A Closed Advancing-Layer Method with Changing Topology Mesh Movement for Viscous Mesh Generation

F. Alauzet[1] and D. Marcum[2]

[1] Gamma3 Team, INRIA Paris-Roquencourt, Le Chesnay, France
Frederic.Alauzet@inria.fr, alauzet@cavs.msstate.edu
[2] Center for Advanced Vehicular Systems and Dept. of Mechanical Engineering, Mississippi State University, USA
marcum@cavs.msstate.edu

Summary. A new closed advancing-layer method for generating high-aspect-ratio elements in the boundary-layer (BL) region is presented. This approach utilizes a recent changing topology moving mesh strategy for deforming the volume mesh as the BL is inflated. It handles very efficiently BL front collision and produces a natural smooth anisotropic blending between colliding layers. Moreover, it provides a robust strategy to couple unstructured anisotropic mesh adaptation and high-aspect-ratio elements pseudo-structured BL meshes. The proposed method is directly compared to a well established open advancing-layer method. Results for typical aerospace configurations are presented that provide a clear comparison between both methods as well as the effectiveness of the changing topology moving mesh strategy. They show that the closed method yields similar results in terms of mesh quality and efficiency, and that the considered moving mesh strategy is an efficient and effective method for deforming the unstructured volume mesh.

Keywords: Unstructured mesh generation, advancing-layer, boundary-layer mesh, viscous grid, moving mesh.

1 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 that usually includes turbulence.

J. Sarrate & M. Staten (eds.), *Proceedings of the 22nd International Meshing Roundtable*,
DOI: 10.1007/978-3-319-02335-9_14, © Springer International Publishing Switzerland 2013

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.

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 [3, 5, 8, 10, 11, 12, 15, 16, 17, 22, 23]. 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 [3, 8, 12, 15, 16, 17, 22, 23] 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. Anisotropic solution adaptation may be performed as part of the volume mesh generation process or in a separate post-processing step. However, as the mesh has to be re-generated from the anisotropic inflated surface, each time we face the mesh existence problem (boundary recovery for Delaunay-type method or mesh closure for Advancing Front) which has robustness issues [1]. 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 [5, 10, 11, 15, 16, 20] can be used which starts from an existing volume mesh. Two main approaches exist.

The first one is local and consists in inserting the BL mesh one vertex after another using a mesh generation scheme with iterative point placement [15, 16]. In the BL region an advancing-normal point placement is used that mimics the advancing-layer approach. Outside of the BL region an isotropic type point placement is used, such as advancing-front, edge bi-section, centroid, etc. A valid closed volume mesh is maintained and standard checks for addition of new points in the mesh are used. This approach works quite well in two dimensions. However, it is not cost effective in three dimensions and sliver elements are introduced in the BL region.

The more common closed approach is global and typically involves iteratively inflating 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 and guaranteed validity of the volume mesh is the primary advantage of the closed methodology. Anisotropic solution adaptation may be performed either in a separate step before BL generation as part of the initial volume mesh or after as part of a post-processing step using a local remeshing strategy [6, 14]. The capability to start with a solution-adapted volume mesh is a significant advantage that effectively eliminates the need for a separate process to adapt within a pseudo-structured BL region. The result is an extremely robust coupled approach wherein the existence problem is solved once for the initial mesh.

The key to a successful closed-type advancing-layer method is an effective, efficient, and robust mesh deformation algorithm. In this work, a new closed advancing-layer method is proposed that relies on a recent changing topology moving mesh algorithm [2]. Mesh deformation is computed by means of a PDE-based method using a linear-elasticity analogy [2, 4, 24]. To drastically improve the robustness, the efficiency and the resulting mesh quality, it is combined with a mesh optimization phase using vertex smoothing and local reconnection. Local reconnection allows us to solve efficiently any shear movements that occur inside the mesh [2]. Numerical results also demonstrate that, with this new approach, merging BL mesh regions are handled naturally and layers can be allowed to merge very closely with a smooth size transition in all directions.

The paper is outlined as follow. Section 2 presents a state-of-the-art open advancing-layer approach. Section 3 describes the proposed closed advancing-layer method based on a recent topology change moving mesh algorithm. Results for typical aerospace configurations are presented in Section 4 that provide a clear comparison of the open and closed advancing-layer methods as well as the effectiveness of our moving mesh strategy.

2 Open Advancing-Layer Method

The overall volume meshing process used in the present work is based on the Advancing-Front and Local Reconnection (AFLR) algorithm [18, 21]. It has proven to be capable of generating high-quality unstructured volume meshes suitable for large-scale high-resolution CFD applications and is widely used in aerospace and other engineering disciplines. It is a combination of automatic point creation, advancing type ideal point placement, and connectivity optimization schemes. A valid grid is maintained throughout the grid generation process. This provides a framework for implementing efficient local

search operations using a simple data structure. It also provides a means for smoothly distributing the desired point spacing in the field using a point distribution function. This function is propagated through the field by interpolation from the boundary point spacing or by specified growth normal to the boundaries. Points are generated using either advancing-front type point placement for isotropic elements, advancing-point type point placement for isotropic right angle elements, or advancing-normal type point placement for high-aspect-ratio elements. The connectivity for new points is initially obtained by direct subdivision of the elements that contain them. Connectivity is then optimized by local-reconnection with a combined Delaunay/min-max type (minimize the maximum angle, maximize the edge weight, etc.) type criterion. The overall procedure is applied repetitively until a complete field grid is obtained.

Within the BL region a high-aspect-ratio element pseudo-structured mesh is generated that is aligned with the boundary surface and is optimal for accuracy and performance of typical CFD solvers. While the standard unstructured meshing procedure previously described can be utilized to generate pseudo-structured elements in the BL region, an open (extrusion) or closed (displacement) approach is far more efficient. A modified procedure using an open type advancing-normal/layer approach is used for volume mesh generation with AFLR. In this approach, the element connectivity is generated along with new points in high-aspect-ratio regions. Local-reconnection is not used to determine the connectivity in these regions. Instead, the connectivity is directly determined as each new point is generated. This produces a very structured connectivity and allows the tetrahedral elements to be easily combined into structured type elements. Typically, the majority of the tetrahedral elements within the high-aspect-ratio region can be combined into prismatic elements. If the surface mesh contains quad faces then hexahedral elements can be formed. For comparison with the closed advancing-layer method only tetrahedral elements are generated in the BL region. The basic algorithm for the open advancing-normal/layer procedure is presented in Algorithm 1.

3 Overall Closed Advancing-Layer Method

This section presents a new closed advancing-layer method based on a topology change moving mesh algorithm. Closed-type advancing-layer methods start from an existing volume mesh that is typically fully resolved and push or insert the boundary layer mesh into the existing mesh. Once an initial mesh has been generated, many boundary layer meshes can be generated with different characteristics without re-generating the initial mesh. Therefore, the problem of the existence of a mesh [9, 13, 21] is solved only once (at the beginning). This is a main asset for coupling unstructured anisotropic mesh adaptation and high-aspect-ratio elements pseudo-structured BL meshes. Indeed, the problem of the existence of a mesh was one of the blocking point to

Algorithm 1. Open advancing-layer algorithm

While *active BL points* > 0

1. Determine a normal vector at each active BL point from the geometry of the inflated boundary surface.
2. Smooth the normal vectors with a weighting dependent upon the distance from the boundary. Initially the normals are not smoothed. At the estimated end of the boundary layer region full smoothing is applied.
3. Generate new points one layer at a time. For efficiency multiple layers can be generated in one step. This can be effective early in the BL mesh generation process when the normal spacing is very small in comparison to the surface tangential spacing. New points are created along the normal vector with the normal spacing determined using geometric growth from the boundary surface.
4. Check distance between new points and surrounding elements and points. A volume triangulation or oct-tree structure is used to efficiently check nearby points. As boundary layers merge new points may be too close and advancement should terminate locally. A new point is rejected if the distance between it and any nearby new (or existing) point is less than a preset fraction of the local element length scale. Boundary-layer advancement is terminated locally if a new point is rejected.
5. New points are also rejected if any of the surrounding elements, that they may produce, fail a quality check (e.g., maximum angle $> 160°$). Boundary-layer advancement is terminated locally if a new point is rejected for quality.
6. Check element aspect ratio. As the grid 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.
7. Inflate the previous surface mesh at points that have continued to advance.

EndWhile

achieve highly anisotropic adapted unstructured meshes for real-life CFD application [1]. This robustness issue has been solved devising local remeshing methods where the validity of the mesh is always guaranteed [6, 14].

However, the efficiency and effectiveness of such closed advancing-layer method relies on the considered mesh deformation. Here, a recent topology change moving mesh strategy is considered. It has proved to be very efficient to handle large geometry displacement and shear motion inside the mesh [2] while preserving the mesh quality. We will see that it provides a natural blending when boundary of layers collide. The overall closed advancing layer algorithm is presented in Algorithm 2.

3.1 Closed Advancing-Layer Method

The closed advancing-layer method follows the open type advancing-normal/layer approach previously presented. A key difference is that no checking

Algorithm 2. Overall closed advancing-layer algorithm

For $i_{lay} = 1, .., n_{lay}$

1. Create layer i_{lay}: for each active point propose is optimal position. No check. Advancing-layer method is given in Algorithm 3.
2. If (mesh deformation criteria) Then
   $\mathbf{d}_{|\partial\Omega_h}$ = Get boundary vertex displacement from inflating boundary layer
   $\mathbf{d}$ = Get vertex displacement by solving the elasticity system $\left(\mathbf{d}_{|\partial\Omega_h}\right)$
   Else
   $\mathbf{d} = \beta\ \mathbf{d}$ increment vertex displacement by the growth rate
   EndIf
3. Set $t = 0$, $T = 1$ and vertex speed $\mathbf{v} = \mathbf{d}$
4. While $(t < T)$
   a. δt = Get moving mesh time step $\left(\mathcal{H}^k, \mathbf{v}, CFL^{geom}\right)$
   b. $\mathcal{H}^k$ = Connectivity optimization $\left(\mathcal{H}^k, Q^{swap}_{target}\right)$
   c. $\mathbf{v}^{opt}$ = Vertex smoothing $\left(\mathcal{H}^k, Q^{smoothing}_{target}, Q_{max}\right)$
   d. $\mathcal{H}^{k+1}$ = Move the mesh $\left(\mathcal{H}^k, \delta t, \mathbf{v}, \mathbf{v}^{opt}\right)$
   e. Check mesh validity:
      If (element quality threshold is exceeded) Then
         Cancel element's vertices displacements
         Freeze element's vertices : $\mathbf{v} = 0$
      EndIf
   f. $t = t + \delta t$

   EndWhile

EndFor

between nearby points and elements is done since mesh deformation is being used to inflate the boundary-surface. Collisions between merging layers are naturally prevented by the linear elasticity model. The basic algorithm for the closed advancing-normal/layer procedure is presented in Algorithm 3. This algorithm is simply the open advancing-normal/layer Algorithm 1 with omissions. The open method is implemented within a well established unstructured mesh generator that is routinely applied to industrially relevant CFD problems. The closed advancing-layer approach is implemented in recently developed software. While it follows an advancing-layer approach that is very much like that in the more mature open method implementation, it does not include many of the features such as boundary normal vector smoothing that accounts for topology, geometry and discontinuities. Termination criteria also differ slightly and are not as sophisticated as in the open method implementation. These differences are simply due to maturity of the software.

Algorithm 3. Closed advancing-layer algorithm

1. Determine a normal vector at each active BL point from the geometry of the inflated boundary surface.
2. Generate new 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.
3. New points are rejected if any of the surrounding elements, that they may produce, fail a quality check (e.g., maximum angle $> 160°$). Boundary-layer advancement is terminated locally if a new point is rejected for quality.
4. Check element aspect ratio. As the grid 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.
5. Inflate the previous surface mesh at points that have continued to advance.

3.2 Linear Elasticity Mesh Deformation Method

For the closed advancing-layer method, the whole mesh must be deformed when the boundary layer is pushed inside the domain. The displacement of all vertices, *i.e.,* the mesh deformation, is computed by means of a PDE-based method using a linear-elasticity analogy [2, 4, 24]. More precisely, the inner vertices movement is obtained by solving an *elasticity-like equation* with a $\mathbb{P}_1$ Finite Element Method (FEM):

$$\mathrm{div}(\sigma(\mathcal{E})) = 0\,, \quad \text{with} \quad \mathcal{E} = \frac{\nabla \mathbf{d} + {}^T\nabla \mathbf{d}}{2}\,, \tag{1}$$

where σ and $\mathcal{E}$ are respectively the Cauchy stress and strain tensors, and $\mathbf{d}$ is the Lagrangian displacement of the vertices. The Cauchy stress tensor follows the Hooke's law for isotropic homogeneous medium. Dirichlet boundary conditions are used and the displacement of vertices located on the BL outer layer and the domain boundary is strongly enforced in the linear system. The FEM system is solved by a Conjugate Gradient algorithm coupled with an LU-SGS pre-conditioner. Another 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 [2].

The cost of the mesh deformation resolutions 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 one fourth of the surface mean size. If the mesh deformation is avoided, then current vertices displacements are updated by the boundary layer growth rate:

$$\mathbf{d}^{ilay+1} = \beta\, \mathbf{d}^{ilay} \quad \text{where } \beta \text{ is the growth rate.}$$

3.3 Local Mesh Optimization

In [2], it has been proposed to couple mesh deformation with local mesh optimization using smoothing and generalized swapping to achieve efficiently large displacement in moving mesh applications. Topology changes are really effective in handling shear and removing highly skewed elements. Here, we briefly recall the mesh optimization procedure in the generalized context of metric-based mesh adaptive optimization.

For 3D adapted meshes, an element's quality is measured in terms of element's shape by the following quality function [7]:

$$Q_{\mathcal{M}}(K) = \frac{\sqrt{3}}{216} \frac{\left(\sum_{i=1}^{6} \ell^2_{\mathcal{M}}(\mathbf{e}_i) \right)^{\frac{3}{2}}}{|K|_{\mathcal{M}}} \quad \in [1, +\infty] \ , \tag{2}$$

where $\ell_{\mathcal{M}}(\mathbf{e})$ and $|K|_{\mathcal{M}}$ are edge length and element volume in metric $\mathcal{M}$. Metric $\mathcal{M}$ is a 3×3 symmetric positive definite tensor prescribing element sizes, anisotropy and orientations to the mesh generator [1, 7]. $Q_{\mathcal{M}}(K) = 1$ corresponds to a perfectly regular element while a high value of $Q_{\mathcal{M}}(K)$ indicates a nearly degenerated element. For non-adapted meshes, the identity matrix $\mathcal{I}_3$ is chosen as metric tensor.

The first mesh optimization tool is vertex smoothing which consists of relocating each vertex inside its ball of elements, *i.e.,* the set of elements having P_i as vertex. For each tetrahedron K_j of the ball of P_i, a new optimal position P_j^{opt} for P_i can be proposed to form a regular tetrahedron in metric space:

$$P_j^{opt} = G_j + \sqrt{\frac{2}{3}} \frac{\mathbf{n}_j}{\ell_{\mathcal{M}}(\mathbf{n}_j)} \ ,$$

where F_j is the face of K_j opposite to vertex P_i, G_j is the gravity center of F_j, $\mathbf{n}_j$ is the inward normal to F_j and $\ell_{\mathcal{M}}(\mathbf{n}_j)$ the length of $\mathbf{n}_j$ in metric $\mathcal{M}$. The final optimal position P_i^{opt} is computed as a weighted average of all these optimal positions $\{P_j^{opt}\}_{K_j \supset P_i}$, the weight coefficients being the quality of K_j. In this way, an element of the ball is all the more dominant if its quality in the original mesh is bad. Finally, the new position is analyzed: if it improves the worst quality of the ball, the vertex is directly moved to its new position. Otherwise, successive relaxed positions with progressively decreasing values of α are checked.

The second mesh optimization tool to improve mesh quality is generalized swapping/local-reconnection. Let α and β be the two tetrahedra's vertices opposite to the common face $P_1P_2P_3$. Face swapping consists of suppressing this face and creating the edge $\mathbf{e} = \alpha\beta$. In this case, the two original tetrahedra are deleted and three new tetrahedra are created. This swap is called $2 \rightarrow 3$. The reverse operator can also be defined which consists of

deleting three tetrahedra sharing such a common edge $\alpha\beta$ and creating two new tetrahedra sharing face $P_1P_2P_3$. This swap is called $3 \rightarrow 2$.

A generalization of this operation exists and acts on shells of tetrahedra [2, 7]. For an internal edge $\mathbf{e} = \alpha\beta$, the shell of $\mathbf{e}$ is the set of tetrahedra having $\mathbf{e}$ as common edge. From a shell of size n, a non-planar *pseudo-polygon* formed by n vertices $P_1...P_n$ can be defined. Performing a three-dimensional swap of edge $\alpha\beta$ requires (i) suppressing edge $\alpha\beta$ and all tetrahedra of the shell, (ii) defining a triangulation of the pseudo-polygon $P_1...P_n$ and (iii) finally creating new tetrahedra by joining each triangle of the pseudo-polygon with vertices α and β. The number of possible triangulations depends on n the number of vertices of the pseudo-polygon. The different edge swaps are generally denoted $n \rightarrow m$ where m is the number of new tetrahedra. In this work, edge swaps $3 \rightarrow 2$, $4 \rightarrow 4$, $5 \rightarrow 6$, $6 \rightarrow 8$ and $7 \rightarrow 10$ have been implemented.

3.4 Moving Mesh Algorithm

For each layer, the mesh motion is split into several stages. This is the while loop in Algorithm 2. Each stage is composed of three main steps:

1. the geometric time step is computed with respect to the mesh characteristics
2. mesh optimization is applied
3. and finally, the mesh is moved and its validity is analyzed. Depending on the validity result, some BL vertices may be frozen or remain active for progressing.

Geometric Time Step. A good restriction to be imposed on the mesh movement coupled with mesh optimization is that vertices cannot cross too many elements on a single move. Indeed, increasing the number of stages to achieve the prescribed displacement makes the mesh motion easier as more mesh optimization are performed. A geometric parameter CFL^{geom} is then introduced to control the number of stages used to perform the mesh displacement between t and $t + \Delta t$. If CFL^{geom} is greater than one, the mesh is authorized to cross more than one element in a single move. The moving geometric time step is given by:

$$\delta t = CFL^{geom} \max_{P_i} \frac{h(\mathbf{x}_i)}{\mathbf{v}(\mathbf{x}_i)}, \tag{3}$$

where $h(\mathbf{x}_i)$ is the smallest altitude of all the elements in the ball of vertex P_i. Therefore, at each stage, the effective displacement of each vertex from t to $t + \delta t$ is given by:

$$\mathbf{x}_i(t + \delta t) = \mathbf{x}_i(t) + \delta t \left(\mathbf{v}(t) + \mathbf{v}^{opt}\right),$$

where $\mathbf{v}^{opt}$ is a correction of the vertex speed due to the smoothing optimization.

Mesh Validity and Managing the BL. At the end of each moving stage, we check if the mesh is valid and meets all quality requirements. The validity requires all mesh elements to have a positive volume. Quality requirements are threshold that are set depending on the initial mesh or actual mesh quality. They govern when 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 be used to stop the BL progression because the BL may lead to anisotropic elements. These anisotropic elements are considered bad for the quality function but, in fact, they good for the transition from the BL to the outer region. The same remark applied for adapted anisotropic meshes as the metric (anisotropy) of the initial mesh may be different from the metric (anisotropy) requested by the BL transition. In this work, the chosen quality requirement is that the height of deformed elements must exceed one fourth of the current BL normal size. This criteria ensures a smooth transition between colliding layers.

If one of these requirement is not reached, then displacements of the analyzed element vertices are cancelled, *i.e.,* the element vertices move back to their previous position at the beginning of this moving stage. This canceling action changes the neighboring elements quality, thus this procedure is iterated until no more vertices are cancelled. At worst, the displacement of all the vertices is cancelled and the mesh come back at the previous stage position where it was valid. Now, for all vertices that have been cancelled, their displacements are frozen, *i.e.,* $\mathbf{v} = 0$, until the next layer moving mesh step.

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

- If the total displacement is greater than 70% of the expected displacement: this BL vertex remains active for progressing
- If the total displacement is less than 30% of the expected displacement: the final position of this vertex becomes the final position of the previous layer and its advancement is terminated at the previous layer
- If the total displacement is in between then this vertex position is accepted and its advancement is terminated at the current layer.

4 Numerical Examples

Examples are presented in this section to illustrate the efficiency and effectiveness of the closed advancing-layer method based on the changing topology moving mesh algorithm. It is compared with the open advancing-layer method implemented within the AFLR3 code. This is a well established unstructured mesh generator that is routinely applied to industrially relevant

CFD problems. The closed advancing-layer approach is implemented in recently developed software. While it follows an advancing-layer approach that is very much like that in the more mature open method implementation, it does not include many of the features such as boundary normal vector smoothing that accounts for topology, geometry and discontinuities. Termination criteria also differ slightly and are not as sophisticated as in the open method implementation. Some differences between the open and closed advancing-layer methods are expected simply due the current state of the implementation. Particularly in how the boundary normals vary as the BL mesh is advanced. All of the BL generation features of the open method can be implemented in the closed advancing-layer procedure. With the closed method the starting isotropic unstructured mesh was generated using the same process as with the open method (AFLR3) in all but the last case presented for as equal a comparison as possible. Alternatively, most any tetrahedral unstructured mesh generator could be used for this step [9, 13].

For the closed method, the CFL^{geom} parameter has been set equal to 4 for all the examples. This is a reasonable compromise between the number of mesh optimizations and the quality of the resulting mesh. Increasing this value will lead to slightly lower mesh quality while moderately reducing the CPU time and vice versa. The robustness of the algorithm is not impacted because a mesh optimization is performed at each layer inflation. Also, the closed advancing-layer method uses an initial volume mesh that is equivalent to the outer field mesh generated for the open method (they are generated by the same process).

All of these examples have been run in *serial* on one core of a 64-bit MacPro with a 2.8 GHz IntelCore2 chipset and 32 Gb of RAM.

4.1 Nacelle

The first test case is a nacelle configuration with high-quality isotropic unstructured triangular faces on the surface. This case was chosen as it has

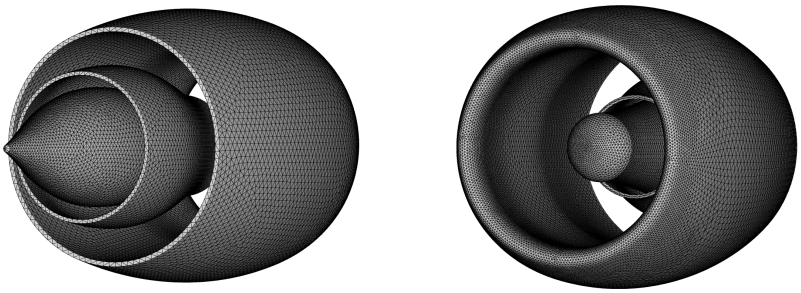

Fig. 1 Nacelle unstructured tri faces surface mesh

Table 1 Nacelle test case. Mesh statistics and CPU time for each method.

Method	# layers	# vertices	# tets	# BL tets	CPU time
Open	46	1 636 282	9 700 808	8 823 652	50s
Closed	46	1 727 014	10 242 430	9 012 741	83s

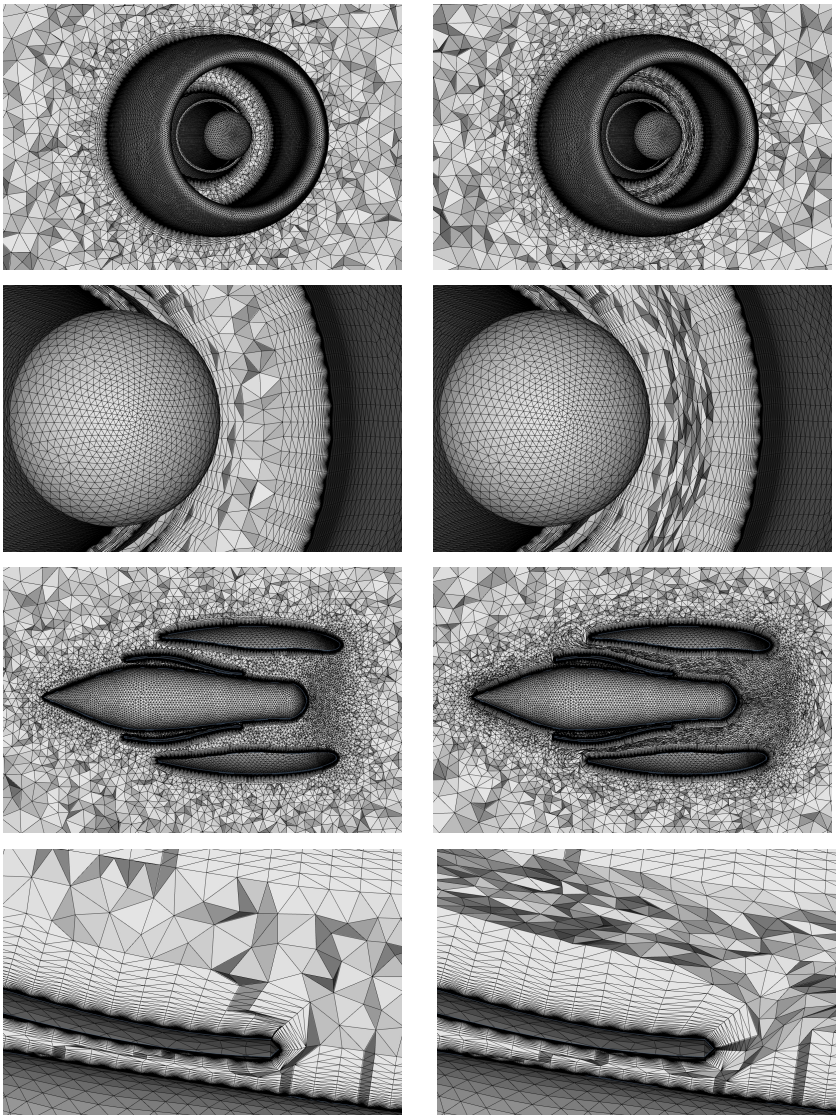

Fig. 2 Nacelle meshes obtained with the open (left) and closed (right) advancing-layer methods

multiple sections with opposing surfaces that are close together. The initial surface mesh is depicted in Figure 1 for this configuration. The surface mesh is composed of 35 357 vertices and 70 706 triangles. The BL parameters are the following: the initial BL spacing is set to 2×10^{-4} and the growth rate is set 1.2. Initially 33 811 vertices are active for BL generation. The closed method starts from an initial unstructured volume mesh composed of 224 789 vertices and 1 227 482 tetrahedra.

Mesh statistics and CPU time for each method are reported in Table 1. These meshes are depicted in Figure 2. Overall the results are quite similar in the resulting mesh as well as the CPU time required. For the open method, 44% of the CPU time is spent to generate the BL region and 56% for the outer part. For the closed method, 5% of the CPU is spent in the BL prescription, 64% in solving the elasticity, 15% in optimizing the mesh, 8% in managing the mesh movement and 8% in the remaining parts. Detail views of the merging BL regions between components show that the moving mesh algorithm of the closed advancing-layer method is able to insert the BL in narrow passages. The distortion of the field elements in these regions ends up providing a natural blending from the BL normal spacing between the components whatever the initial mesh prescription.

4.2 Wing-Body-Tail-Nacelle

The second test case is a wing-body-tail-nacelle configuration with high-quality isotropic unstructured triangular faces on the surface. This case was chosen as it has more overall complexity and multiple concave and merging regions. It also has considerably more surface faces and results in a large number of tetrahedra in the BL region. This highlights the scaling characteristics of the open and closed advancing-layer methods. The initial surface mesh is depicted in Figure 3 for this configuration. The surface mesh is composed of 245 472 vertices and 490 936 triangles. The BL parameters are the following: the initial BL spacing is set to 10^{-4} and the growth rate is set 1.2. Initially, 238 639 vertices are active for BL generation. The closed method

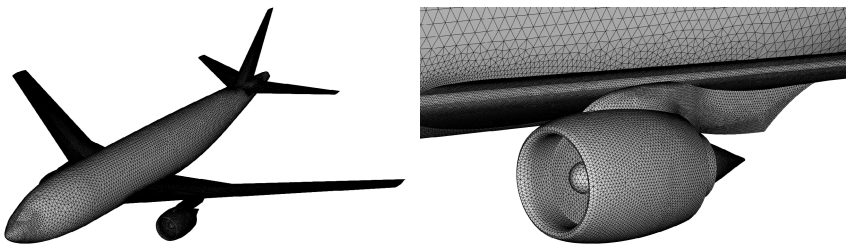

Fig. 3 Wing-body-tail-nacelle unstructured tri faces surface mesh

Table 2 Wing-body-tail-nacelle test case. Mesh statistics and CPU time.

Method	# layers	# vertices	# tets	# BL tets	CPU time
Open	55	13 473 601	80 004 700	68 942 146	18min
Closed	55	14 539 244	86 341 381	72 625 667	33min

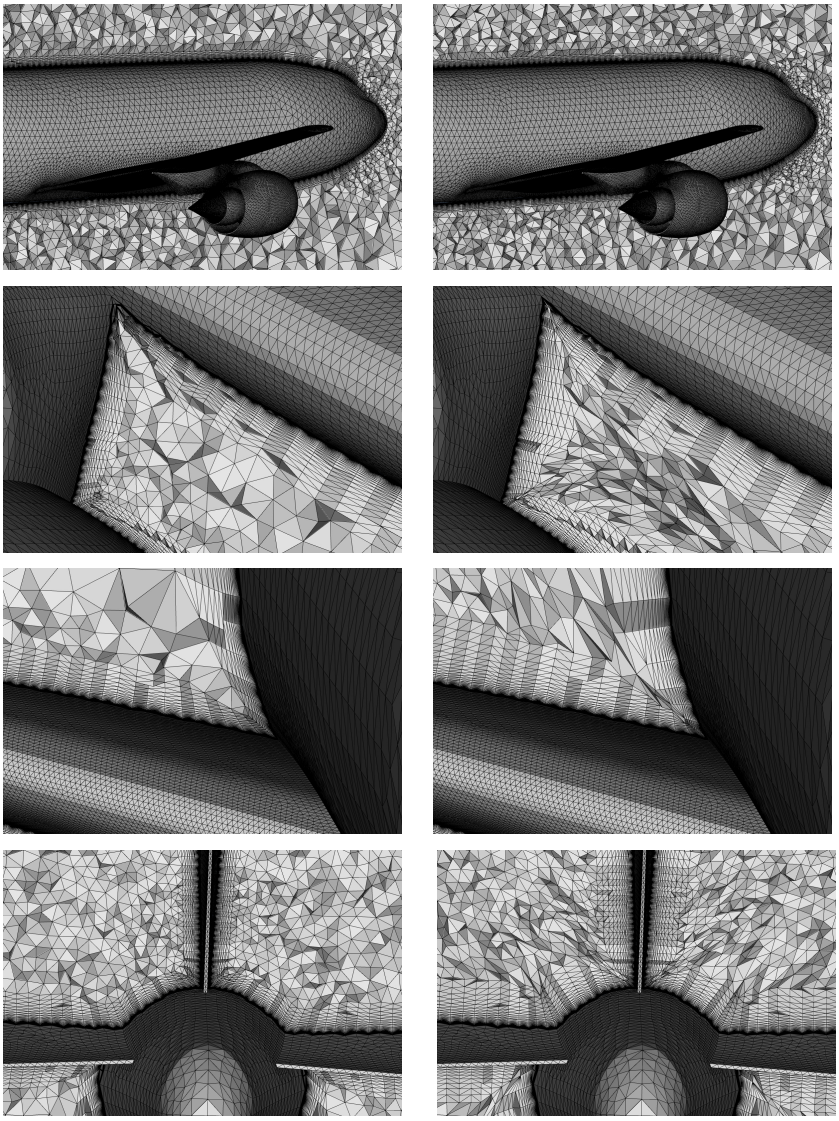

Fig. 4 Wing-body-tail-nacelle meshes obtained with the open (left) and closed (right) advancing-layer methods

starts from an initial unstructured volume mesh composed of 2 435 027 vertices and 13 691 340 tetrahedra.

Mesh statistics and CPU time for each method are reported in Table 2. These meshes are depicted in Figure 4. Again, overall the results are quite similar in the resulting mesh as well as the CPU time required. The additional CPU time for the closed advancing-layer method is more evident in this case. For the open method, 39% of the CPU time is spent to generate the BL region and 61% for the outer part. For the closed method, 3% of the CPU is spent in the BL prescription, 78% in solving the elasticity, 10% in optimizing the mesh, 7% in managing the mesh movement and 2% in the remaining parts. Detail views of the wing-nacelle junction, wing-body junction and tail regions show that the moving mesh algorithm of the closed advancing-layer method is able to effectively insert the BL in concave regions. The distortion of the field elements in these regions again ends up providing a natural blending from the BL normal spacing in these areas. This is particularly noticeable in the wing-nacelle junction (second row of figures in Figure 4) where the closed method is able to narrow the distance between merging layers.

4.3 Landing Gear

The third example is a landing gear geometry provided by NASA, cf. Figure 5. This geometry is extremely complex with several geometric details and presents many boundary layer collision. It points out the robustness of both algorithms to handle geometry configurations representative of industrial applications. The surface mesh is composed of 404 642 vertices and 809 320 triangles. The BL parameters are the following: the initial BL spacing is set to 10^{-4} and the growth rate is set 1.2. The initial number of active nodes for the BL generation is 404 230. The closed advancing-layer method starts from an initial unstructured volume mesh composed of 3 671 082 vertices and 20 586 391 tetrahedra.

Mesh statistics and CPU time for each method are reported in Table 3. For the open method, 56% of the CPU time is spent to generate the BL region and 44% for the outer part. For the closed method, 5% of the CPU is spent in the BL prescription, 70% in solving the elasticity, 11% in optimizing the mesh, 10% in managing the mesh movement and 4% in the remaining parts. These meshes are depicted in Figure 6. As with the previous cases the overall results are quite similar in the resulting mesh as well as the CPU time required. Characteristics are similar with the closed method fully filling concave junctions and open method slightly under-resolving them. Both methods are able to allow full BL advancement from directly opposing surfaces with a smooth transition in between.

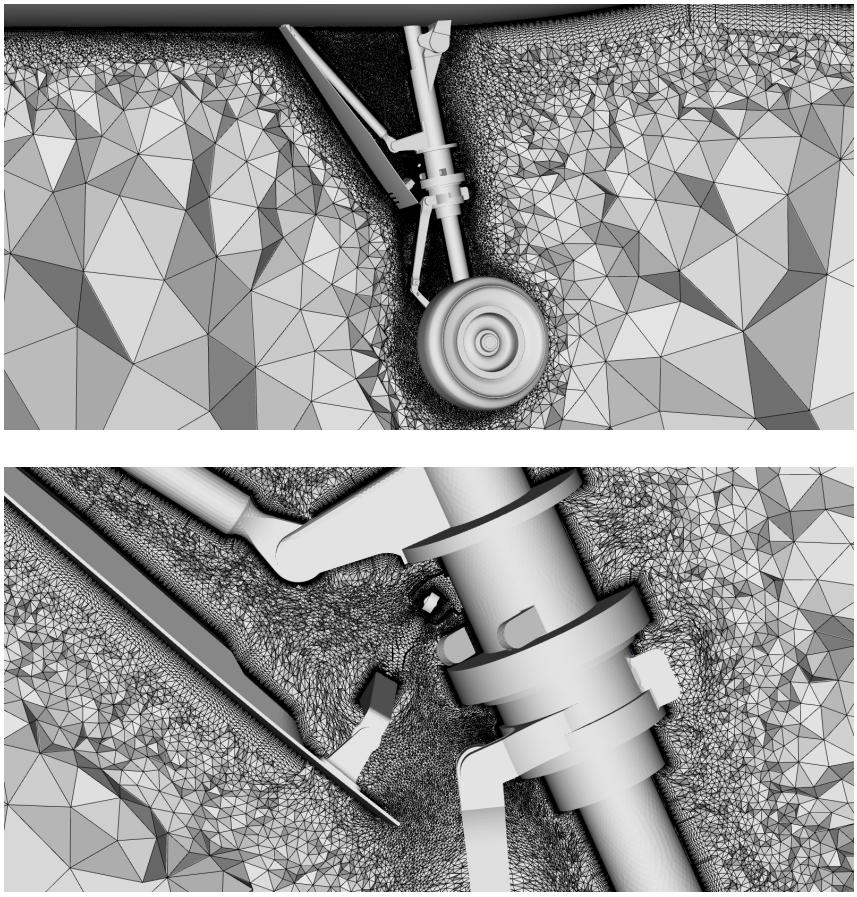

Fig. 5 General view of the landing gear boundary layer mesh obtained with the closed advancing-layer method

4.4 Mesh Quality

In order to compare the closed and the open methods, comparison criteria must be chosen. Quality functions such as that given by Relation (2) are not good criteria for comparison since the optimal metric is not known, *i.e.,* the metric that takes into account the blending between the BL region and the outer region. For example in all of the previous cases, the quality given by Relation (2) was between 1 and 2 for over 90% of the elements. However, extremes values that varied from 10 to 200 were typically in regions where high distorted elements were advantageous, *i.e.,* narrow regions.

An alternative criterion that quantifies the length-scale transition between the BL and outer region can be constructed from the volume ratio between neighboring elements that are in the BL transition region:

Table 3 Landing gear test case. Mesh statistics and CPU time for each method.

Method	# layers	# vertices	# tets	# BL tets	CPU time
Open	43	13 558 441	79 949 586	65 899 111	48min
Closed	51	15 367 304	90 866 372	70 179 445	52min

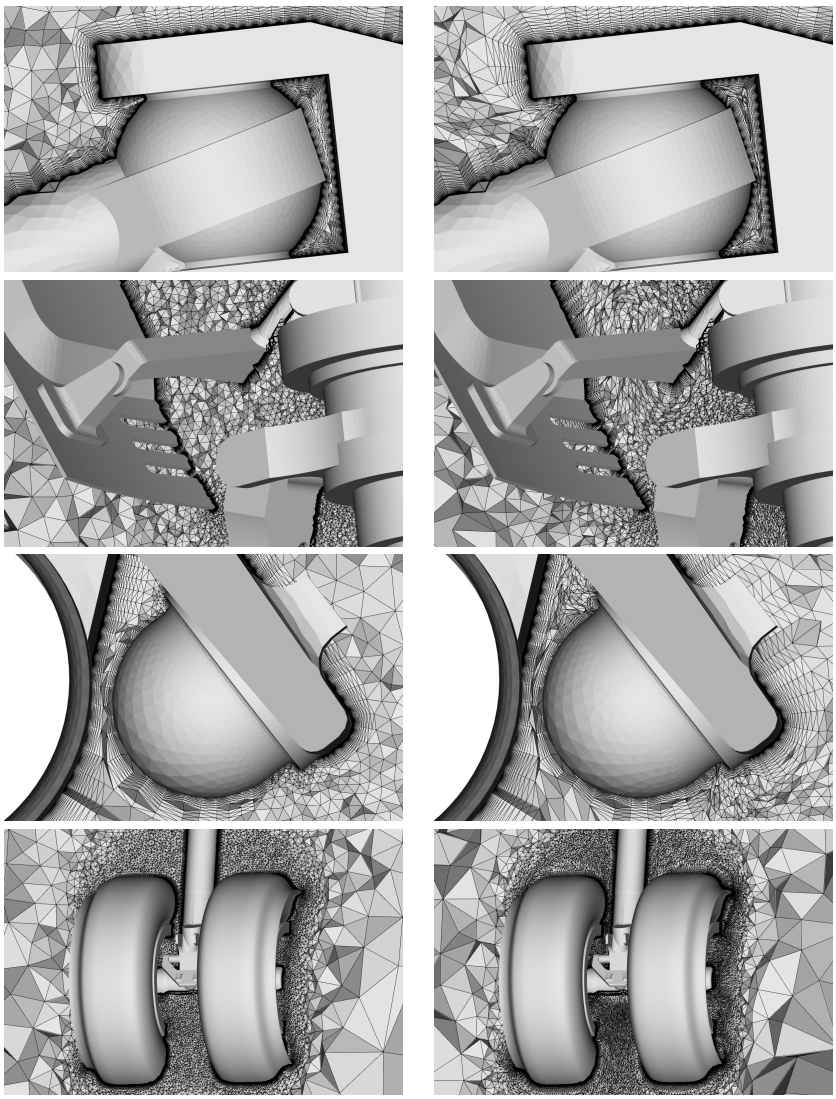

Fig. 6 Landing gear meshes obtained with the open (left) and closed (right) advancing-layer methods

$$Q_{VR} = \frac{\max(|K_{bl}|, |K_{outer}|)}{\min(|K_{bl}|, |K_{outer}|)} \tag{4}$$

where K_{bl} and K_{outer} are BL and outer regions tetrahedra, respectively, that are sharing a common face. For most CFD flow solvers, a low value for Q_{VR} benefits efficiency and accuracy. Therefore, a smooth blending of the anisotropy typically present in the BL transition region is advantageous. Figure 7 presents the BL transition region volume ratio histograms for the results of the previous cases with both methods. As shown and as expected, the closed method produces a more optimal volume ratio in the BL transition region. Similar results can be obtained with an open method if metric-based blending is used for a smoother transition [19].

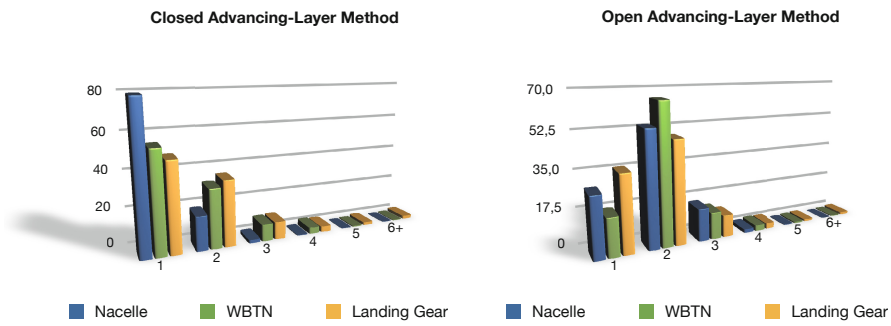

Fig. 7 Comparison of volume ratio histograms for the open (left) and closed (right) advancing-layer methods

4.5 ONERA M6 Wing Anisotropic Adapted Mesh

The three previous examples have illustrated that the closed advancing-layer method produces results that are comparable to the open method. Now, we demonstrate that the closed method can be coupled with unstructured anisotropic mesh adaptation. To this end, a BL mesh is generated starting from an anisotropic adapted volume mesh of an ONERA M6 wing profile. The initial adapted mesh is composed of 118 053 vertices and 643 841 tetrahedra, cf. Figure 8. The BL parameters are the following: the initial BL spacing is set to 10^{-4}, the growth rate is set 1.2 and the generation of 20 layers is requested. The initial number of active nodes for the BL generation is 14 736.

The final mesh with the BL is composed of 412 139 vertices and 2 409 074 tetrahedra, see Figure 9. 1 763 523 tetrahedra are inside the BL region. The CPU time to generate the BL mesh is 7min: 1% of the CPU being spent in the BL prescription, 58% in solving the elasticity, 35% in optimizing the mesh, 5% in managing the mesh movement and 1% in the remaining parts. Detailed views in Figure 9 points out that the closed advancing-layer method

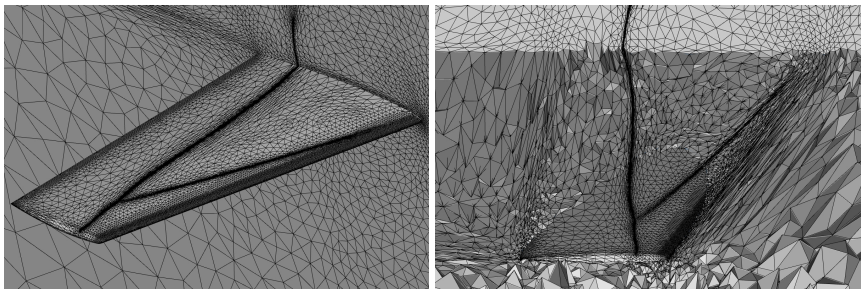

Fig. 8 Anisotropic adapted surface and volume mesh of an ONERA M6 wing

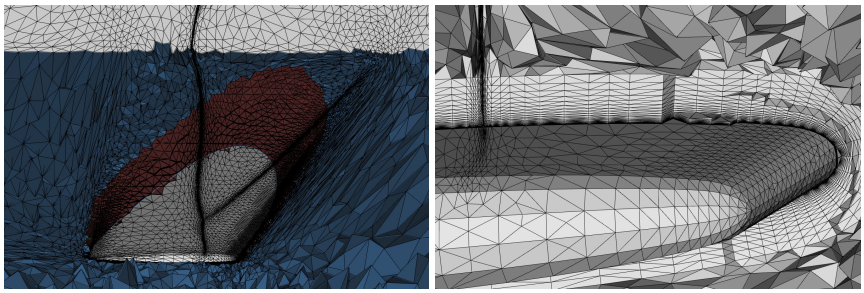

Fig. 9 BL mesh generated with the closed advancing-layer method for the ONERA M6 wing anisotropic adapted mesh. For the left picture, BL region is in red, outer region is in blue and surface mesh in grey. Anisotropic mesh adaptation is preserved while inserting the BL mesh.

was able to push the BL mesh while preserving anisotropic adapted regions of the initial mesh as all mesh optimization operators work in the given metric field.

5 Conclusion

A new closed advancing-layer method capable of generating high-quality pseudo-structured high-aspect-ratio elements in the BL region was presented and evaluated. The approach utilizes a recently developed changing topology moving mesh strategy for deforming the volume mesh as the BL interface is inflated. Results for typical aerospace configurations were presented and directly compared to those for a well established open advancing-layer method. The overall results indicate that the closed method yields similar results in terms of mesh quality and efficiency, and that the changing topology moving mesh strategy is an efficient and effective method for deforming the unstructured volume mesh. Details of the results illustrate the method handles BL front collisions very effectively in allowing layers to merge close together and producing a natural smooth anisotropic blending between colliding layers.

Moreover, it provides a robust strategy to couple unstructured anisotropic mesh adaptation in the outer region and pseudo-structured high-aspect-ratio elements in the BL region. The efficiency for the closed method lagged slightly behind that of the open method and required $1.5 - 2$ times the CPU time to generate the BL region mesh. Most of the closed advancing-layer method CPU time is spent in solving the elasticity PDE. This CPU time can be substantially reduced by considering an elasticity dedicated mesh to solve the mesh deformation as proposed in [2]. Therefore the differences between open and closed method CPU times are expected to be significantly minimized.

References

1. Alauzet, F.: Contributions aux méthodes numériques pour l'adaptation de maillage et le maillage mobile. Habilitation à Diriger des Recherches, Université Pierre et Marie Curie, Paris VI, Paris, France (2012)
2. Alauzet, F.: Efficient moving mesh technique using generalized swapping. In: Jiao, X., Weill, J.-C. (eds.) Proceedings of the 21st International Meshing Roundtable, vol. 123, pp. 17–37. Springer, Heidelberg (2013)
3. Aubry, R., Löhner, R.: Generation of viscous grids at ridges and corners. Int. J. Numer. Meth. Engng. 77, 1247–1289 (2009)
4. Baker, T.J., Cavallo, P.: Dynamic adaptation for deforming tetrahedral meshes. AIAA Journal 19, 2699–3253 (1999)
5. Bottasso, C.L., Detomi, D.: A procedure for tetrahedral boundary layer mesh generation. Engineering Computations 18, 66–79 (2002)
6. Dobrzynski, C., Frey, P.J.: Anisotropic Delaunay mesh adaptation for unsteady simulations. In: Proceedings of the 17th International Meshing Roundtable, pp. 177–194. Springer (2008)
7. Frey, P.J., George, P.L.: Mesh generation. Application to finite elements, 2nd edn. ISTE Ltd. and John Wiley & Sons (2008)
8. Garimella, R.V., Shephard, M.S.: Boundary layer mesh generation fro viscous flow simulations. Int. J. Numer. Meth. Fluids 49, 193–218 (2000)
9. George, P.L., Hecht, F., Saltel, E.: Automatic mesh generator with specified boundary. Comput. Methods Appl. Mech. Engrg. 92, 269–288 (1991)
10. Hassan, O., Morgan, K., Probert, E.J., Peraire, J.: Unstructured tetrahedral mesh generation for three-dimensional viscous flows. Int. J. Numer. Meth. Engng. 39, 549–567 (1996)
11. Ito, Y., Nakahashi, K.: Unstructured mesh generation for viscous flow computations. In: Proceedings of the 11th International Meshing Roundtable, pp. 367–377 (2002)
12. Löhner, R.: Generation of unstructured grids suitable for RANS calculations. In: 37th AIAA Aerospace Sciences Meeting (January 1999)
13. Löhner, R., Parikh, P.: Three-dimensional grid generation by the advancing front method. Int. J. Numer. Meth. Fluids 9, 1135–1149 (1988)
14. Loseille, A., Löhner, R.: Adaptive anisotropic simulations in aerodynamics. In: 48th AIAA Aerospace Sciences Meeting, AIAAP 2010-169, Orlando, FL, USA, pp. 2010–2169 (January 2010)
15. Loseille, A., Löhner, R.: Boundary layer mesh generation and adaptivity. In: 49th AIAA Aerospace Sciences Meeting, AIAAP 2011-894, Orlando, FL, USA (January 2011)

16. Marcum, D.L.: Generation of unstructured grids for viscous flow applications. In: 33th AIAA Aerospace Sciences Meeting (January 1995)
17. Marcum, D.L.: Adaptive unstructured grid generation for viscous flow applications. AIAA Journal 34(8), 2440–2443 (1996)
18. Marcum, D.L.: Unstructured grid generation using automatic point insertion and local reconnection. In: Thompson, J.F., Soni, B., Weatherill, N.P. (eds.) The Handbook of Grid Generation, ch. 18, pp. 1–31. CRC Press (1998)
19. Marcum, D.L., Alauzet, F.: Unstructured mesh generation using advancing layers and metric-based transition. In: 21th AIAA Computational Fluid Dynamics Conference, AIAAP 2013-2710, San Diego, CA, USA, pp. 2013–2710 (June 2013)
20. Marcum, D.L., Alauzet, F., Marechal, L.: `Bloom`: a closed advancing-layer boundary-layer mesh generator. Internal report, INRIA (2011)
21. Marcum, D.L., Weatherill, N.P.: Unstructured grid generation using iterative point insertion and local reconnection. AIAA Journal 33(9), 1619–1625 (1995)
22. Pirzadeh, S.: Viscous unstructured three dimensional grids by the advancing-layers method. In: 32nd AIAA Aerospace Sciences Meeting (January 1994)
23. Sharov, D., Nakahashi, K.: Hybrid prismatic/tetrahedral grid generation for viscous flow applications. AIAA Journal 36(2), 157–162 (1998)
24. Yang, Z., Mavriplis, D.J.: Higher-order time integration schemes for aeroelastic applications on unstructured meshes. AIAA Journal 45(1), 138–150 (2007)