersections checks with a safety exclusion zone can be used to more optimally address the issue.

Alternatively, a closed-type advancing-layer method can be used which starts from an existing volume mesh. It inflates iteratively the boundary surface as each layer (or group of layers) is advanced while deforming the existing volume mesh. Volume element quality and vertex proximity checks are performed as the mesh is deformed in response to the boundary surface displacement. If the criteria for these checks are not met then BL advancement is terminated locally. After all layers are generated the result is a complete volume mesh with a BL region and a displaced tetrahedral volume mesh outside of that. A controllable quality and guaranteed validity of the volume mesh is the primary advantage of the closed methodology, assuming a robust mesh movement strategy is used. The result is an extremely robust coupled approach wherein the existence problem is solved once for the initial mesh.

The high-order mesh deformation algorithm appears to be the core component of the two previous high-order mesh generation algorithms. To this end, we have proposed a connectivity-change moving mesh methods for high-order meshes. It is based on a high-order resolution of the linear elasticity equation in which all the degrees of freedom of the problem are intrinsically represented. This gives us the motion of the vertices. Then it uses local mesh optimization operators such as mesh smoothing (for vertices and nodes) and generalized swapping coupled with an optimization of the position of the nodes inside the swap cavity. All this methodology can be first applied to the generation of a mesh starting from a mesh and then to the generation of closed-advancing high-order boundary layer meshes. This paper discusses the direct generation of curved BL mesh by extending the closed-type advancing-layer method of to high-order meshes.

The paper is outlined as follow. Section II recalls the high-order connectivity-change moving mesh method based on a high-order elasticity solver and mesh optimization operators for high-order meshes. The methodology to curve the initial mesh without boundary layer based on this high-order connectivity-change moving mesh method is discussed in Section III. Section IV presents the proposed high-order closed advancing-layer method based on a connectivity optimization based moving mesh algorithm. First results obtained on simple geometries are presented in Section V to assess the proposed strategy.

II. High-order connectivity-change moving mesh method

The generation of a mesh from an initial mesh and the closed advancing-layer method to generate BL meshes both rely on the high-order connectivity-change moving mesh method. Indeed, in both cases,
the whole mesh must be deformed when the surface mesh is curved or when the boundary layer is inserted within the domain. The $P^2$ connectivity-change moving mesh method is based on:

- a high-order resolution of the linear elasticity equation in which all the degrees of freedom of the problem are intrinsically represented
- $P^2$ mesh optimization operators

which are briefly described in the following.

A. High-order Finite Element resolution of linear elasticity equation

In this section, the resolution of linear elasticity equation with a $P^k$ Finite Element Method (FEM) is presented. Let us consider the linear elasticity equations:

$$\nabla \cdot (\sigma(\mathcal{E})) = -\mathbf{f}, \quad \text{with} \quad \mathcal{E} = \frac{\nabla \xi + \nabla \xi^T}{2},$$

(1)

where $\sigma$, and $\mathcal{E}$ are respectively the Cauchy stress and strain tensors, $\xi$ is the Lagrangian displacement and $\mathbf{f}$ are the body forces. The Cauchy stress tensor follows the Hooke’s law for isotropic homogeneous medium, where $\nu$ is the Poisson ratio, $E$ the Young modulus of the material, and $\lambda$ and $\mu$ are the Lamé coefficients:

$$\sigma(\mathcal{E}) = \lambda \text{Tr}(\mathcal{E}) I + 2\mu \mathcal{E} \quad \text{or} \quad \mathcal{E}(\sigma) = \frac{1 + \nu}{E} \sigma - \frac{\nu}{E} \text{Tr}(\sigma) I.$$

Dirichlet boundary conditions are used to enforce the displacement on the boundary. For symmetry planes or imprint surfaces, wall-slip boundary conditions are used to enforce a displacement in the tangent plane or on the curve surface. After a standard mathematical analysis using a continuous Galerkin approximation of the variational form associated to Problem (1) on a mesh $\mathcal{H}$, the Finite Element Method leads us to the following linear system:

$$\mathbb{K} \Xi = F,$$

(2)

where $\mathbb{K}$ is the linear elasticity stiffness matrix with blocks of size $d \times d$ and $F$ is a vector with blocks of size $d$, where $d$ is the dimension, and are defined (for $d = 2$) by:

$$K_{IJ} = \begin{pmatrix}
\sum_{K \ni (P_I, P_J)} \int_K (\lambda + 2\mu) \frac{\partial \phi_I}{\partial x} \frac{\partial \phi_J}{\partial x} + \mu \frac{\partial \phi_I}{\partial y} \frac{\partial \phi_J}{\partial y} \mathrm{d}\Omega \\
\sum_{K \ni (P_I, P_J)} \int_K \mu \frac{\partial \phi_J}{\partial y} \frac{\partial \phi_I}{\partial x} + \lambda \frac{\partial \phi_J}{\partial x} \frac{\partial \phi_I}{\partial y} \mathrm{d}\Omega \\
\sum_{K \ni (P_I, P_J)} \int_K \mu \frac{\partial \phi_J}{\partial x} \frac{\partial \phi_I}{\partial y} + (\lambda + 2\mu) \frac{\partial \phi_J}{\partial y} \frac{\partial \phi_I}{\partial x} \mathrm{d}\Omega
\end{pmatrix},$$

$$F_I = \begin{pmatrix}
\sum_{J \ni (P_I, P_J)} \int_K \phi_J \phi_I \mathrm{d}\Omega \\
\sum_{J \ni (P_I, P_J)} \int_K \phi_J \phi_I \mathrm{d}\Omega
\end{pmatrix}.$$

$K$ are the elements of the mesh (edges, triangles, tetrahedra) and $P_I, P_J$ are two nodes/vertices of the mesh. The solution vector $\Xi$ is defined by blocks (for $d = 2$) as:

$$\Xi = \begin{pmatrix}
\xi_{I,1} \\
\xi_{I,2}
\end{pmatrix},$$

where $\xi_{I,J}$ (resp. $f_{I,J}$) are referring to the $j$th coordinates of the evaluation of $\xi$ (resp. $\mathbf{f}$) at $P_I$. The $\phi_I$ are the $P^k$ finite element shape functions associated with the nodes/vertices $P_I$. In the case of the Lagrangian interpolation in a $d$-simplex, these functions restricted to a simplex are a polynomial combination of the $d+1$ elementary barycentric functions of this simplex. More precisely, in $P^1$, they are exactly these barycentric functions. The key feature in this method is to compute the integrals

$$\int_K \frac{\partial \phi_J}{\partial x_l} \frac{\partial \phi_I}{\partial x_m} \mathrm{d}\Omega \quad \text{and} \quad \int_K \phi_J \phi_I \mathrm{d}\Omega.$$

In $P^1$, the exact formulas are known. For $P^k$ with $k \geq 2$, there are two cases:
• The high-order element $K$ is straight. In other words, $K$ has exactly the same shape as a simplicial element. In this case, it is possible to compute these integrals without any quadrature as analytical formulas can be found.\textsuperscript{14}

• The high-order element $K$ is curved. In this case, the previous trick cannot work. The element $K$ is mapped to a straight reference element $\bar{K}$ on which a Gauss quadrature formula is used. Since a Gauss quadrature of order $n$ is exact for polynomials of degree $2n - 1$ or less, if the order of Gauss quadrature is high enough, the exact result can be found as both functions product and mapping are polynomial. Nonetheless, both computation of the jacobian of the mapping and Gauss quadrature are costly in terms of CPU.

Once the linear system is assembled, both $K$ and $F$ are modified using a pseudo-elimination technique in order to take into account Dirichlet and wall slip boundary conditions. System (2) is then solved by a Conjugate Gradient algorithm coupled with a LU-SGS pre-conditioner. An advantage of elasticity-like methods is the opportunity they offer to adapt the local material properties of the mesh, especially its stiffness, according to the distortion and efforts born by each element, see.\textsuperscript{2}

\section*{B. High-order mesh optimization}

Several optimization techniques exist to correct an invalid $P^k$ mesh\textsuperscript{23,36} and to optimize the geometrical accuracy.\textsuperscript{37} The idea here is to extend two classic mesh quality-based optimization operators\textsuperscript{2} to $P^2$ meshes to increase its quality. All the $P^k$ optimization operators are based on the following quality function:

$$Q = \alpha \frac{h S_k}{V_k} \frac{\max(V_1, V_k)}{\min(V_1, V_k)} \left( \frac{N_{\text{max}}^K}{N_{\text{min}}^K} \right)^{1/d},$$

with:

- $d$ the dimension, $S_k$ the exterior surface of the polyhedron (in 2D, the half perimeter of the polygon) defined by nodes and vertices $(A_i)_{1 \leq i \leq n}$
- $V_k$ the volume of the curved element that can be deduced from the jacobian.
- $h$ the element’s largest edge $P^k$-length (e.g., the length of the union of straight-sided lines defined by the nodes)
- $V_1$ the element’s volume/surface of the equivalent $P^1$ element, e.g., the element defined by the vertices
- $N_{\text{min}}^K$ (resp. $N_{\text{max}}^K$) the smallest (resp. largest) control coefficient of the Jacobian of the element
- $\alpha$ is a normalization factor, dependent of the dimension such that $Q = 1$ for a regular simplex, $\alpha = \sqrt[3]{3}$ in 2D and $\alpha = \sqrt[3]{36}$ in 3D.

This quality function is actually a product of 3 terms. $\textcircled{1}$ is only a generalization of the $P^1$ quality function and measures the gap to the regular element. $\textcircled{2}$ measures the distance between the volume of the curved element and the volume of the straight element and ensures the function to be greater than 1. And, finally $\textcircled{3}$ gives a measure of the distortion of the element, it can detect if the element is invalid or almost invalid by taking an infinite value. Note that if the element is straight, the standard $P^1$ quality function is recovered: $Q = Q_{P^1} = \alpha \frac{h S_k}{V_k} = \alpha \frac{h}{\rho}$ where $\rho$ is the inradius of the straight element. Based on these definitions, this element-wise quality measure is between 1 and the infinity. The closer the element quality is to 1, the better the quality is.

$P^2$ swap operator. The swap operator locally changes the connectivity of the mesh in order to improve its quality. In 2D, it consists in flipping an edge shared by two triangles to form two new triangles with the same four vertices (see Figure 1, left). In 3D, two types of swap exist: face and edge swapping. The face swapping is the extension of the 2D edge swapping, it consists in replacing the common face of two neighboring tetrahedra by the edge linking the opposite vertices to the face of each tetrahedron, also called...
2 \to 3. The edge swapping is a bit different. First, the shell of the edge to delete (e.g. the set of elements containing this edge) is constructed. From a shell of size \( n \), a non-planar pseudo-polygon formed by \( n \) vertices is obtained. The swap consists in deleting the edge, generating a triangulation of the polygon and creating two tetrahedra for each triangle of the triangulation thanks to the two extremities of the former edge. These swaps are designated as \( n \to m \) with \( n \geq 3 \), where \( n \) is the initial number of tetrahedra and \( m = 2(n - 2) \) is the final number of tetrahedra (see Figure 1, right). For each possible swapped configuration, if the worst quality of all the elements the shell is improved, the configuration is kept and will be in the new mesh unless another swapped configuration of the shell provides a better quality improvement.

To generalize it to \( P^2 \) meshes, the inner nodes of the edges of the shell have to be taken into account. For instance, for the \( P^2 \) case in 2D, there is one node on the swapped edge and if we want the swap to be performed, we need first to find an optimal position for the node in the swapped configuration and then to check if this configuration improves the quality function (see Figure 2). The key feature is therefore to find a functional whose optimum will give the optimal position for the node in the swapped configuration in term of quality. In this context, the idea is to find a simple and smooth functional that will be easy to optimize. The quality function is not a good candidate as it is not smooth. In,\(^{13}\) we have proposed such a functional. Using the result of the optimization problem in the swap configuration gives therefore a very good approximation of the best configuration that can be obtained. Since the best swap configuration is found, we are able to conclude if this swap will increase the quality or not. The resolution of these optimization problems is performed thanks to a L-BFGS algorithm.\(^{21}\)

![Figure 1. Left, the swap operation in 2D. Middle, edge swap 3 \to 2 and face swap 2 \to 3. Right, generalized swap 5 \to 6 in 3D. For all these pictures, shells are in black, old edges are in red, new edges are in green.]

**Figure 1.** Left, the swap operation in 2D. Middle, edge swap 3 \to 2 and face swap 2 \to 3. Right, generalized swap 5 \to 6 in 3D. For all these pictures, shells are in black, old edges are in red, new edges are in green.

**Figure 2.** Three steps of \( P^2 \) swap in 2D.

\( P^2 \) mesh smoothing. Mesh smoothing is a technique that consists in relocating some points inside the mesh to improve the quality of the elements. In \( P^1 \), the idea is to relocate each vertex \( P_i \) inside its ball of elements (see Figure 3). For each element \( K_j \) in the ball of \( P_i \), the opposite face to \( P_i \) denoted by \( F_j \) gives an optimal position \( P_{opt}^i \). Then, the vertex is relocated considering a weighted average of the proposed positions. If the proposed new location of the vertex does not improve the ball configuration in term of quality, then a relaxation is performed to check if an improved configuration exists between the original location and the new one. The strategy to find the optimal configuration is described in.\(^2\)

To extend it to \( P^2 \) meshes, the edges’ node needs to be taken into account. The idea here, is to perform two independent smoothing operations:
- A vertex smoothing

The vertex smoothing is simply a generalization of the $P^1$ smoothing. The optimal position of the vertex is computed in the same way as in $P^1$, and the vertex is located exactly in the same way as before. In order to be consistent with the $P^1$ vertex smoothing and to keep in the ball straight edges that are initially straight, the displacement of all the inner nodes of the ball cavity is set to half of the value of the displacement of the central vertex (see Figure 4). In the exact same way as in $P^1$ if the final configuration does not improve the quality, relaxation is performed the original location and the new one.

- A node smoothing.

The optimization of the node position follows the same algorithm as in the $P^2$ swap operator to find its optimal position. For this purpose, the functional is re-used to find the optimal node position (see Figure 5). In this case, there is always only one node coordinates to optimize and consequently the optimization problem is quadratic. In the exact same way as in $P^1$, if the final configuration does not improve the quality, relaxation is performed the original location and the new one.

![Figure 3. Laplacian smoothing in two dimensions.](image)

![Figure 4. $P^2$ laplacian smoothing in two dimensions.](image)

![Figure 5. $P^2$ node smoothing in two dimensions.](image)
III. High-order mesh generation by curving an initial Euler $P^1$ mesh

Most of the techniques to generate an high-order mesh is to start from a $P^1$ mesh and then to curve it, in a way or another, in order to obtain a $P^k$ mesh. The main reason to use a post-treatment is that all existing $P^1$ mesh generation algorithms can be reused. It would be harder to implement a directly high-order mesh generator. Our choice here is to use the linear elasticity equation as a model for the motion of the vertices to generate a $P^k$ mesh from a $P^1$ mesh but only in the context where no BL is present.

For this purpose, we solve the elasticity problem presented in Section II A. Dirichlet boundary conditions are used to represent the gap between the $P^k$-nodes of the initial straight boundary elements and their position on the real boundary. For mesh boundary vertices, the gap is equal to 0. To compute the gap at the nodes, a continuous representation of the surface mesh is required. It can be either provided by CAD/analytical model or deduced from initial $P^1$ mesh via a cubic reconstruction technique.39 Once Dirichlet boundary conditions are set, the high-order finite element linear elasticity code is called. The use of an high-order FE resolution rather than on a subdivided $P^1$ mesh aims the degrees of freedom to be intrinsically represented. This gives more consistency to the obtained motion.

The elasticity problem using the high-order FEM provides the new position of the internal vertices and nodes. It is then used to generate the high-order mesh by moving the vertices and nodes of the initial straight mesh with the associated values in the elasticity solution vector. This moving step is coupled with mesh optimization as described in Section II B. If some elements remain invalid after optimization, it is always due to non suitable boundary displacements. In this case, the FEM solution is proportionally reduced in the vicinity (boundary included) of the invalid element until global validity is obtained. The process is summarized by Algorithm 1.

Algorithm 1 Mesh curving algorithm

1. Generate a $P^1$ mesh.
2. Perform $P^1$ mesh optimization pre-processing: generalized swapping and vertex smoothing.
3. Perform cubic reconstruction of the boundary or use its analytical representation to set Dirichlet boundary conditions for the linear elasticity equation.
4. Solve linear elasticity equation on the $P^1$ mesh with the FEM at the order of the wanted mesh.
5. Generate the $P^k$ mesh by moving the $P^1$ mesh with the solution of the linear elasticity.
7. Check validity of $P^k$ elements and locally relax the FEM solution if necessary or desired until it is valid.

The major fact with this method is that the deformed mesh is only made of isotropic or almost isotropic elements. In this context, the use of the elasticity problem is efficient and always provides a valid mesh. Optimization in the pre-processing makes elements more isotropic and therefore helps curvature process to be more robust whereas optimization in the post-processing improves the quality of the mesh and untangle invalid elements if any.

We now present the result of this algorithm on the NASA Common Research Model used for the AIAA CFD Drag Prediction Workshop.35 We consider an initial coarse linear mesh composed of 32 479 vertices, 173 468 nodes and 118 012 tetrahedra, see Figure 6. The initial mesh average quality is 2.49 with a worst quality of 271 due to the presence of sharp anisotropic trailing and leading edges on the input wing surface mesh. Curving the mesh without optimization provides an invalid configuration with 10 invalid elements. Optimization in post and pre processing decrease average quality to 2.13 and worst quality to 271. Note that the number of nodes and tetrahedra is changed to respectively 173 744 and 118 288. The optimization ensures a valid curved mesh at the end that would not have been obtained without it. We can observe that the curvature is not propagated a lot in the volume as it is not visible after 2 or 3 layers.

7 of 18

American Institute of Aeronautics and Astronautics
Figure 6. NASA Common Research Model aircraft. From top to bottom, global view of the surface, zoom on the surface and view of the surface and the volume mesh. Left, initial $P^1$ mesh and, right, final $P^2$ mesh. $P^2$ meshes are generated with Algorithm 1 using the cubic reconstruction.
IV. Boundary layer mesh generation by high-order closed advancing-layer method

A closed advancing-layer boundary layer (BL) mesh generation approach is used for the present work. It is based on the existing advancing-layer method described in\textsuperscript{3,4} as it has proven to be a robust procedure that consistently produces high-quality linear BL meshes for complex configurations. This approach starts with a fully resolved volume mesh and inserts the BL mesh one or more layers at a time using a mesh deformation process. This paper describes its extension for $P^2$ meshes. It relies on the high-order connectivity-change moving mesh method presented in Section II to inflate the boundary layer (BL) inside the volume mesh.\textsuperscript{14} This moving mesh strategy has proved to be very efficient to handle large geometry displacement and shear motion inside the mesh while preserving the mesh quality.\textsuperscript{6–8} The main advantages of this strategy are the following

- it is robust because it starts from an already existing high-order mesh and it always produces a valid result
- it automatically and naturally handles BL front collision as BL is inflated inside an existing volume mesh
- it produces automatically high-quality meshes with smooth size blending between colliding BL fronts and in concave regions

Figures 7, 8 and 9 present the resulting $P^1$ mesh obtained with the closed advancing-layer BL mesh generation method on the 3rd AIAA CFD High Lift Prediction Workshop geometry. The surface mesh has been provided by Boeing and the initial volume mesh has been generated using AFLR.\textsuperscript{29} The BL parameters follow the meshing guidelines: the initial BL spacing is set to 0.00117, the 5 first layers keep a constant size, the initial growth rate is 1, the growth rate acceleration is 1.025 and the maximal growth rate is 1.2. A maximum of 54 layers have been generated leading to a final mesh composed of 25 633 441 vertices, 152 036 662 tetrahedra and 981 238 triangles. The first stops occur at layer 22 and less than one percent of the BL vertices are stopped until layer 36. In Figure 8, we observe that all the BL stops occur in the regions between the slat and the wing, between the flap and the wing and at the wing-body junction. The BL grows up to layer 54 elsewhere. This case also points out the capability of this method to maintain quality elements in BL collision regions. For example the regions between the wing and both slat and flap are shown in Figure 9. We notice a very smooth size variation between the BL mesh and the deformed volume mesh.

The following sections present the main points of the high-order BL mesh generation closed advancing-layer method.

A. Overall closed advancing-layer high-order boundary layer mesh generation

The overall process to generate high-order meshes with boundary layers follows the following stages:

1. Generate an initial $P^1$ surface mesh for the given CAD geometry
2. Generate an initial (Euler) $P^1$ volume mesh using an advancing front\textsuperscript{30} or a Delaunay-based\textsuperscript{18} mesh generation method. This mesh has no boundary layer mesh.
3. Generate a (Euler) $P^2$ surface and volume mesh using the method of Section III. This mesh has no boundary layer mesh.
4. Generate directly the $P^2$ boundary layer mesh inside the given $P^2$ Euler mesh by inflating the BL mesh inside the domain using the connectivity-change high-order moving mesh method presented in Section II.

The last step of the overall process is detailed in the following discussion.
Figure 7. 3rd AIAA CFD High Lift Prediction Workshop geometry.

Figure 8. 3rd AIAA CFD High Lift Prediction Workshop final BL outer skin mesh. Left, view of the top of the wing and, right, view of the bottom of the wing.

Figure 9. 3rd AIAA CFD High Lift Prediction Workshop final BL mesh. Top left, a cut plane through the volume showing the BL mesh around the multi-elements wing. Top right, a close-up view in the gap between the slat and the wing. Bottom left, a close-up view in the gap between the flap and the wing. Bottom right, a close-up view on the wing in the region where the slat ends.
B. High-order closed advancing-layer method

The high-order advancing-layer method is an extension of the existing linear method.\textsuperscript{3,4} This algorithm is based on that described in.\textsuperscript{29} In it, each BL vertex and BL node (high-order case only) is advanced one layer at a time. The primary steps in the advancing-normal/layer procedure are presented in Algorithm 2.

\begin{algorithm}
\caption{Closed advancing-layer algorithm}
\begin{enumerate}
\item Determine a normal vector at each active BL point from the geometry of the inflated boundary surface. For high-order elements this includes the element-edge and element-face nodes along with the vertices.
\item Generate proposed BL points one layer at a time. New points are created along the normal vector with the normal spacing determined using geometric growth from the boundary surface.
\item Proposed points are rejected if any of the elements that include it are invalid or fail a quality check.
\item Check element aspect ratio. As the mesh advances and the normal spacing increases the element aspect ratio will eventually be isotropic. Boundary-layer advancement is terminated locally when the aspect ratio on the next layer would be greater than a preset factor ($\leq 1$).
\item Send displacement vectors for the proposed BL points to the outer region mesh deformation process (see Algorithm 3). Note that this algorithm may reject one or more of the proposed BL points.
\item Boundary-layer advancement is terminated locally if the proposed BL point was rejected by any of the preceding steps.
\item If the proposed BL points are accepted then the BL interface is advanced.
\item Continue if there are still active BL points and the number of layers generated does not exceed a preset maximum.
\end{enumerate}
\end{algorithm}

With $P^2$ elements the treatment of vertices and element-edge (or mid-side) points require special procedures. Two steps are required to advance one layer of elements. First the mid-side points that are advanced at a half-step in the normal direction must be inserted and then the vertices and mid-side points that are advanced a full-step. There is also the issue of whether to create the element edge parallel to the BL normal vector as a straight or curved edge. We use a straight edge as it provides better quality. Also, the rejection of any vertex or element-edge point implies the rejection of all associated element-edge points and elements. This includes rejection of BL points advanced a full-step forcing the rejection of previously accepted BL points advanced only a half-step. The impact of this issue and need for two steps can be minimized by using only a full-step advancement and then optimizing the location of the half-step points. In this case, the rejection of any full-step BL points automatically rejects the associated half-step ones. And, if the location of the half-step points cannot be optimized, then they force rejection of associated full-step points. With $P^3$ and higher order elements, additional procedure will be required.

C. BL high-order mesh deformation method

The preceding advancing-layer BL method provides a displacement vector for the mesh deformation process at each vertex and node on the BL interface surface. Given this displacement vector the volume mesh is deformed using the high-order connectivity change moving mesh algorithm to insert the layer. The overall algorithm is presented in Algorithm 3. More details can be found in\textsuperscript{3,4} for the linear case. The cost of the mesh deformation can be reduced seeing that it is not efficient to solve the mesh deformation at each layer insertion. Especially, at the beginning when the boundary layer inflation is quite small. In our approach, the mesh deformation is solved each 10 layers or when the sum of the layer normal sizes, since the last mesh deformation solution, is larger than two times of the surface mean size. If the mesh deformation is avoided, then inner vertices displacements are updated using the boundary layer growth rate:

$$d_{i+1} = \beta_{i+1} d_i$$

where $\beta_{i+1}$ is the growth rate at the $i$th layer.

A good restriction to be imposed on the mesh movement to limit the apparition of flat or inverted elements is that vertices and nodes cannot cross too many elements on a single move between two mesh
Algorithm 3 High-order closed advancing-layer BL mesh generation

For $i_{lay} = 1, \ldots, n_{lay}$

1. Create layer $i_{lay}$: For each active point propose its optimal position using the advancing-layer method given in Algorithm 2

2. If (mesh deformation criteria) Then
   - $d|_{\partial \Omega_h} =$ Get boundary vertex and node displacement from inflating boundary layer
   - $d =$ Get inner vertex and node displacement by solving the elasticity system ($d|_{\partial \Omega_h}$)
   Else
   - $d = \beta d$ increment vertex and node displacement by the growth rate
   EndIf

3. Set $t = 0, T = 1$ and vertex speed $v = d$

4. While ($t < T$)
   - (a) $\delta t =$ Get moving mesh time step ($H^k, v, CFL_{geom}$)
   - (b) $H^k =$ Connectivity optimization ($H^k, Q_{swap}$)
   - (c) $v^{opt} =$ Vertex smoothing ($H^k, Q_{target}^{smoothing}, Q_{max}$)
   - (d) $v^{opt} =$ Node smoothing ($H^k, Q_{target}^{smoothing}, Q_{max}$)
   - (e) $H^{k+1} =$ Move the mesh ($H^k, \delta t, v, v^{opt}$)
   - (f) Check mesh validity:
     - If (element quality threshold is exceeded) Then
       - Cancel element’s vertices and nodes displacements
       - Freeze element’s vertices and nodes : $v = 0$
     EndIf
   - (g) $t = t + \delta t$
   EndWhile

5. Move back vertices and nodes that have moved less than a threshold percentage of the layer size

EndFor

optimizations. Therefore, a geometric parameter $CFL_{geom}$ is introduced to control the number of stages used to perform the mesh displacement. If $CFL_{geom}$ is greater than one, the mesh is authorized to cross more than one element in a single move. In practice, $CFL_{geom}$ is usually set to 1. The moving geometric time step is given by:

$$\delta t = CFL_{geom} \max_{P_i} \frac{h(x_i)}{v(x_i)},$$

where $h(x_i)$ is the smallest height of all the elements in the ball of vertex $P_i$ and $v(x_i)$ is the velocity of vertex $P_i$.

In the case of imprint surfaces, moving vertices and nodes are moved and smoothed in the tangent plane, then re-projected on the surface after each moving step.

D. Managing the boundary layer progression

Checking mesh validity. At the end of each moving step, we check if the mesh is valid and meets all quality requirements. The validity requires all mesh elements to have a positive volume after the mesh deformation. Quality requirements are thresholds that are set depending on the initial mesh or actual mesh quality. They govern when the BL region will stop inflating: lower bound stops the BL sooner while larger
bound allows the BL to propagate farther. For isotropic unstructured meshes, quality functions cannot
directly be used to stop the BL progression because the BL inflation may lead to anisotropic elements.
These anisotropic elements are considered bad for the quality function but, in fact, they are good for the
transition from the BL to the outer region. To solve this issue, quality criteria check for an element are only
activated if the height of the element is below one fourth of the current layer normal size. In that case, we
compare its height \( h \) and its quality \( Q \) of with respect to its initial height \( h_{ini} \) and initial quality \( Q_{ini} \) (i.e.,
evaluated before inflating the BL) weighted by a prescribed coefficient, both quality criteria are violated if:

\[
\frac{h}{h_{ini}} \leq \alpha \quad \text{and} \quad \frac{Q}{Q_{ini}} \geq \beta
\]

In this work, we use \( \alpha = 0.25 \) and \( \beta = 3 \). This criteria ensures a smooth transition between colliding layers.

Checking boundary layer validity. As some vertices may be frozen at each moving step, some faces
may have only a part of its three vertices continuing growing. In some tricky situation, this may lead to an
invalid element in the boundary layer which has not been previously detected by the closed advancing-layer
algorithm. Indeed, the decomposition of the prism into tetrahedra changed. Thus, if a vertex or a node of
a BL face has been frozen, then the BL elements validity should be checked. If an invalid element is going
to be created in the BL, then all the vertices and nodes of that face are marked frozen.

Finalizing the BL position. At the end of the mesh movement, the final position at the current layer
of each BL vertex and node is compared with its targeted final position. Vertices and nodes are categorized
in two groups:

- If the total displacement is greater than 60% of the expected displacement: this BL vertex or node
  remains active for progressing.
- If the total displacement is less or equal to 60% of the expected displacement: this BL vertex or node
  advancement is terminated and we seek for an optimal final position.

The threshold 60% has been chosen because we limit the maximum number of moving step for each layer to
five, and it is appropriate in that case.

If a BL vertex or node advancement is terminated, then we search for an optimal final position. First,
we try to move it back to its position at the end of the previous layer, i.e., its initial position at the current
layer. Most of the time, this move back is possible. However, again in complicated situations, moving back
the vertex or the node may significantly degrade the mesh quality or possibly create invalid elements. In
that case, we analyze the mesh configuration when the vertex or the node is moved by 0%, 40% and 60%
of its expected displacement, and the best one is selected. In \( P^2 \), the position of the mid-layer nodes is
adjusted accordingly to the shrink of the expected displacement. Also, when a high-order boundary node
has an expected displacement which is reduced, the displacement of the neighboring vertices is reduced as
well so that it does not distort that much the inflation of the boundary mesh.
V. Numerical Examples

The first results obtained on simple geometries are presented to assess the proposed strategy.

A. Spheres

The first example is a domain represented by two spheres, the inner sphere is of radius 1 and the outer one of radius 3, see Figure 10 (left). The initial domain is composed of 12,517 vertices and nodes, 1,290 $P^2$-triangles and 8,394 $P^2$-tetrahedra. We are growing BL meshes on each sphere to check the behavior of the moving mesh method when two BL meshes are colliding. The parameters are: the initial BL spacing is set to 0.0001, the growth rate is 1.2, and 21 layers are generated leading to a final mesh composed of 123,629 vertices and nodes, 89,634 $P^2$-tetrahedra and 1,290 $P^2$-triangles. This viscous $P^2$-mesh is displayed in Figure 10 (right). Close-up views of the boundary layer are shown in Figure 11.

We observe that the proposed methodology is able to manage nicely the BL mesh collision with a nice transition between the BL mesh region and the outer mesh region. The curvature of the domain is properly accounted in the BL mesh and the elements of the initial mesh are curved to fit the shape of the BL.

Figure 10. Spheres case. Left, initial $P^2$-mesh and, right, final $P^2$-mesh after insertion of the boundary layer.

Figure 11. Spheres case. Close-up views in the boundary layer region.
B. Cylinder

The second example is a cylinder of height 2 and radius 0.2 included in a spherical domain of radius 10, see Figure 12 (left). The initial domain is composed of 113,695 vertices and nodes, 9,598 $P^2$-triangles and 77,894 $P^2$-tetrahedra. The parameters are: the initial BL spacing is set to 0.0005, the growth rate is 1.12, and 20 layers are generated leading to a final mesh composed of 900,895 vertices and nodes, 653,452 $P^2$-tetrahedra and 9,598 $P^2$-triangles. This viscous $P^2$-mesh is displayed in Figure 12 (right). Close-up views of the boundary layer are shown in Figure 13.

This simple test case is interesting because it has a curved part and a straight part. We note that on the straight part, the BL mesher behaves like in the $P^1$-case which is consistent. It also means that no artificial curvature is created. In the curved part, the curvature is transferred properly through the layers. The elements on the last layer seem to be straight but they are a little bit curved, see Figure 13 (right). We also observe that the ridge of the cylinder is correctly managed by the process, so does the normal smoothing, see Figure 13 (left).

Figure 12. Cylinder case. Left, initial $P^2$-mesh and, right, final $P^2$-mesh after insertion of the boundary layer.

Figure 13. Cylinder case. Close-up views in the boundary layer region.
C. Rocket

The third example is a little bit more complex. It is a notional rocket model. This geometry is interesting because it has at the same time spherical like curvature on the nose, cylindrical like curvature on the body, multiple circular convex and concave ridges, and a sharp circular trailing edge, see Figure 14. The rocket length is 7 and it has a maximal diameter of 2.5. The initial domain is composed of 227 258 vertices and nodes, 10 850 $P^2$-triangles and 162 275 $P^2$-tetrahedra. The parameters are: the initial BL spacing is set to

Figure 14. Rocket case. Left, rocket geometry and, right, initial $P^2$ surface mesh.

Figure 15. Rocket case. Left, rocket geometry and, right, initial $P^2$ surface mesh.
0.00004, the growth rate is 1.15, and 20 layers are generated leading to a final mesh composed of 1075018 vertices and nodes, 781,632 $P^2$-tetrahedra and 10,850 $P^2$-triangles.

Figure 15 displays close-up views of the boundary layer mesh at geometric singularities. The top left picture shows the management of the circular trailing edges, and the other pictures the management of circular convex and concave ridges. We note again that the whole algorithm is behaving properly on these geometric singularities.

VI. Conclusion

A closed advancing-layer method for generating $P^2$ boundary layer meshes has been presented. This method is a combination of a high-order connectivity-change moving mesh method that works on the isotropic part of the mesh and a high-order closed advancing layer algorithm which proposes the degrees of freedom for the boundary layer mesh. The obtained results are promising and take into account high-order features.

Future work will consider more realistic geometries. Some perspectives can be driven regarding this work. First, the backtracking algorithm is at the time made in a $P^1$ sense, e.g. the nodes are moving back according to the straight edge but not according the curved edge (note the most of the time the considered are straight). This should help to keep the validity in corner areas for instance. Then, the mesh deformation algorithm is based on a linear elasticity analogy and it does work fine for linear meshes. However, it has some drawbacks with higher-order meshes especially for the motion of the high-order nodes that curves high-order elements where it is not necessary. To cope with that, some alternatives are possible: we can decide to perform a $P^1$ linear elasticity motion end then perform $P^2$ node smoothing to enforce the curvature where needed. It could be also possible to change the deformation model and use some models like IDW. Finally, the whole process could be applied to generated hybrid boundary layer meshes including prisms, pyramids or hexahedra. To this end, the high-order validity of such elements is already established.

VII. Acknowledgment

This work was supported by a public grant as part of the Investissement d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

References


