

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305715551>

Anisotropic boundary layer mesh generation for immersed complex geometries

Article in *Engineering With Computers* · July 2016

DOI: 10.1007/s00366-016-0469-7

CITATIONS

0

READS

30

3 authors:



[Laure Billon](#)

MINES ParisTech

1 PUBLICATION 0 CITATIONS

SEE PROFILE



[Youssef Mesri](#)

IFP Energies nouvelles

22 PUBLICATIONS 53 CITATIONS

SEE PROFILE



[Elie Hachem](#)

MINES ParisTech

62 PUBLICATIONS 192 CITATIONS

SEE PROFILE

Anisotropic boundary layer mesh generation for immersed complex geometries

Laure Billon · Youssef Mesri · Elie Hachem

15/01/2016

Abstract This paper proposes a new method to build boundary layer meshes over an immersed complex geometry. It allows to generate an anisotropic semi-structured mesh with a smooth gradation of mesh size from a geometry immersed into an arbitrary coarse domain, while capturing and keeping the interface. The idea is to generate an *a priori* mesh fitting the geometry boundary layer which is ready for simulations. The mesh size distribution is driven by a levelset distance function and is determined using physical parameters available before the simulation, based on the boundary layer theory. The aspect ratio is then determined knowing the shape of the geometry, and all is applied in a metric tensor field using a gradation thanks to the new multi-levelset method. Then, the mesh generator adapt the initial mesh on the given metric field to create the desired boundary layer mesh.

Keywords meshing · metric · boundary layer · immersed

1 Introduction

The success of the finite element method lies essentially in its flexibility to represent the geometry on which the problem is defined. However, for several applications with highly complex geometries or very localized phenomena (such as interfaces and boundary layers), generating a correct geometry representation is a difficult task. In this work, we consider an alternative approach based on a unified mesh with immersed geometries.

Some of those problems include fluid flow around a solid that generates a boundary layer, a wake and, at a sufficient velocity, some turbulence and detachment phenomena. These phenomena happen at different scales and are characterized by a sharp flow gradient in a certain direction, which leads to an highly anisotropic behavior. A minimum mesh size is required along the flow gradient direction, but much larger sizes can be used in the other directions. It results in highly anisotropic element, which are particularly interesting to improve computational efficiency. However, in order to do an accurate numerical simulation of a turbulent flow problem around a complex geometry, the space needs to be properly

discretised. Indeed elements size, their aspect ratio and corresponding directions impact accuracy and efficiency of the computation as much as their smooth variation.

This paper proposes a technique for the construction of semi-structured boundary layers over immersed 3D complex geometries. The interface of the immersed 3D object is implicitly represented by a distance function, also called levelset, which is used as a geometric tool to initialize and keep a surface with some desirable properties such as smoothness and constant known gradient. The distance function is defined as a scalar field for each vertex and filled with its distance to the surface of the immersed object [13]. This distance field is used to derive a metric map that allows capturing the interface through the application of local remeshing algorithms [19]. Many authors (for example [3, 2, 11]) showed that the definition of a metric field simplifies the generation of adapted and anisotropic meshes. The metrics associated to the Riemannian space specify the sizes of the mesh and the corresponding directions of the stretching. Then an adapted and anisotropic mesh in the Euclidean space can be represented as an isotropic and unitary mesh in the Riemannian space.

Once the anisotropic mesh around the interface is built, the boundary layer must be carefully constructed with consideration given to the flow physics and the abilities of the flow models used [24, 5, 23, 14]. In the classical body-fitted methods, boundary layers have been studied extensively both experimentally and computationally and near wall resolution requirements have been described for different modeling approaches [17, 18, 10, 23, 4, 6]. These well-established techniques will be extended here in the framework of immersed methods.

For non-structured meshes, this is a very challenging task as one has to devise an automatic way to build a semi-structured boundary layer in the normal direction to the implicit surface. We will perform this task by using the concept of mesh generation controlled by a metric map as mentioned above. We can consider here, our approach as a unified metric-based *a priori* and *a posteriori* mesh generation and adaptation. Indeed, in a number of problem classes, typically viscous boundary layers which form near walls, it is desirable to have the highly anisotropic mesh in specific regions that has a semi-structured nature and to have that structure maintained during transient mesh adaptation. The metric framework allows combining both *a priori* and *a posteriori* size maps in a flexible way without need to rigid geometric constraints on the boundary layer regions. In this paper we will focus on the *a priori* mesh generation part.

The paper is organized as follows. First, the boundary layer theory is recalled in Section 2. In Section 3, after a brief reminder on the main principles of metric-based mesh adaptation, we focus on the normal mesh size and the new multi-levelset method. Section 4 is dedicated to the study of the geometry shape and to its consideration in the building of the metric tensor. Then, Section 5 presents simple 2D to complex 3D meshes illustrating this procedure. Finally, Section 6 is dedicated to conclusions and future work.

2 Boundary layer theory

The boundary layer is the area close to the wall where the viscous effects of the fluid are predominant. Because of its viscosity the fluid adhere to the wall, such that the velocity of the fluid on the wall is null, the velocity increases with the distance to the wall. This zone close to the wall where the fluid velocity is below 99% of the free stream velocity (u_∞) is called the boundary layer (Figure 1). The boundary layer is critical in the study of high Reynolds flows as it is the place where turbulence occurs, vortices begin, and detachments appear. So that the accuracy of the numerical simulation depends almost on the quality of the

mesh in the boundary layer. The boundary layer mesh needs to be fine enough according to the solver used to capture all turbulence phenomena. We will use the boundary layer theory [24, 7] and particularly its structure to design our meshes.

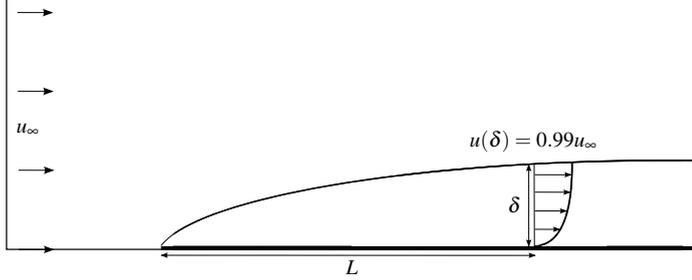


Fig. 1 Boundary layer over a flat plate, where δ is its thickness at a defined distance L from the leading edge in the stream-wise direction

Recall that, boundary layers are composed of several layers, one on top of each other, where different flow behavior apply. In the first layer the flow is laminar, and became gradually turbulent from one layer to the next one, as vortices appear and growth. So, to capture accurately the physical behavior of the flow in the boundary layer, we have to build a mesh which respects this structure. It means that we will try to construct the mesh in layers, where the first layer has a thickness h_{min} and the next layers has h_{min} increased by a factor growth α as thickness, and so on, until the global boundary layer thickness δ is reached (Figure 2).

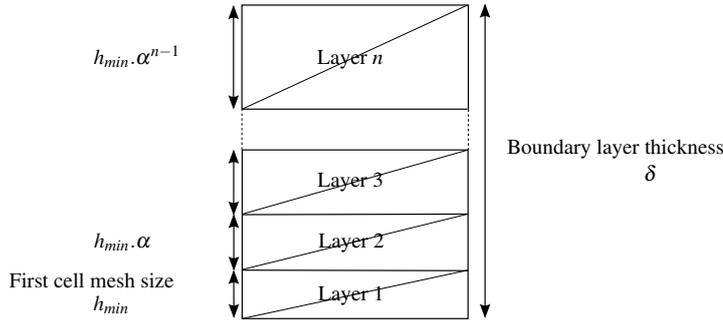


Fig. 2 Structure of boundary layers

The first layer thickness h_{min} is defined as:

$$h_{min} = \frac{y_0^+ \nu}{u_\tau} \quad (1)$$

where y_0^+ referred to the dimensionless wall distance for the first cell, which is fixed considering the expected accuracy and the computational resources available. Moreover u_τ is the friction velocity, the flow velocity close to the wall where the fluid viscosity generates a

frictional force τ_ω called wall shear stress defined by [7]:

$$u_\tau = \sqrt{\frac{\tau_\omega}{\rho}} \quad (2)$$

with

$$\tau_\omega = \mu \left. \frac{\partial u}{\partial y} \right|_{y=0} \quad (3)$$

As it is impossible to determine τ_ω without knowing the velocity solution, and as we try to build an *a priori* mesh we propose to estimate it using the skin friction. The skin friction coefficient is obtained by normalizing this constraint by the dynamic pressure:

$$C_F = \frac{\tau_\omega}{\frac{1}{2}\rho u_\infty^2} \quad (4)$$

For which we use the Schlichting approximation [24] valid for all flows with a Reynolds number below 10^9 and tested on flat plate:

$$C_F = [2\log_{10}(Re_L) - 0.65]^{-2.3} \quad (5)$$

Therefore, we can simplify (1) in order to determine the minimum mesh size on the wall of the boundary layer only in function of the Reynolds number Re_L of the flow to study at the characteristic length L of the problem, and the accuracy desired for the mesh y_0^+ :

$$h_{min} = \frac{Ly_0^+}{Re_L \sqrt{\frac{C_F}{2}}} \quad (6)$$

Moreover, power laws allow us to give an approximation of the boundary layer thickness δ for both laminar and turbulent flows [7]:

$$\frac{\delta_{lam}}{L} = \frac{5}{Re_L^{1/2}} \quad (7)$$

$$\frac{\delta_{turb}}{L} = \frac{0.38}{Re_L^{1/5}} \quad (8)$$

The last parameter needed to perfectly determine the structure of the boundary layer is the growth factor, which is used to allow the cell mesh sizes to increase from the wall to the end of the boundary layer. As it is not affordable to impose h_{min} as mesh size to all cells in the boundary layer and moreover it is useless. Thus cell mesh sizes are allowed to increase in the direction normal to the wall by a growth factor α which is conventionally set to 1.2. Note also that the number of layers can be easily deduce from the previous parameters using the following expression:

$$n = \frac{\ln\left(1 - \delta \frac{1-\alpha}{h_{min}}\right)}{\ln(\alpha)} \quad (9)$$

3 Metric-based mesh generation

In this section, all the procedure to generate a mesh as detailed in the previous section will be presented. In order to build the boundary layer mesh, a new technique has been developed. This technique allows to build a semi-structured mesh, ensuring smooth transitions between mesh sizes all along the domain, and optimizes the number of nodes. Moreover the procedure is automatic and easy to parametrized. The idea is to build the mesh from a metric tensor which takes into account the physical parameters of the simulation.

3.1 Principles of grid generation using metrics

The purpose of this paragraph is to define the concept of metrics that is widely used in this paper. Non Euclidean metrics are often used to generate unstructured anisotropic meshes [17, 16]. The main idea of metric-based mesh adaptation, is to generate a unit mesh in a prescribed Riemannian metric space. Such that this mesh can be created by any unstructured uniform mesh generator. Then, this mesh is converted in an unstructured anisotropic mesh by the inverse application in the physical space.

A metric M in $\mathbb{R}^{d \times d}$ is a real symmetric positive definite matrix, where d represents the dimension of space. Thus, the metric M can be diagonalized:

$$M = {}^t R \Lambda R \quad (10)$$

where R is an orthogonal matrix whose lines are composed of M eigenvectors $(v_i)_{i=1..d}$ while Λ is the diagonal matrix composed of M eigenvalues $(\lambda_i)_{i=1..d}$, which are strictly positive.

The orthogonalization property of the metric allows us to define the dot product of two vectors in \mathbb{R}^d with respect to the metric M :

$$(\mathbf{u}, \mathbf{v})_M = (\mathbf{u}, M\mathbf{v}) = {}^t \mathbf{u} M \mathbf{v} \in \mathbb{R} \quad (11)$$

The associated norm of a vector in \mathbb{R}^d is then defined by:

$$\|\mathbf{u}\|_M = \sqrt{(\mathbf{u}, \mathbf{u})_M} \quad (12)$$

Knowing that R is an orthogonal matrix, so ${}^t R \cdot R = I_d$, thus one can deduce that application $\Lambda^{1/2} R$ defines the mapping from the physical space (\mathbb{R}^d, I_d) to the metric space (\mathbb{R}^d, M) . Then, we can recover:

$$(\mathbf{u}, \mathbf{v})_M = ((\Lambda^{1/2} R)\mathbf{u}, (\Lambda^{1/2} R)\mathbf{v}) = {}^t \mathbf{u} M \mathbf{v} \quad (13)$$

The geometrical representation of a metric tensor is the ellipsoid which represent the unit ball for the dot product (Figure 3):

$$\mathcal{E}_M = \{\mathbf{u} \mid \|\mathbf{u}\|_M = 1\} \quad (14)$$

In the context of mesh adaptation, a metric is defined at each vertex of the initial mesh and sent to the mesh generator, which build the desired mesh. The mesh is then generated using metrics, then in order to drive the mesh we only have to fill Λ with $\lambda_i = 1/h_i^2$, where h_i is the desired mesh size in the desired direction \mathbf{v}_i , and finally fill R with the prescribed directions \mathbf{v}_i .

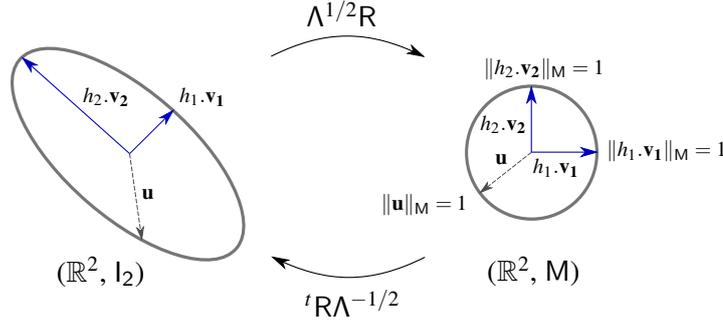


Fig. 3 Geometric interpretation of a metric tensor in 2 dimensions, left in physical space, right in metric space

3.2 Mesh size

3.2.1 Immersed interface

In monolithic domain framework, geometries are immersed in the domain using a surface mesh (STL stereolithography format), or directly a NURBS (see [12] for details). The first step of the proposed mesh adaptation algorithm is the construction of the levelset field (ϕ) on the computational mesh (Ω) for the immersed geometry (Ω_{im}). It is obtained by computing the minimum distance between the immersed geometry and the nodes of the computational mesh. Therefore, the interface Γ_{im} is defined as the zero isovalue of the levelset function:

$$\begin{cases} \phi(x) = d(x, \Gamma_{im}), & x \in \Omega \cap \Omega_{im} \\ \phi(x) = -d(x, \Gamma_{im}), & x \in \Omega \setminus \Omega_{im} \\ \Gamma_{im} = \{x, \phi(x) = 0\} \end{cases} \quad (15)$$

Note that the same algorithm can be used in the case of body fitted meshes. Indeed, the immersed approaches are also interesting in the case of moving boundaries or multiphases flow thanks to their ability to build, using an appropriate method, easily and automatically meshes around complex geometries. Drawbacks of immersed technique lie in the difficulty to smoothly describe the geometry interface [21] and also the computational cost associated to the residual nodes inside the geometry. We will come back to these challenging issues.

Moreover, as the initial mesh is chosen arbitrarily and usually coarse, it remains difficult to capture the geometry interface when the features of the immersed geometry are smaller than the element size. We propose to perform in this case a pre-meshing iterative procedure that consists in inserting additional nodes around the immersed geometry as described in Figure 4.

3.2.2 Multi-levelset

In order to build the boundary layer mesh layer by layer in a structured way using unstructured elements, the second step of the proposed mesh adaptation algorithm is to localize, using a multi-levelset method, all the needed sub-layers inside the boundary layer thickness as shown in Figure 5. Indeed, one cell in the wall-normal direction is prescribed (18) between two levelset functions. Thereby the vertices inside each layer are well identified and

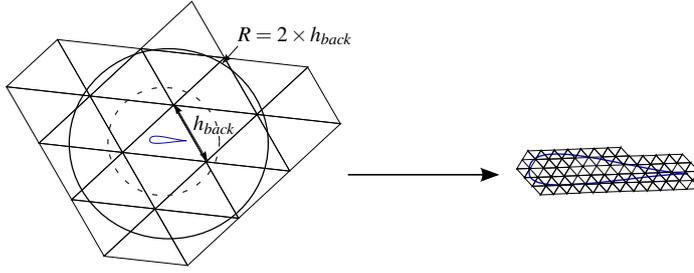


Fig. 4 Pre-meshing procedure for initial coarse meshes

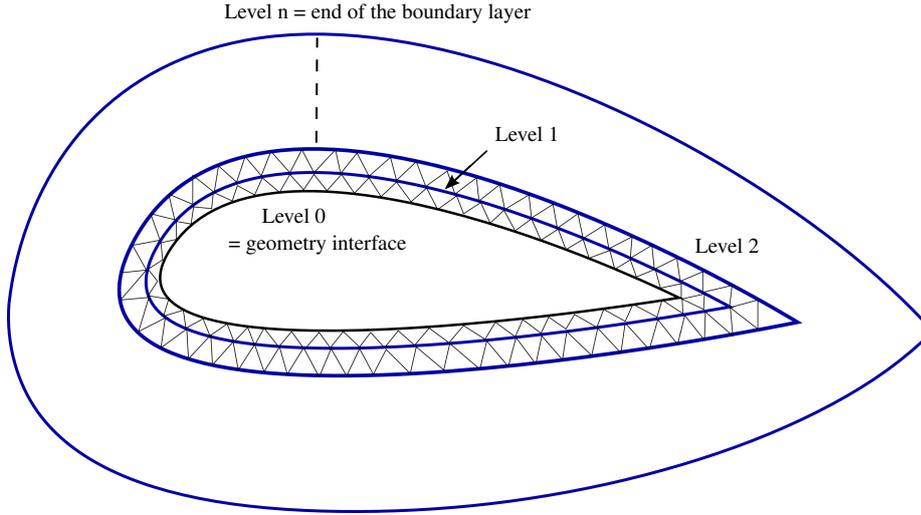


Fig. 5 Each cell layer of the boundary layer is defined using a levelset function. The zero isovalue of each levelset is represented

the corresponding mesh size previously defined can be imposed. This technique is really interesting as it drives the mesh adaptation and structures it.

Thus, the first layer is represented by a levelset function equals to the levelset of the object at the same vertex plus h_{min} , the distance between the two levelset functions. Moreover, we imposed h_{min} as mesh size between these two levelsets. The k -layer is then represented using a levelset function equals to the levelset of the $(k-1)$ -layer, $Levels[k-1]$, at the same vertex plus the distance between the two levelset functions. Moreover, the k -layer levelset, $Levels[k]$, is also equals to the levelset of the object, $Levels[0]$, at the same vertex plus the thickness of the $k-1$ first layers.

Thus:

$$\begin{aligned} Levels[k] &= Levels[k-1] + h_{min} \cdot \alpha^{k-1} \\ &= Levels[0] + h_{min}(1 - \alpha^k)/(1 - \alpha) \end{aligned}$$

where $h_{min} \cdot \alpha^{k-1}$ is also the mesh size imposed between $Levels[k-1]$ and $Levels[k]$. And so on until the end of the boundary layer, represented by the n -layer (see Figure 2).

Remark 1 For several applications the internal flow is not studied then we work on minimizing as much as possible the number of vertices inside the geometry by using a large growth factor inside as it gets far from the interface. However to keep accurate computation in the wall vicinity we have to ensure that the interface is still well captured. Therefore, we duplicate the first three layers in the internal side of the interface. It allows to save degrees of freedom and therefore computational time.

3.2.3 Size gradation over the domain

Using the same approach, we can prescribe the mesh size distribution in the rest of the domain by combining a size gradation and the distance function. It allows us to control the element sizes in the vicinity of the geometry and in wake regions as well as to increase progressively the mesh size far from the interface and thus minimizing the total number of elements. Indeed, to solve accurately the relevant features, a fine mesh has to be ensured in the vicinity of the interface and in the wake regions. Moreover, the quality of the mesh, and thereby the accuracy of the simulation strongly depends on the smooth varying of cell mesh size over the domain.

To do so, we simply propose to compute the levelset function for two boxes: the first one covers the whole domain (Box1) while the second one covers the immersed interface zone (Box2) as shown in Figure 6. The position as well as the size of Box2 are defined by the end user and normally depend on the study case and the nature of the flow.

Finally, to ensure a smooth transition all along these zones, we set the mesh size to grow linearly from the maximum mesh size of the boundary layer, $h_{min} \cdot \alpha^{n-1}$, to the maximum mesh size defined on Box2. Then a logarithmic growth of the mesh size is applied from Box2 to Box1.

All the obtained mesh sizes h_n , computed for the wall-normal direction, can be gathered now to build an isotropic mesh using the following metric:

$$\mathbf{M} = \begin{pmatrix} \frac{1}{h_n^2} & 0 & 0 \\ 0 & \frac{1}{h_n^2} & 0 \\ 0 & 0 & \frac{1}{h_n^2} \end{pmatrix} = \frac{1}{h_n^2} \mathbf{I}_d \quad (16)$$

Note that for high Reynolds number flow, the required minimum mesh size in the first layers tends to very small values which, in the case of isotropic meshes, will lead to an extremely large number of elements and put an excessive demand on computational resources. Therefore, we propose to pursue using anisotropic elements in the boundary layer, high aspect ratio R_a in the tangential directions. The mesh size in this direction is then defined as $h_t = R_a \cdot h_n$.

As proposed in the previous sections, the multi-levelset method will be used to vary and to decrease the aspect ratio from the interface to the end of the boundary layer and to ensure a smooth transition between anisotropic and isotropic elements. The anisotropic ratio is then set such as h_t is equal to the maximum mesh size in Box2 (H_2) using the following expression:

$$R_a = \frac{H_2}{h_n} \quad (17)$$

We will show that such continuity in tangential mesh size produces the desired semi-structured mesh.

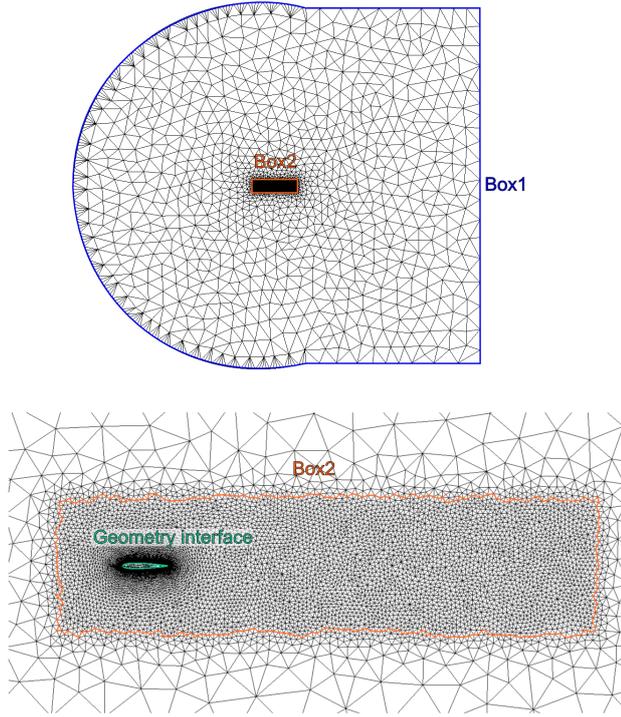


Fig. 6 Presentation of the domain subdivision

3.2.4 Mesh directions

In the last step to build the boundary layer metric, directions have to be properly defined. Therefore, let P be a point of the domain close to the geometry interface, let N be the normal vector to the surface, with ϕ the levelset function of the surface :

$$\mathbf{N} = -\frac{\nabla\phi(P)}{|\nabla\phi(P)|} \quad (18)$$

The tangential plane is defined by $\mathbf{I}_d - \mathbf{N} \cdot \mathbf{N}^T$, so one solution for the metric is:

$$\mathbf{M} = \begin{cases} \frac{1}{h_n^2} \mathbf{N} \cdot \mathbf{N}^T + \frac{1}{h_t^2} (\mathbf{I}_d - \mathbf{N} \cdot \mathbf{N}^T) & \text{inside Box2, where } h_t = H_2 \\ \frac{1}{h_t^2} \mathbf{I}_d & \text{outside Box2} \end{cases} \quad (19)$$

Several drawbacks for this formulation are observed at this level. First, the tangential mesh size does not depend on the geometry and consequently its shape, the curvature and the complexity of the geometry are not taken into account. Second, we only define one tangential mesh size even if we are in three dimensions whereas the geometry can have completely different behavior in its two tangential directions. Finally, the tangential directions are not well defined, such that the geometry interface can not be smoothly recovered by element faces in 3D. Nevertheless, this metric formulation is exact in terms of directions

for the 2D cases.

4 Curvature

In this section, we will focus on the geometry curvature to define properly the tangential directions and associated mesh sizes for any complex geometry. It will allow us to define properly the anisotropic ratio and to ensure that the interface is smoothly and well described.

First, let us define the shape operator S :

$$S = \nabla_T \mathbf{N} = \frac{1}{|\nabla\phi(P)|} (\mathbf{I} - \mathbf{N} \cdot \mathbf{N}^T) \text{Hes}(\phi)(P) \quad (20)$$

The Hessian matrix $\text{Hes}(\phi)$ measures the second-order rate of change of ϕ while $\mathbf{I} - \mathbf{N} \cdot \mathbf{N}^T$ projects it onto the tangent plane. Therefore, it is obvious that $S \cdot \mathbf{N} = 0$, so that \mathbf{N} is an eigenvector of S associated to the zero eigenvalue.

The two other eigenvalues of S and their corresponding eigenvectors are exactly the principal curvatures (κ_{min} and κ_{max}) and the principal directions associated to the geometry surface at P .

To determine these eigenvalues, let us find the roots of $\det(S - \lambda \mathbf{I}_d) = 0$. In three dimensions, it is well known that

$$\begin{cases} \det(S - \lambda \mathbf{I}_3) = \lambda^3 - \text{tr}(S)\lambda^2 + Z(S)\lambda - \det(S) \\ Z(S) = -\frac{1}{2}(\text{tr}(S^2) - \text{tr}(S)^2) \end{cases} \quad (21)$$

As zero is an eigenvalue of S , $\det(S)$ is necessarily zero. Moreover,

$$\det(S - \lambda \mathbf{I}_3) = \lambda(\lambda - \kappa_{max})(\lambda - \kappa_{min}) \quad (22)$$

Finally,

$$\begin{cases} \kappa_{max} = \text{tr}(S) + \sqrt{\left(\frac{\text{tr}(S)}{2}\right)^2 - Z(S)} \\ \kappa_{min} = \text{tr}(S) - \sqrt{\left(\frac{\text{tr}(S)}{2}\right)^2 - Z(S)} \end{cases} \quad (23)$$

The associated eigenvectors ($\mathbf{T}_1, \mathbf{T}_2$) are then deduced from a Gauss resolution. Such that $(\mathbf{N}, \mathbf{T}_1, \mathbf{T}_2)$ is an orthonormal basis at P .

Once the directions and their associated curvatures have been determined, curvatures need to be transform into tangential mesh sizes. As curvature corresponds to the inverse of the radius of the circle \mathcal{C} which could fit the surface at best at P , the tangential mesh size can be deduced from a geometrical approach as shown in Figure 7.

Let P coordinates be (x_P, y_P) , h_{min} be the mesh size in the wall-normal direction N and h_t be the mesh size in the tangential direction T . We want to recover h_t from the curvature κ and the normal mesh size. For that purpose, we need to solve $d(Q, \mathcal{C}) = h_{min}$ such that the nodes of the mesh stay close to the interface with a precision of h_{min} on the interface.

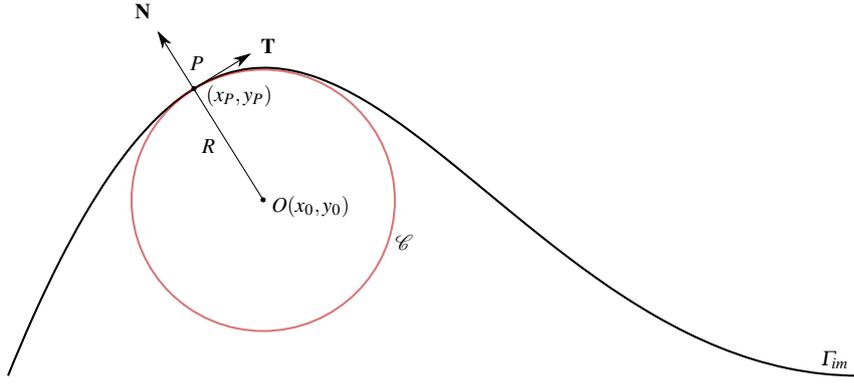


Fig. 7 The interface Γ_{im} is represented in black, while the best fit circle in $P \in \Gamma_{im}$ is drawn in red. The circle \mathcal{C} is centered in $O(x_0, y_0)$ and has for radius R . The vector normal and tangential to the interface at P are denoted \mathbf{N} and \mathbf{T}

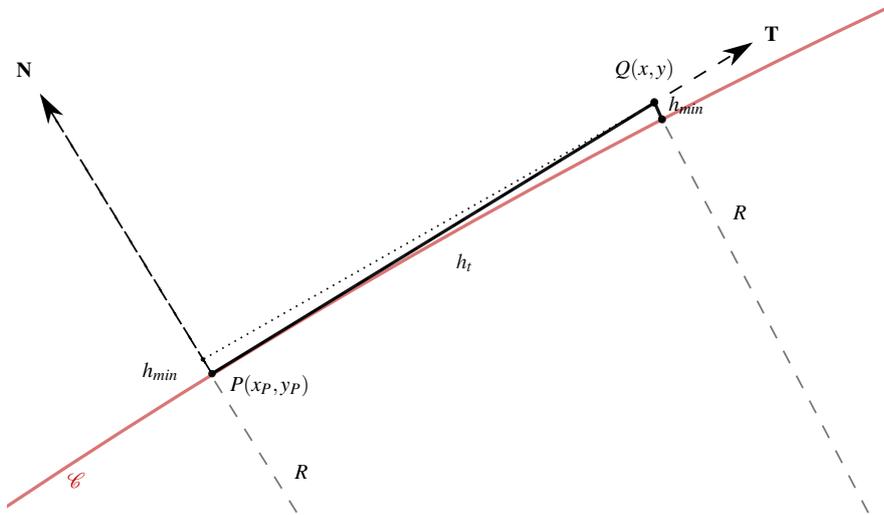


Fig. 8 This is a zoom in around P on figure 7. The point $Q(x, y)$ is the point where the circle with center O and radius $R + h_{min}$ cross the straight line driven by \mathbf{T} . Thus, h_t is the distance from P to Q

Thereby, let us consider Q , its coordinate are:

$$\begin{aligned} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} x_P \\ y_P \end{pmatrix} + h_t \cdot \mathbf{T} \\ &= \begin{pmatrix} x_P \\ y_P \end{pmatrix} + \frac{h_t}{R} \begin{pmatrix} y_0 - y_P \\ x_P - x_0 \end{pmatrix} \end{aligned} \quad (24)$$

with T the orthogonal vector to \mathbf{N} .

The point Q belongs to the circle of center O and of radius $R + h_{min}$, it means that:

$$(x - x_0)^2 + (y - y_0)^2 = (R + h_{min})^2 \quad (25)$$

Let us replace (x, y) with respect to (24):

$$\left(h_t \frac{y_0 - y_P}{R} + (x_P - x_0)\right)^2 + \left(h_t \frac{x_P - x_0}{R} + (y_P - y_0)\right)^2 = (R + h_{min})^2 \quad (26)$$

By expand and factor, we get:

$$((y_0 - y_P)^2 + (x_0 - x_P)^2) \left(1 + \left(\frac{h_t}{R}\right)^2\right) = (R + h_{min})^2 \quad (27)$$

Knowing that P belongs to \mathcal{C} , it simplifies:

$$R^2 \left(1 + \left(\frac{h_t}{R}\right)^2\right) = (R + h_{min})^2 \quad (28)$$

Thus,

$$h_t = \sqrt{2Rh_{min} + h_{min}^2} \quad (29)$$

Finally, to prevent the case of straight line where $\kappa = 0$ and then $h_t = +\infty$, a maximum and a minimum mesh size have to be defined. The minimum mesh size is defined as h_n and the maximum one as H_2 where h_n is the local mesh size in the normal direction computed from the boundary layer theory and H_2 is the user defined largest mesh size in Box2.

The boundary layer metric can now be written as:

$$M = \begin{pmatrix} \mathbf{N} \\ \mathbf{T}_1 \\ \mathbf{T}_2 \end{pmatrix} \begin{pmatrix} \frac{1}{h_n} & 0 & 0 \\ 0 & \frac{1}{h_{t1}} & 0 \\ 0 & 0 & \frac{1}{h_{t2}} \end{pmatrix} (\mathbf{N} \ \mathbf{T}_1 \ \mathbf{T}_2) \quad (30)$$

5 Numerical results

In this section, we present several 2D and 3D numerical examples to illustrate the effectiveness of the proposed mesh adaptation procedure allowing to produce an *a priori* mesh adapted on the boundary layer of the immersed geometry. The mesh fit the boundary layer with semi-structured highly anisotropic elements as prescribed in the metric. The MTC parallel mesher and remesher [11, 19] is used to generate meshes from the boundary layer metric previously defined.

5.1 Elementary examples

We first consider elementary geometries that are immersed in a larger domain $[0, 8.5] \times [0, 2.75]m^2$. This study is considered as a first step to ensure that the shape is well captured and sharp angles are well handled.

5.1.1 Flat plate

The iterative boundary layer *a priori* meshing procedure is highlighted on the very well known flat plate benchmark. The leading edge is placed at a distance $x = 0.5m$ of the inlet, the characteristic length of the flat plate is $4m$, the Reynolds number is 10^4 (turbulent) and $y_0^+ = 1$. Such that, the minimum mesh size to reach is $h_{min} = 5.61 \cdot 10^{-3}m$. The initial background mesh size $H_{back} = 0.45m$, the maximum mesh size in Box2 is set to $H_2 = 0.1m$ and in Box1 is set to $H_1 = 0.5m$. This leads to a final boundary layer composed of 7 layers with a maximum anisotropic ratio of 18.

Figure 9 depicts the iterative procedure step by step, starting from the arbitrary coarse mesh towards the boundary layer mesh. The minimum mesh size is progressively decreased from H_{back} to h_{min} at each iteration.

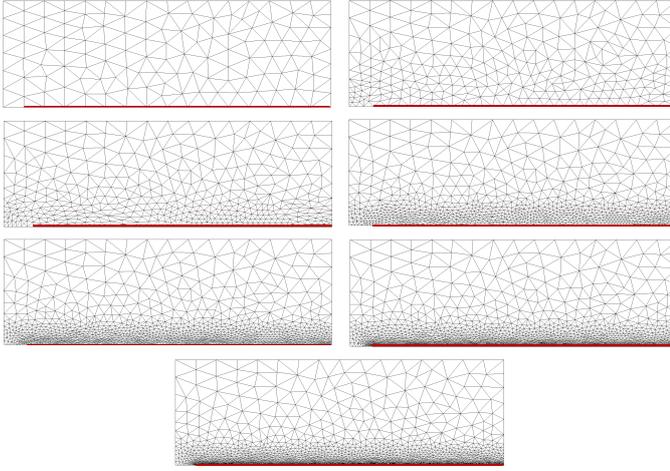


Fig. 9 Iterative procedure to generate a boundary layer mesh over a flat plate (red) from an arbitrary coarse mesh

The initial mesh consists of 140 elements and the final obtained mesh reaches 1882 elements. The minimum mesh size, the number of element in the mesh and their distribution over the domain for each mesh are described in Table 1.

Table 1 Distribution of elements for flat plate meshes

$h_{min} (m)$	Total # of elements	Box2	Boundary Layer
0.45	140	33 (23.6%)	17 (12.1%)
0.13	279	100 (35.8%)	68 (24.4%)
$6.4 \cdot 10^{-2}$	434	226 (52.1%)	164 (37.8%)
$3.2 \cdot 10^{-2}$	989	747 (75.5%)	486 (49.1%)
$1.6 \cdot 10^{-2}$	1310	1063 (81.1%)	745 (56.9%)
$8.1 \cdot 10^{-3}$	1663	1416 (85.1%)	1122 (67.5%)
$5.61 \cdot 10^{-3}$	1882	1637 (87.0%)	1330 (70.7%)

As expected, the obtained mesh respect well the computed metric. We note that most of the elements are localized in the boundary layer. The number of element in the Box2 zone is controlled and the mesh size in the far regions is coarse.

5.1.2 Circle

In this example, we consider an immersed circle of radius $0.3m$. The *a priori* mesh is generated for the study of a flow at a Reynolds number $Re = 10^5$ (turbulent), with $y_0^+ = 1$ and a characteristic length $L = 0.6m$ equals to the circle diameter. Such that, the minimum mesh size to reach is $h_{min} = 1.11 \cdot 10^{-4}m$. The maximum mesh size on Box2 is set to $H_2 = 0.05m$ and the maximum mesh size on the whole domain is set to $H_1 = 0.5m$. This leads to a final boundary layer composed of 24 layers with a maximum anisotropic ratio of 74.

As explained in Section 3.2.1, Figure 10 shows that if the element sizes of the initial mesh are greater than the features of the immersed geometry, then a pre-meshing procedure is applied. It allows to insert automatically additional nodes around the prescribed geometry and to better capture the interface, the zero isovalue of the levelset. The minimum mesh size reached in the pre-meshing procedure is equal to H_2 .

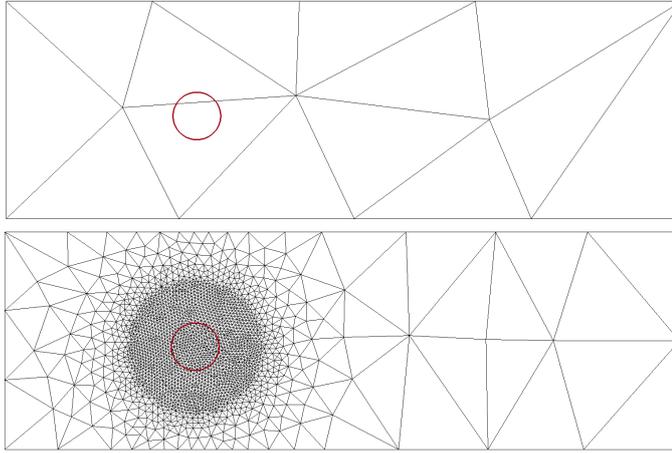


Fig. 10 Pre-meshing on a circle (right), represented in red, when the initial mesh is too coarse (left)

The final boundary layer mesh is presented in Figure 11. It contains 7023 elements where 93.7% of the mesh elements are localized inside Box2. The elements are distributed as follow: 3451 elements in the boundary layer which correspond to 49.1% of the total number of elements in the mesh, and 1593 elements inside the geometry which correspond to 22.7%. These elements are needed to properly fit the interface of the circle. Thus, there are 3 times more elements inside the geometry than between Box1 and Box2, nevertheless the mesh size is coarse inside the circle and the area is less extended.

Figure 12 presents a zoom on a quarter of the circle. The interface of the circle is colored in red while the limit of the boundary layer is represented in blue. As expected, the mesh elements respect very well the curvature with an appropriate mesh size, gradation and

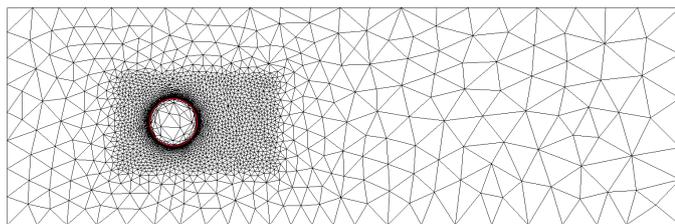


Fig. 11 Boundary layer mesh on a circle, the interface is represented in red

smooth transitions. We can clearly notice how the shape and the orientation of the elements match the directional features of the circle. The quality of the resulting mesh reflects again the potential of the proposed method to capture these layers.

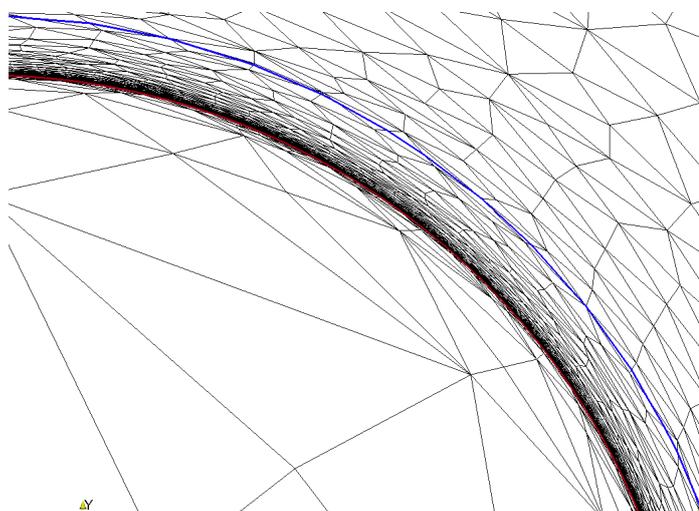


Fig. 12 Zoom on a quarter of the circle, the interface is represented in red and the limit of the boundary layer in blue

5.1.3 Square

We consider in this final elementary test case a square of $0.6 \times 0.6m^2$. The characteristics of the flow are the same as those of the circle test case. Thus, $h_{min} = 1.11 \cdot 10^{-4}m$, let us recall that $H_2 = 0.05m$ and $H_1 = 0.5m$. This leads to a final boundary layer composed of 24 layers with a maximum anisotropic ratio of 455. The final boundary layer mesh is represented in Figure 13, and the interface is colored in red.

The mesh is composed of 3562 elements, where 3113 elements are inside Box2 which represents 87.4% of the total mesh elements. In fact, Box2 is composed of 1096 elements inside the boundary layer (30.8% of the total elements) and of 395 elements inside the square

(11.1%). Using the same characteristics of the flow, we obtain in this case a totally different element size distribution. This highlights the interactions between the anisotropic ratio and the curvature. Indeed, a larger anisotropy was allowed on the square than on the circle which saves more elements.

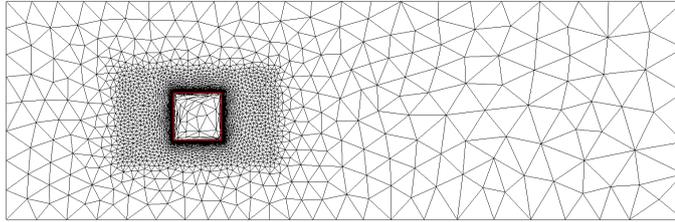


Fig. 13 Boundary layer mesh on a square, the interface is represented in red

Figure 14 shows a zoom on the corner where a sharp change in the curvature occurs. The angle is well captured and in fact, this kind of angles result in isotropic elements of mesh size h_{min} . Moreover, the first 6 levelsets from the multi-levelset method are represented in green, the interface of the square is represented in red while the inside layer is represented in orange. Note that the mesh elements respect well the layers prescribed by the multi-levelset method, which leads to a semi-structured mesh. Furthermore, the first three layers are well reproduced inside, between the interface and the inside layer.

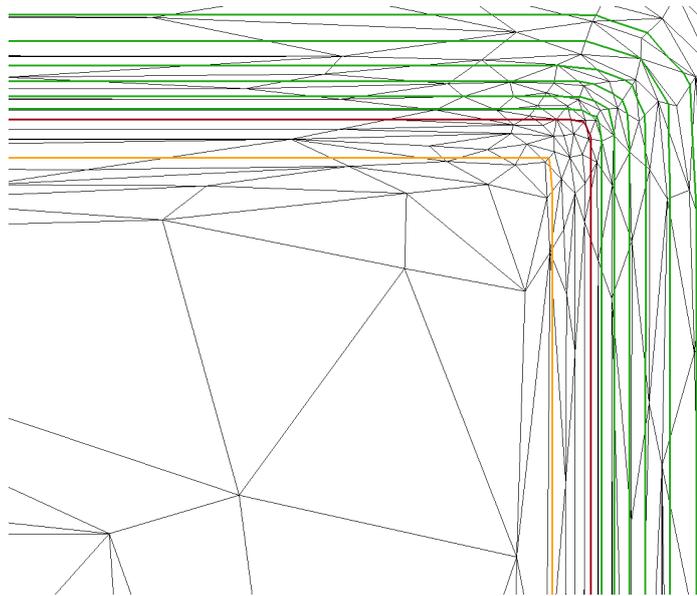


Fig. 14 Zoom on a square corner represented in red, first layers of the multi-levelset method are drawn in green and the inside layer is represented in orange

5.2 NACA0012

We consider next the well known NACA0012 benchmark [22]. This geometry is interesting because it is used to validate numerical codes and contains various curvature. We reproduced a smaller version of the NASA domain drawn in Figure 15 which couple a half-circle at inlet of radius $50m$ and a rectangle at outlet of size $47.1 \times 48.3m^2$. The domain is very large compared to the $1m$ characteristic length of the NACA0012 profile, and is usually used to avoid any perturbation for the boundary conditions.

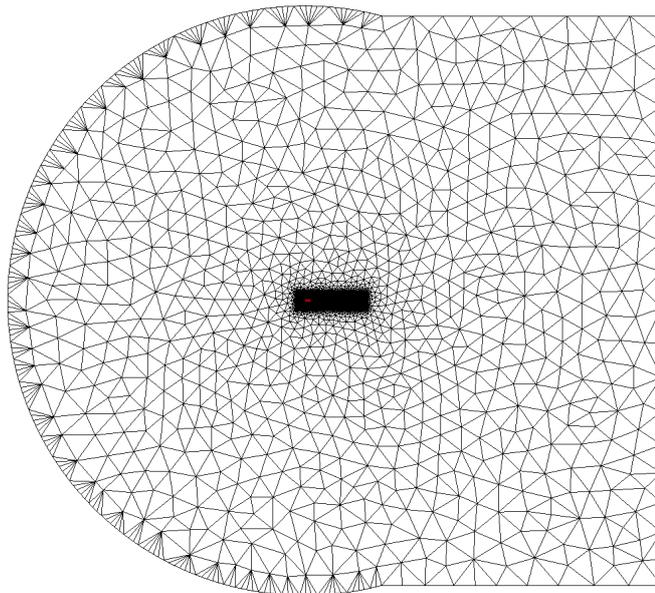


Fig. 15 Representation of the domain where the NACA0012 is immersed, its interface is drawn in red

Likewise the flow parameters are chosen equal to the NASA benchmark on NACA0012, the Reynolds number is set to $Re = 6 \cdot 10^6$ (turbulent), $y_0^+ = 1$, such that $h_{min} = 4.46 \cdot 10^{-6}m$. This leads to a final boundary layer composed of 36 layers with a maximum anisotropic ratio of 2387. A zoom on the final boundary layer mesh around the NACA0012 profile is represented in Figure 16, with in red the interface of the NACA0012. The mesh is composed of 21370 elements, whose 19804 elements are in Box2 which represents 92.7% of the total number of element. Moreover, the boundary layer is composed of 9647 elements (45.1%) and there are 3310 elements (15.5%) inside the geometry. Here again, we notice the structure in the mesh, the perfect orientation of elements and of their anisotropy. Furthermore, no problems are noticed by the sharp trailing edge. Again, the quality of the resulting mesh reflects the potential of the proposed boundary layer adaptation method to handle high Reynolds number flows.

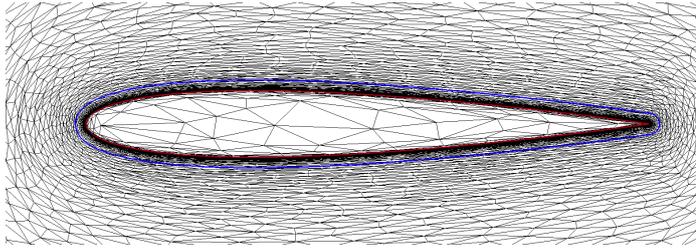


Fig. 16 Zoom around the interface of the NACA0012 (red), the limit of the boundary layer is plotted in blue

5.3 Ahmed body

This example aims at emphasizing the robustness of the proposed mesh adaptation technique to handle 3D geometries. Therefore, we consider to generate a boundary layer mesh around an immersed Ahmed body vehicle [1,9,26,8]. It is a simplified vehicle model generating high turbulent behavior according to the slant angle. Here, the study is made with a slant angle of 25° , the most turbulent angle. According to the references, the Ahmed body is immersed in a 3D channel of dimensions $10 \times 2 \times 1.5m^3$, the characteristic length of the Ahmed body is $L = 1m$, while the flow has a Reynolds number $Re = 4.25 \cdot 10^6$, obviously the flow is turbulent. In order to limit the number of elements, y_0^+ is chosen large, $y_0^+ = 5$. The minimum mesh size is set to $h_{min} = 3.1 \cdot 10^{-5}m$. This leads to a final boundary layer composed of 27 layers with a maximum anisotropic ratio of 3226.

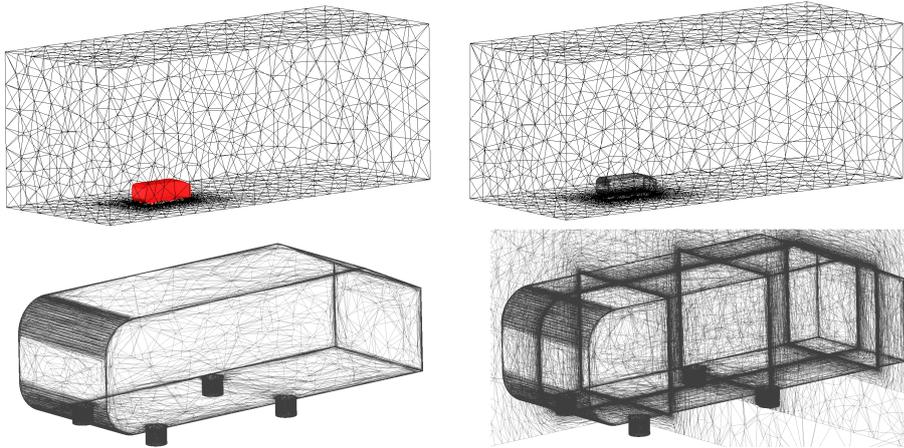


Fig. 17 The zero-isovalue of the levelset function and the boundary layer mesh for the immersed Ahmed vehicle

The obtained mesh is composed of 2167057 elements and 378259 nodes. Figure 17 gives a view of the mesh over the domain and around the zero-isovalue of the levelset function, which prove its ability to describe accurately the Ahmed body shape. Moreover, a zoom on the interface along the Y-Plane is given in figure 18. It shows how the elements are

highly stretched at the interface caused by the sharp gradient of the levelset function and the curvature. This reflects well the accuracy and the details of the geometry. As can be seen in these Figures, the anisotropic mesh adaptation proves to be very efficient to retrieve all the features of the geometry compared to isotropic meshes for the same amount of elements.

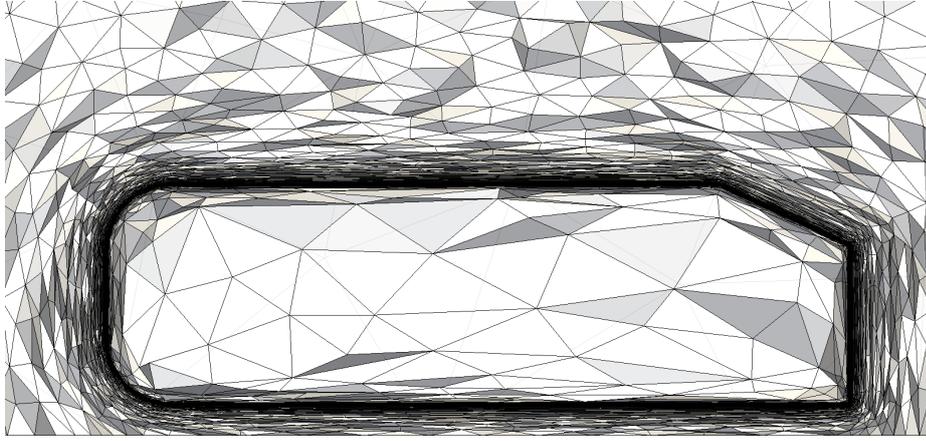


Fig. 18 Zoom around the interface of the immersed Ahmed body along the Y-plane cut

5.4 Towards complex geometries

To go further, we consider a very complex 3D geometry, a F1 car immersed inside a numerical wind channel. The mesh has been generated for a minimum mesh size of $h_{min} = 6 \cdot 10^{-4}m$. This leads to a final boundary layer composed of 10 layers with a maximum anisotropic ratio of 167. Recall that the air movement around a F1 car is quite complex and interesting; i.e. it allows the study of the influence of different airfoils and their positions to optimize the aerodynamic design. Therefore, an accurate boundary layer mesh is needed.

Figure 19 shows the obtained mesh for the immersed F1 car inside a wind channel. The mesh is composed of 21026520 elements and 3602483 nodes. Note the concentration of the resolution all along the boundary layers, in particular near very fine geometrical features such as the wheels. This reflects well the anisotropy caused by the sharp gradient of the levelset function as well as the curvature.

Taking a closer look at the mesh in Figure 20, we can detect the good orientation of the elements with the stretching in the relevant direction. This demonstrates the ability of the algorithm to generate unstructured accurate boundary layer meshes and to effectively control the elements sizes, orientations and locations.

Finally, we can clearly see in Figure 21 the quality of the interface rendered by the zero-isovalue of the levelset function. Such accuracy is required when using immersed techniques for high Reynolds number flows.

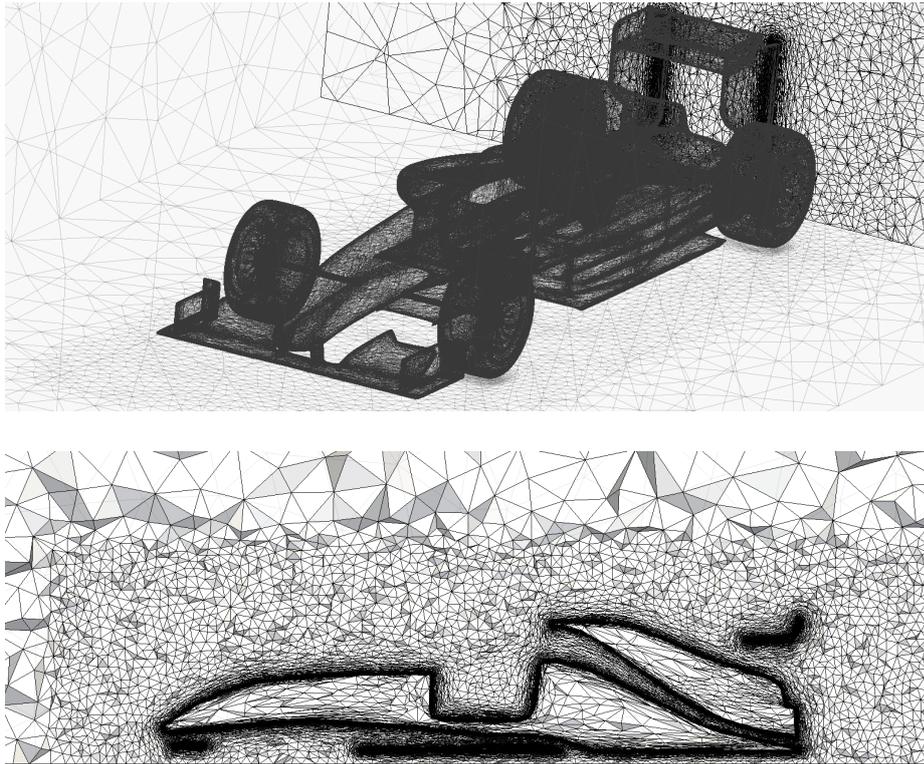


Fig. 19 Top: Boundary layer mesh for an immersed F1 car, Bottom: Zoom around the interface along the Y-plane cut

6 Conclusions and Future Work

We have presented a new anisotropic boundary layer mesh adaptation procedure for immersed geometry. It is based on the use of multi-levelset method to locate the boundary layer, to control the mesh size distribution and orientation ensuring a smooth gradation. Taking into account the physical parameters of the simulation and the curvature of the geometry, the numerical *2D* and *3D* applications show that, starting from an arbitrary coarse domain, the method provides very accurate representation of the immersed interfaces and their boundary layers even for complex geometries. The natural extension of this work is to consider a right angle computation in the wall vicinity [20, 15], and an improved computation of the aspect ratio [25].

Acknowledgements This work has been done in the MAIDESC ANR project which is supported by the french ministry of Research under contract ANR-13-MONU-0010.

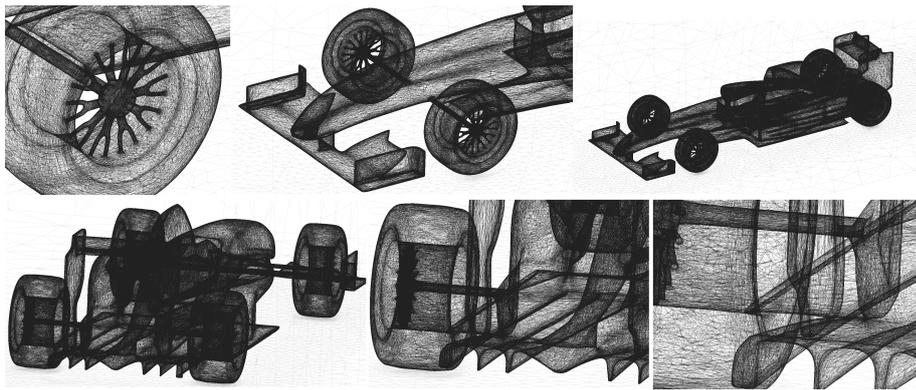


Fig. 20 Some cuts over the F1 car zero iso-value levelset boundary layer mesh

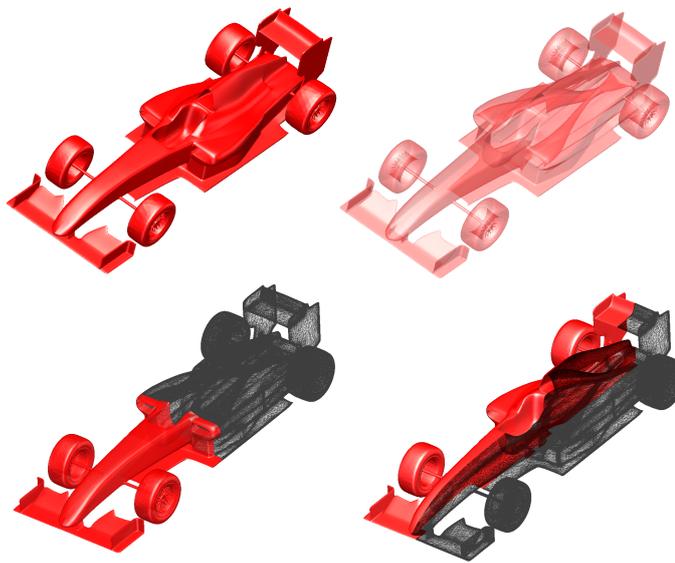


Fig. 21 The zero-iso-value of the levelset highlighting the complexity of the geometry

References

1. Ahmed, S., Ramm, G., Falin, G.: Some Salient Features Of The Time-Averaged Ground Vehicle Wake. SAE Technical Paper 840300 (1984). URL <http://papers.sae.org/840300/>
2. Borouchaki, H., George, P.L., Hecht, F., Laug, P., Saltel, E.: Delaunay mesh generation governed by metric specifications. Part I. Algorithms. *Finite Elements in Analysis and Design* **25**(1–2), 61–83 (1997). DOI 10.1016/S0168-874X(96)00057-1. URL <http://www.sciencedirect.com/science/article/pii/S0168874X96000571>
3. Borouchaki, H., George, P.L., Mohammadi, B.: Delaunay mesh generation governed by metric specifications Part II. Applications. *Finite Elements in Analysis and Design* **25**(1–2), 85–109 (1997). DOI 10.1016/S0168-874X(96)00065-0. URL <http://www.sciencedirect.com/science/article/pii/S0168874X96000650>

4. Bottasso, C.L., Detomi, D.: A Procedure for Tetrahedral Boundary Layer Mesh Generation. *Engineering with Computers* **18**(1), 66–79 (2002). DOI 10.1007/s003660200006. URL <http://link.springer.com/article/10.1007/s003660200006>
5. Chitale, K.C., Rasquin, M., Sahni, O., Shephard, M.S., Jansen, K.E.: Boundary Layer Adaptivity For Incompressible Turbulent Flows. arXiv:1405.0620 [physics] (2014). URL <http://arxiv.org/abs/1405.0620>. ArXiv: 1405.0620
6. Connell, S., Braaten, M.: Semi-structured mesh generation for 3d Navier-Stokes calculations. 12th Computational Fluid Dynamics Conference URL <http://arc.aiaa.org/doi/abs/10.2514/6.1995-1679>
7. Cousteix, J.: *Aérodynamique : turbulence et couche limite*. Cépaduès-editions (1989)
8. Fares, E.: Unsteady flow simulation of the Ahmed reference body using a lattice Boltzmann approach. *Computers & Fluids* **35**(8–9), 940–950 (2006). DOI 10.1016/j.compfluid.2005.04.011. URL <http://www.sciencedirect.com/science/article/pii/S0045793005001581>
9. Franck, G., Nigro, N., Storti, M.A., D'Elía, J.: Numerical simulation of the flow around the Ahmed vehicle model. *Latin American applied research* **39**(4), 295–306 (2009). URL http://www.scielo.org.ar/scielo.php?pid=S0327-07932009000400003script=sci_arttextlng=en
10. Garimella, R.V., Shephard, M.S.: Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering* **49**(1-2), 193–218 (2000). DOI 10.1002/1097-0207(20000910/20)49:1/2<193::AID-NME929>3.0.CO;2-R. URL [http://onlinelibrary.wiley.com/doi/10.1002/1097-0207\(20000910/20\)49:1/2<193::AID-NME929>3.0.CO;2-R/abstract](http://onlinelibrary.wiley.com/doi/10.1002/1097-0207(20000910/20)49:1/2<193::AID-NME929>3.0.CO;2-R/abstract)
11. Gruau, C., Coupez, T.: 3d tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering* **194**(48–49), 4951–4976 (2005). DOI 10.1016/j.cma.2004.11.020. URL <http://www.sciencedirect.com/science/article/pii/S0045782505000745>
12. Hachem, E., Feghali, S., Codina, R., Coupez, T.: Immersed stress method for fluid–structure interaction using anisotropic mesh adaptation. *International Journal for Numerical Methods in Engineering* **94**(9), 805–825 (2013). DOI 10.1002/nme.4481. URL <http://onlinelibrary.wiley.com/doi/10.1002/nme.4481/abstract>
13. Jones, M.W., Baerentzen, J.A., Sramek, M.: 3d Distance Fields: A Survey of Techniques and Applications. *IEEE Transactions on Visualization and Computer Graphics* **12**(4), 581–599 (2006). DOI 10.1109/TVCG.2006.56
14. Kallinderis, Y., Kavouklis, C.: A dynamic adaptation scheme for general 3-D hybrid meshes. *Computer Methods in Applied Mechanics and Engineering* **194**(48–49), 5019–5050 (2005). DOI 10.1016/j.cma.2004.11.023. URL <http://www.sciencedirect.com/science/article/pii/S0045782505000770>
15. Loseille, A.: Metric-orthogonal Anisotropic Mesh Generation. *Procedia Engineering* **82**, 403–415 (2014). DOI 10.1016/j.proeng.2014.10.400. URL <http://www.sciencedirect.com/science/article/pii/S1877705814016798>
16. Loseille, A., Alauzet, F.: Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error. *SIAM Journal on Numerical Analysis* **49**(1), 38–60 (2011). DOI 10.1137/090754078. URL <http://epubs.siam.org/doi/abs/10.1137/090754078>
17. Marcum, D., Alauzet, F.: Aligned Metric-based Anisotropic Solution Adaptive Mesh Generation. *Procedia Engineering* **82**, 428–444 (2014). DOI 10.1016/j.proeng.2014.10.402. URL <http://www.sciencedirect.com/science/article/pii/S1877705814016816>
18. Mavriplis, D.J.: An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness. *Journal of Computational Physics* **117**(1), 90–101 (1995). DOI 10.1006/jcph.1995.1047. URL <http://www.sciencedirect.com/science/article/pii/S0021999185710479>
19. Mesri, Y., Dignonnet, H., Coupez, T.: Advanced parallel computing in material forming with CIMLib. *European Journal of Computational Mechanics* **18**(7-8), 669–694 (2009). DOI 10.3166/ejcm.18.669-694. URL <http://www.tandfonline.com/doi/abs/10.3166/ejcm.18.669-694>
20. Mesri, Y., Guillard, H., Coupez, T.: Automatic coarsening of three dimensional anisotropic unstructured meshes for multigrid applications. *Applied Mathematics and Computation* **218**(21), 10,500–10,519 (2012). DOI 10.1016/j.amc.2012.04.014. URL <http://www.sciencedirect.com/science/article/pii/S0096300312003979>
21. Quan, D.L., Toulorge, T., Marchandise, E., Remacle, J.F., Bricteux, G.: Anisotropic mesh adaptation with optimal convergence for finite elements using embedded geometries. *Computer Methods in Applied Mechanics and Engineering* **268**, 65–81 (2014). DOI 10.1016/j.cma.2013.09.007. URL <http://www.sciencedirect.com/science/article/pii/S004578251300234X>
22. Rumsey, C.: 2d NACA 0012 Airfoil Validation for Turbulence Model Numerical Analysis (2015). URL http://turbmodels.larc.nasa.gov/naca0012numerics_al.html

23. [Sahni, O., Jansen, K.E., Shephard, M.S., Taylor, C.A., Beall, M.W.: Adaptive boundary layer meshing for viscous flow simulations. *Engineering with Computers* **24**\(3\), 267–285 \(2008\). DOI 10.1007/s00366-008-0095-0. URL <http://link.springer.com/article/10.1007/s00366-008-0095-0>](#)
24. [Schlichting, H.: *Boundary-layer theory*, mcgraw hill edn. \(1979\)](#)
25. [Shewchuk, J.R.: What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures. Tech. rep., In Proc. of the 11th International Meshing Roundtable \(2002\)](#)
26. [Strachan, R., Knowles, R., Lawson, N.: Comparisons between CFD and experimental results for a simplified car model in wall proximity. Tech. rep.](#)