TOPOGRAPHY ADAPTED MESH GENERATION FOR ATMOSPHERIC BOUNDARY LAYER FLOW SIMULATION

Abel Gargallo-Peiró Matias Avila Arnau Folch

Computer Applications in Science and Engineering Department
Barcelona Supercomputing Center
C/ Jordi Girona 29, 08034 Barcelona, Spain
{abel.gargallo,matias.avila,arnau.folch}@bsc.es

ABSTRACT

We present a new hybrid meshing procedure specifically designed to simulate Atmospheric Boundary Layer (ABL) flows featuring Coriolis effects for onshore applications. Two new meshing contributions for ABL flow simulation are proposed. First, a new metric-based adaptive meshing process to discretize the topography is presented. The adaptivity process is specifically written to deal with the discrete nature of topographic data, reparemeterizing smoothly the topography to allow computing the first and second order derivatives required by the desired metrics. Second, we present a new hybrid mesh generation procedure to discretize the ABL. We generate a prismatic boundary layer that captures the gradients of the Surface Boundary Layer, and following, an unstructured mesh is generated to discretize the rest of the domain. We show examples to illustrate the reduction in the element count with respect to standard semi-structured approaches, and we present the simulation result obtained on an onshore topographical scenario.

Keywords: Topography, Atmospheric Boundary Layer Flows, Hybrid Meshes, Mesh adaptation

1. INTRODUCTION

Simulation of Atmospheric Boundary Layer (ABL) flows is of interest to various scientific disciplines and related applications including, among others, meteorology, atmospheric transport of pollutants, or wind energy. In the ABL, orographic gradients, ground surface drag, and atmospheric thermal instabilities from radiative forcing, can generate turbulence and strong wind shear (vertical velocity gradients) in the so-called Surface Boundary Layer (SBL), which extends up to a $10 - 20\%$ of the total ABL depth [1, 2, 3, 4, 5]. The need to capture these high gradients impose requirements to the mesh generators employed in ABL simulation. The requirement to reproduce the high-gradients of the SBL has been translated in most mesh generators for ABL simulation in a fully structured mesh in the normal direction to the surface [6, 7, 8, 9, 10, 11].

In addition to the structure along the vertical direction, most mesh generators specifically designed to discretize the ABL [6, 7, 8, 9] are also semi-structured in the surface and do not feature adaptivity to the terrain. They feature hexahedra and use block structured strategies to provide a fine resolution in the zone of interest and less resolution far away from this region. However, since in each block the mesh is structured, inevitably finer resolution of the interest areas is extended to the rest of the domain. Although having the drawback of increasing the required number of elements of the mesh, this structured strategies have been exploited in offshore cases (analyzing the wind resource on the sea) or in topographic scenarios that do not feature high complexity or high gradients in or-

der to align the mesh with the wind inflow direction, generating one mesh for each simulated wind inflow direction.

Herein, we are interested in generating meshes in complex topographic scenarios, see for instance results in Section 6.2. In those scenarios, although when simulating without Coriolis there is a dominant wind direction, the complexity of the topography does not allow a priori to determine the best alignment of the mesh with the flow in the interest close-to surface region. In addition, herein we model the Coriolis effect in the flow, which translates in the fact that, even in offshore scenarios, the wind direction close to the ground is different than the wind direction in the top of the domain. The change of the wind direction determined by the Coriolis effect depends on the latitude of the scenario but in general can impose a change in the direction around 20 degrees from the ground to 2km over the topography. Thus, taking into account the Coriolis force, not even in offshore cases there is a unique wind direction, since it changes according to the height. Taking into account both that the wind direction changes at different heights and also due to the influence of the topography, in this work we propose to use an unstructured mesh on the surface.

Herein, we propose two new meshing contributions for Atmospheric Boundary Layer flow simulation. First, we propose an adaptive process to the topography, to have elements of the desired length on the topography and capture the curvature of the features present in the terrain. Second, we propose to generate the volume mesh in a different approach than the standard in simulation of ABL flows. First, we generate a prismatic boundary layer that captures the gradients of the Surface Boundary Layer, and following using an unstructured mesh to discretize the rest of the ABL. We take into account that away from the ground the vertical velocity gradient is small and the flow is almost isotropic. In addition, the influence of the topography is minor and a coarser mesh can capture the same flow features than the structured fine mesh that we have generated close to the topography to capture its geometrical features. Thus, tetrahedral elements deliver a geometrical flexibility that can be exploited to increase the element size in all the directions.

The rest of the paper is organized as follows. First, in Section 2 we present the previous works on the field of mesh generation for ABL flows. Second, in Section 3 it is presented the modeling of the discrete input topography and the two metrics used to adapt the surface mesh to the topography. Third, in Section 4 the surface mesh adaptation procedure is presented. Next, in Section 5 the combined structured and unstructured volume mesh generation process is detailed. Finally, in Section 6 different topographic scenarios are discretized with the presented methodology and the generated meshes are applied to simulate Atmospheric Boundary Layer flows.

2. RELATED WORK

The standard approaches for Atmospheric Boundary Layer simulation feature structured or semi-structured meshes. The use of structured grids was first used in Finite Difference and Finite Volumes applications, introducing the effect of the topography by means of a change of coordinates in the formulation of the problem instead of discretizing the topography with the mesh [12, 13]. Also in Finite Element and Finite Volume applications the use of structured hexahedral meshes represents the current most standard procedure. This semi-structured meshing strategy exploits the advantage of hexahedra to align the mesh with the flow on offshore applications and also exploits the tensor structure of hexahedra to reproduce the close-to-surface boundary layer. Several mesh generation approaches have based on solving a system of hyperbolic Partial Differential Equations (PDEs) to improve the orthogonality of the mesh and the desired element volume [14, 6, 7]. Similarly, the use of elliptic PDEs has been explored to determine the location of the mesh nodes [15]. In contrast with these approaches, in [9, 16] it is proposed a procedure that combines sweeping the quadrilateral surface mesh with a quality-based mesh optimization, to determine the best configuration of the nodes of the mesh according to the chosen quality measure.

There are several alternatives to the use of hexahedral elements. First, regarding the simulation of ABL flows, in [11] a fully prismatic mesh is generated to discertize the boundary layer. This work takes advantage of generating a triangle surface mesh to avoid extending the finer mesh size in the interest region to the rest of the domain and, simultaneously, takes advantage of the tensor direction of the prisms to discretize the boundary layer. As an alternative to structured or semi-structured approaches, different approaches featuring tetrahedra have been followed for problems that require a mesh conformal with the topography but that do not require a boundary layer close to the surface [17, 18, 19, 20, 21, 22, 23].

In this work, we aim to develop an specific-purpose approach to deal with ABL flows featuring Coriolis effects on complex topographies. In contrast with previous works, we aim to generate an hybrid prismatic and tetrahedral mesh that features the main advantages of both structured and unstructured meshes for our target CFD model. First, in the surface mesh generation process, we exploit the flexibility of the well-established triangle meshers to generate a triangle surface mesh adapted to the topography. Second, as pre-

viously highlighted, in the first $10-20\%$ of the volume, a boundary layer is required. Thus, we take advantage of the tensor structure of prismatic elements to discertize the ABL with the desired growing ratio. Finally, once in a region with smaller vertical gradients, we take advantage of the flexibility of tetrahedra to discertize the domain with the desired element size, and not limited by the structure of the mesh in the first region of the domain close to the surface.

3. TOPOGRAPHY GEOMETRY MODELING

In this section, it is detailed the strategy followed herein to model the geometry data for topographic landscapes. Topographic geometries are discrete due to the existent data extraction processes. Thus, first we present the approach to obtain a smooth geometry representation that allows computing first and second order derivatives of the underlying topography, see Section 3.1. Following, in Section 3.2, the two metrics used in this work to adapt to the geometry are presented, particularizing them for the case of topographic geometries.

3.1 High-order approximation of the topography

This work is devoted to generate meshes conformal with the terrain. The geometry corresponds to real measured data that can be given in many formats, such as contour topography maps, Cartesian grids or point clouds. Herein, all the input frameworks are converted into a triangle mesh that is used as a geometry representation. Following, we define the proposed parameterization of the target surface (topography). Due to the existent data extraction procedures, topographic data for wind resource assessment in this work always ensures that each point in the plane has a unique height value. This is so since the triangle mesh that represents the geometry can be interpreted as a planar triangle mesh with a height value in each of the points of the mesh. In particular, we define Π_Σ as the parametric plane, the region in $\mathbb{R}^2$ where the topography is defined (region of the plane where the topography data has been measured). Thus, given a point $\mathbf{x} \in \Pi_\Sigma$ the height function of this point $\mathrm{z}_h(\mathbf{x})$ on the topography is defined as

$$\begin{array}{rccc} \mathrm{z}_h: & \Pi_\Sigma \subset \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & \mathbf{x} = (x,y)^t & \longmapsto & \mathrm{z}_h(\mathbf{x}), \end{array} \tag{1}$$

where $\mathrm{z}_h(\mathbf{x})$ is computed by finding the triangle to which this point belongs and interpolating the height value in the triangle. The function z_h unequivocally determines a parameterization $\boldsymbol{\varphi}_h$ of the topography surface Σ as

$$\begin{array}{rccc} \boldsymbol{\varphi}_h: & \Pi_\Sigma \subset \mathbb{R}^2 & \longrightarrow & \Sigma \subset \mathbb{R}^3 \\ & \mathbf{x} = (x,y)^t & \longmapsto & \mathbf{z} = (x,y,\mathrm{z}_h(\mathbf{x}))^t. \end{array} \tag{2}$$

This parameterization maps a point in the parametric plane to a point in the topography. As highlighted, it is a discrete parameterization that finds the surface triangle to which the point belongs, and computes the exact location of this point in the topography. To compute for instance the curvature of the target surface (Section 3.2) in the mesh adaptation process (Section 4), it is required to perform queries of the first and second order derivatives of the geometry. However, since topography representation is a piece-wise linear triangle geometry, the derivatives of the topography geometry are not well defined. On the one hand, there are discontinuities of the first order derivative in the edges between the elements of the topography geometry. On the other hand, since the geometry mesh features piece-wise linear elements, the curvature of each triangle configuring the geometry is null. Thus, similarly to [24, 25], herein we propose to reconstruct local high-order approximations of the geometry that allow queries of the derivatives of the surface representation.

To generate the high-order approximation in a point $\mathbf{x} \in \Pi_\Sigma$, first it is located the triangle that contains this point in the mesh that defines the geometry. Following, several layers of elements adjacent to the container elements are gathered. Each new layer is computed as the triangles neighboring to the current considered elements. Then, we compute the least squares approximation of the desired order of the cloud of points determined by the neighborhood of elements around the target point. Specifically, given a set of topography points $\{\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}\}$ the high-order approximation used in this work is written as

$$\mathrm{z}_p(x,y) = \sum_{\{i,j\}\in I_p} a_{ij} x^i y^j, \tag{3}$$

where p is the desired polynomial degree, $I_p = \{\{i,j\} \mid i,j \geq 0 \text{ and } i+j \leq p\}$ is the set of indexes lower or equal to p, and $\{a_{ij}\}_{\{i,j\}\in I_p}$ are the coefficients of the polynomial $\mathrm{z}_p(x,y)$ on x and y. In particular, we seek the approximation that better fits the cloud of points in the least squares sense:

$$\{a_{ij}\}_{\{i,j\}\in I_p} = \underset{\substack{a_{ij}\in\mathbb{R} \\ \{i,j\}\in I_p}}{\operatorname{argmin}} \sum_{k=1}^{n_p} \left(\mathrm{z}_p(x_k,y_k) - z_k\right)^2, \tag{4}$$

where $\mathbf{z}_k = (x_k, y_k, z_k)$ are the cloud point coordinates.

Note that if there are less points than parameters a_{ij} the problem stated in Eq. (4) is ill-conditioned. Thus, by default p levels of elements around the container element are considered. Once all the neighboring points

are gathered, they are counted to ensure that the number of points in the cloud is higher or equal than the number of parameters. Otherwise, an extra layer of neighboring elements is included, repeating this process until the number of points ensures a well-posed minimization problem. In particular, if we have the exact same number of parameters we would obtain a surface containing all the points of the cloud.

Once computed the smooth high-order approximation, the new parameterization $\boldsymbol{\varphi}_p$ is defined as

$$\begin{array}{rccc} \boldsymbol{\varphi}_p: & \Pi_\Sigma \subset \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \\ & \mathbf{x} = (x,y)^t & \longmapsto & \mathbf{z} = (x, y, \mathrm{z}_p(x,y))^t. \end{array} \tag{5}$$

Note that this parameterization has well defined first and second order derivatives that will allow to compute the desired metrics on the geometry in Section 3.2. The order of approximation used in this work in the presented examples is three, and, accordingly to what has been previously stated, the number of levels of adjacency to compute the approximation is also three (without requiring in any example to automatically computing extra layers of neighbors).

3.2 Metrics over the topography geometry

The main objective of this work is to generate meshes that feature the desired edge length on the surface and that at the same reproduce the curvature of the geometry. Thus, the edge lengths of the surface mesh will be measured taking into account two different metrics: the metric of the tangent space, and the metric derived from the Hessian of the parameterization.

First, some notation is stated regarding the measurement of edge lengths in surfaces with respect to a given metric. Following Section 3.1, we denote a general parameterization of a surface Σ as:

$$\begin{aligned} \boldsymbol{\varphi}: \Pi_\Sigma \subset \mathbb{R}^2 &\longrightarrow \Sigma \subset \mathbb{R}^3 \\ \mathbf{x} = (x,y)^t &\longmapsto \mathbf{z} = \boldsymbol{\varphi}(\mathbf{x}). \end{aligned} \tag{6}$$

We define the curve $\boldsymbol{\gamma}(t)$ between two points $\mathbf{z}_1$ and $\mathbf{z}_2$ on Σ in terms of the parameterization $\boldsymbol{\varphi}$ and the edge $[\mathbf{x}_1, \mathbf{x}_2]$ on the parametric space as

$$\begin{aligned} \boldsymbol{\gamma}: [0,1] &\longrightarrow \Gamma \subset \Sigma \subset \mathbb{R}^3 \\ t &\longmapsto \boldsymbol{\varphi}(\mathbf{x}(t)), \end{aligned} \tag{7}$$

where $\mathbf{x}(t)$ is defined as

$$\begin{aligned} \mathbf{x}: [0,1] &\longrightarrow [\mathbf{x}_1, \mathbf{x}_2] \subset \Pi_\Sigma \subset \mathbb{R}^2 \\ t &\longmapsto \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1), \end{aligned} \tag{8}$$

and with $\boldsymbol{\gamma}$ fulfilling that $\boldsymbol{\gamma}(0) = \mathbf{z}_1$ and $\boldsymbol{\gamma}(1) = \mathbf{z}_2$. The length of the curve on the surface is

$$l(\mathbf{z}_1, \mathbf{z}_2) = \int_0^1 \sqrt{< \frac{\partial \boldsymbol{\gamma}}{\partial t}(t), \frac{\partial \boldsymbol{\gamma}}{\partial t}(t) >}\, \mathrm{d}t, \tag{9}$$

where $< \mathbf{a}, \mathbf{b} >= \mathbf{a} \cdot \mathbf{b}$ is the scalar product between the vectors $\mathbf{a}$ and $\mathbf{b}$.

Next, following the ideas in [26], the chain rule is used in the definition of the curve in Eq. (7) and Eq. (8):

$$\frac{\partial \boldsymbol{\gamma}}{\partial t} = \begin{pmatrix} \frac{\partial \boldsymbol{\varphi}}{\partial x} & \frac{\partial \boldsymbol{\varphi}}{\partial y} \end{pmatrix} \cdot \frac{\partial \mathbf{x}(t)}{\partial t} = \nabla \boldsymbol{\varphi} \cdot (\mathbf{x}_2 - \mathbf{x}_1),$$

and rewrite the length of the curve in Eq. (9) in terms of the parametric coordinates of the two surface nodes:

$$\begin{aligned} & l_{\mathcal{M}_1}(\mathbf{z}_1, \mathbf{z}_2) = \\ & = \int_0^1 \left((\mathbf{x}_2 - \mathbf{x}_1)^t \cdot \nabla \boldsymbol{\varphi}^t \cdot \nabla \boldsymbol{\varphi} \cdot (\mathbf{x}_2 - \mathbf{x}_1) \right)^{1/2} \, \mathrm{d}t \\ & = \int_0^1 \left((\mathbf{x}_2 - \mathbf{x}_1)^t \, \mathcal{M}_1(\mathbf{x}(t)) \, (\mathbf{x}_2 - \mathbf{x}_1) \right)^{1/2} \, \mathrm{d}t, \end{aligned} \tag{10}$$

where

$$\begin{aligned} & \mathcal{M}_1(\mathbf{x}(t)) \equiv \mathcal{M}_1 = \\ & \quad = \nabla \boldsymbol{\varphi}^t \cdot \nabla \boldsymbol{\varphi} = \begin{pmatrix} \frac{\partial \boldsymbol{\varphi}}{\partial x} & \frac{\partial \boldsymbol{\varphi}}{\partial y} \end{pmatrix}^t \cdot \begin{pmatrix} \frac{\partial \boldsymbol{\varphi}}{\partial x} & \frac{\partial \boldsymbol{\varphi}}{\partial y} \end{pmatrix}, \end{aligned} \tag{11}$$

is the matrix expression of the first fundamental form of the surface Σ at the point on the parametric space $\mathbf{x}(t)$, see Eq. (8). In particular, for isotropic mesh generation with a desired length h of an edge known in each region of the domain, we define the tangent metric in terms of $\mathcal{M}_1$ as

$$\mathcal{M}_T := \frac{1}{h^2} \mathcal{M}_1, \tag{12}$$

and the corresponding length measure as

$$\begin{aligned} & l_{\mathcal{M}_T}(\mathbf{z}_1, \mathbf{z}_2) := \\ & \int_0^1 \left((\mathbf{x}_2 - \mathbf{x}_1)^t \cdot \mathcal{M}_T(\mathbf{x}(t)) \cdot (\mathbf{x}_2 - \mathbf{x}_1) \right)^{1/2} \, \mathrm{d}t. \end{aligned} \tag{13}$$

Note that we would ideally like the edges of the mesh to have measure 1 with the metric $\mathcal{M}_T$. In this manner, in the adaptive procedure that will be presented in Section 4, Eq. (13) will be used to compute the length of the curve on the surface in terms of the coordinates of the nodes in the parametric space. Next, these elements with measure greater than one (with a safety factor) will be refined until all edges fulfill the desired metric.

In addition to measuring the edge length of the mesh taking into account the length of the curve described in the surface, we also want to capture the curvature of the surface. That is, the second order derivatives of the surface will be also taken into account in the adaptive procedure. To do so, we explicitly exploit that the parameterization $\boldsymbol{\varphi}$ in Eq. (6) for topographic geometries can be rewritten as $\boldsymbol{\varphi}_p(x,y) = (x, y, \mathrm{z}_p(x,y))$, as detailed in Eq. (3) and Eq. (5) from Section 3.1. Thus, the parameterization can be also understood as

a field z_p over a 2D mesh on x and y. Hence, herein we propose to adapt the mesh to capture the curvature of the geometry using techniques for 2D mesh adaptation to reduce the interpolation error of $z_p(x, y)$.

Following the ideas presented in [27, 28], from the Hessian $\mathcal{H}$ of the topography at a point,

$$\mathcal{H} = \begin{pmatrix} \frac{\partial^2 z_p}{\partial x \partial x} & \frac{\partial^2 z_p}{\partial x \partial y} \\ \frac{\partial^2 z_p}{\partial y \partial x} & \frac{\partial^2 z_p}{\partial y \partial y} \end{pmatrix},$$

the following metric is defined:

$$\mathcal{M}_C = \mathbf{V}\ \mathbf{D}\ \mathbf{V}^t, \tag{14}$$

where $\mathbf{V} = (\mathbf{e_1}, \mathbf{e_2})$ is the matrix composed by the eigenvectors $\mathbf{e_1}$ and $\mathbf{e_2}$ of $\mathcal{H}$, $\mathbf{D} = \beta\ \mathrm{diag}(|\lambda_1|, |\lambda_2|)$ a diagonal matrix with its absolute value of the eigenvalues, and β a curvature discretization parameter. Herein, β has a default value of 1, which has been determined experimentally for the average metrics that arise in topographic applications after testing several topographic scenarios. However, it can also be specified by the user to discretize the curvature of the geometry with the desired accuracy, or computed according to [29, 30, 31] to provide an estimate of the number of vertices in terms of the metric complexity. Similarly to Eq. (13), the length of the curve between $\mathbf{z}_1$ and $\mathbf{z}_2$ with respect to the metric $\mathcal{M}_C$ is defined as

$$l_{\mathcal{M}_C}(\mathbf{z}_1, \mathbf{z}_2) := \int_0^1 \left((\mathbf{x}_2 - \mathbf{x}_1)^t \cdot \mathcal{M}_C(\mathbf{x}(t)) \cdot (\mathbf{x}_2 - \mathbf{x}_1) \right)^{1/2}\ \mathrm{d}t, \tag{15}$$

where $\mathbf{x}(t)$ is defined in Eq. (8). In particular, if the user has prescribed an element size of h, the edges of a mesh should ideally fulfill $l_{\mathcal{M}_T} = 1$ and $l_{\mathcal{M}_C} = 1$. However, these two conditions may not be achievable simultaniously since, for instance, the curvature of the geometry may demand $l_{\mathcal{M}_T} << 1$. Following, in Section 4 it is detailed the adaptive mesh generation proposed in this work, based on the two presented metrics. First, it will be generated a coarse topography mesh, which will be locally refined it until no edges of the mesh have length greater than 1, or if the length of the edges is lower than a minimum value set by the user. With these two conditions, the edge length of the surface mesh is controlled and it is ensured that the mesh accurately reproduces the curvature of the geometry.

4. SURFACE MESH: UNSTRUCTURED TOPOGRAPHY MESH GENERATION

In this section, we detail our surface mesh adaptation procedure for topographic geometries. The topography is parameterized according to the technique proposed in Section 3.1, and the edge lengths of the mesh are measured according to the tangent and curvature metrics, Eq. (13) and Eq. (15) respectively, both detailed in Section 3.2.

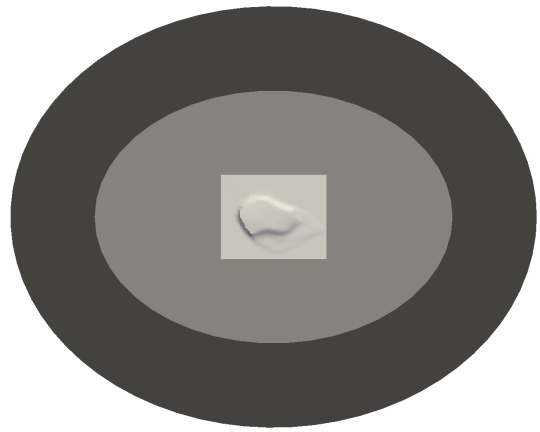


Figure 1: Synthetic example illustrating the domain regions: farm (light gray), transition (gray) and buffer (dark grey).

In this work, the topography is divided into three regions, with three different levels of resolution illustrated in Figure 1: the interest wind farm area (higher resolution, light gray), a transition area (gray), and an elliptical buffer area to impose the boundary conditions (lower resolution, dark gray). First, the farm area is meshed. The farm is a quadrilateral domain featuring the region of interest in the simulation (for instance, area where a wind farm is to be designed). The adaptive process to the topography is applied in the farm region, where high geometric accuracy is required to discretize the features of the topography. The transition area is an elliptical domain that encircles the farm region and is meshed with a triangle mesh that smoothly matches the fine mesh of the transition with the element size of the buffer, the outer region. Finally, it is defined an additional elliptic region with a coarse element size to impose the boundary conditions. Herein, the outer boundary is chosen to be elliptical to avoid the discontinuities that the corners of a quadrilateral domain can induce in ABL simulations with Coriolis and to impose in a continuous manner the input/output flow conditions.

4.1 Adaptive mesh generation process

To generate a surface mesh adapted to the topographic features, two inputs are required: the maximum element size h_{max} and the minimum allowed size h_{min}. First, a planar triangle mesh with the maximum element size is generated using the Triangle mesh generator [32]. In Figure 2(a) the initial mesh with constant

Algorithm 1 Surface mesh generation process adapted to the topography

Input: Topography surface Σ, Maximum edge length h_{max}, Minimum edge length h_{min}
Output: Mesh M

```
1: function MeshTopography(Σ,h_max,h_min)
2:     M ← generate planar tri mesh of size h_max
3:     elemsToRefine← findElemsToRefine(M,Σ)
4:     while elemsToRefine≠ ∅ do
5:         M ← refinePlanarMesh(M,elemsToRefine)
6:         elemsToRefine←findElemsToRefine(M,Σ)
7:     end while
8:     M ← mapToSurface(M,Σ)
9:     M ← optimizeMeshOnSurface(M,Σ)
10:    return M
11: end function
```

element size is illustrated. This mesh constitutes the starting point for the adaptive procedure. Following, the edge lengths of all the elements of the mesh are measured using the tangent and curvature metrics. If an edge of an element is greater than 1, this implies that the edge is longer than desired with the metric. The condition to accept the element as it is or refine it is relaxed to those elements with lengths greater than $\sqrt{2}$, see [24, 26]. Thus, this element is included in a list of elements to be refined.

Once all the mesh elements have been checked, we locally refine the mesh where it is required. The refinement is performed by using the previous mesh as background mesh and asking to those elements included in the refinement list to have half of the size of the triangle. This process is repeated until all the elements have length below the desired threshold for both metrics, or if the minimum edge length is below the desired minimum length. We highlight that the minimum length is checked using the euclidean metric to control the minimum length in the mesh for the simulation.

The iterative procedure to perform the mesh refinement is detailed in Algorithm 1. In particular, while there are elements to refine, Line 4, the mesh is refined as previously detailed. Every time that the mesh is refined, the edge lengths with respect the two metrics are computed. Once the edge lengths are computed, the elements with edges not fulfilling the desired edge lengths are listed again to be refined, Line 6. However, to control the minimum allowed element size and to avoid obtaining excessively refined meshes, those elements with length under the euclidian metric higher than h_{min} are not included again in the refinement process. The process concludes when no elements violate any metric, or when the elements violating the metrics have already achieved the minimum allowed edge length. At this point, the nodes of the planar triangle mesh are mapped to the topography in Line 8, discretizing it.

In Figure 2 the adaptive process is applied to generate a mesh on the Bolund peninsula geometry. In this illustrative example, the input mesh sizes are $h_{max} =$ 5m and $h_{min} = 0.5$m. First, in Figure 2(a) a planar mesh of constant element size is generated, Line 2 in Algorithm 1. Then Figures 2(b) to 2(h) illustrate the successive refinements of the mesh according to Line 5. Finally, in Figure 2(i) the final surface mesh is illustrated. In the procedure, the initial number of elements is 5089, and the final one is 9037.

Once the adaptive process is finalized, it is obtained a mesh that has elements of the desired size on the surface, and that reproduces the curvature of the geometry up to the minimum allowed mesh size. Up to this point local mesh refinement has been performed in order to improve the accuracy of the geometric approximation. However, in this process the quality of the generated mesh for simulation has still not been assessed. Thus, to conclude the generation of the topography surface mesh, in Line 9 a quality optimization of the surface elements is performed. Following, in Section 4.2 we detail the quality measure that will be used in this work to validate the generated meshes, and the current mesh configuration will be improved by means of reallocating the nodes on the surface to improve the mesh quality.

4.2 Surface mesh optimization

The final step of the generation of the surface mesh is an optimization of the location of the mesh nodes on the exact topography to obtain a mesh which minimizes the elemental distortion (maximizes the quality). For each element on the surface, its ideal is defined as an equilateral triangle of the desired size. The optimization procedure targets that each surface element reassembles the ideal as much as possible. It has to be taken into account that this ideal configuration can not in general be achieved since the mesh topology is now fixed and since the surface elements have their nodes tied to the geometry.

To measure if an element is valid, and to quantify how much it differs from the desired configuration, we use a distortion measure (see Knupp [33] for a review of measures). A distortion measure quantifies in the range $[1, \infty)$ the deviation of an element with respect to an ideal configuration (for instance, the equilateral triangle with the desired size for the triangle case). In this work, the distortion of an element with nodes $\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}$ is denoted as $\eta(\mathbf{z}_1, \ldots, \mathbf{z}_{n_p})$. The distortion takes value 1 when the element presents the desired configuration, and tends to infinity when the element degenerates. Following the ideas for high-order elements in [34, 35], the distortion measure used in

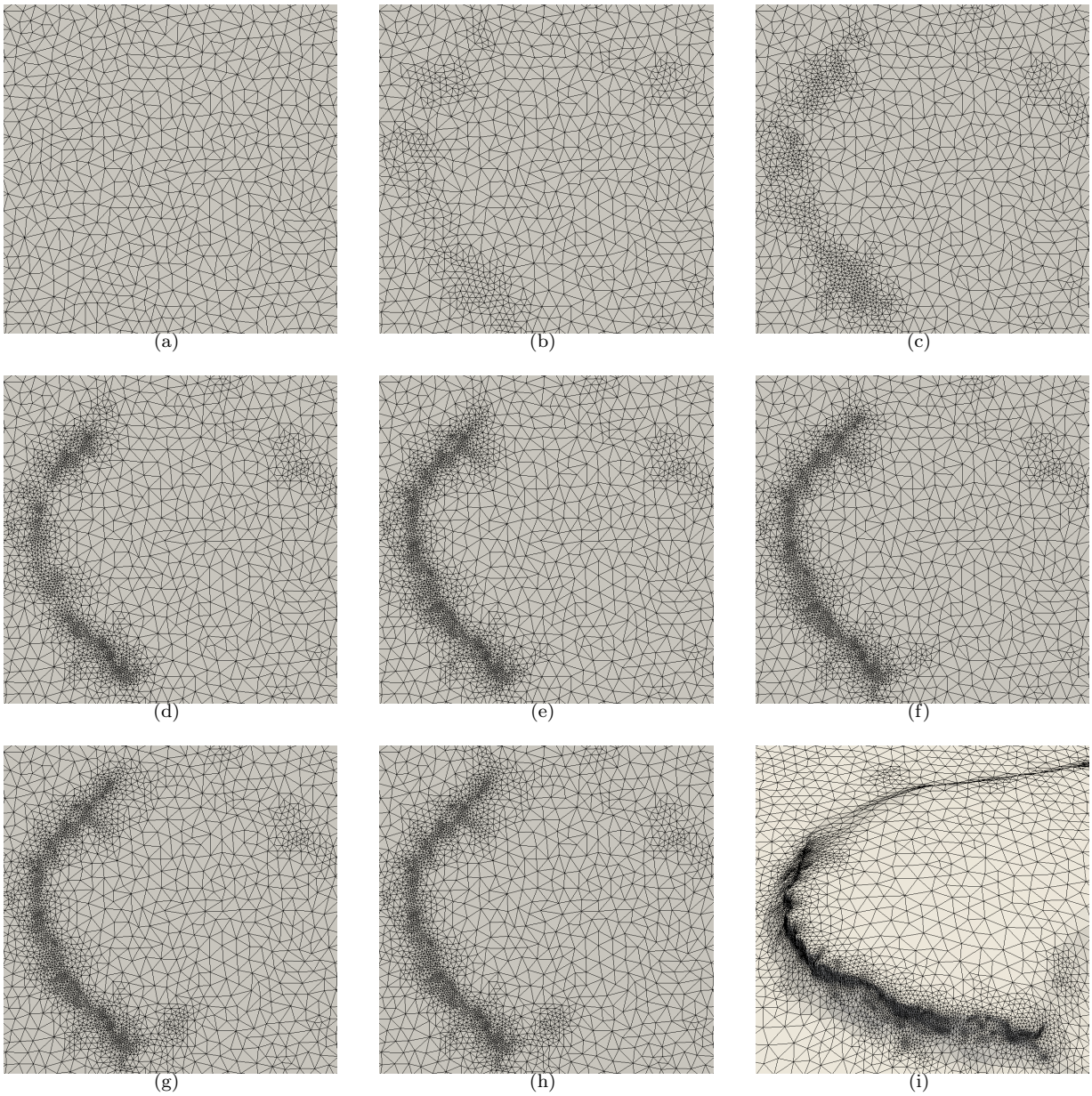


Figure 2: (a)-(h) Mesh adaptive process for the Bolund geometry illustrated in Figure 1. (i) Final adapted topography surface mesh.

this work can be written for any given element with nodes $\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}$ as:

$$\eta(\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}) := \frac{\|\eta_{sh}(\mathbf{D}\boldsymbol{\phi})\|_{E^I}}{\|1\|_{E^I}}, \tag{16}$$

where η_{sh} is defined in [33] as

$$\eta_{sh}(\mathbf{D}\boldsymbol{\phi}) := \frac{\|\mathbf{D}\boldsymbol{\phi}\|_F^2}{d \ \mid \det \mathbf{D}\boldsymbol{\phi} \mid^{2/d}},$$

where $\boldsymbol{\phi}(\mathbf{z}_1, \ldots, \mathbf{z}_{n_p})$ is the mapping between the ideal E^I and physical elements, $\mathbf{D}\boldsymbol{\phi}$ is its Jacobian, $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_{E^I}$ is the $\mathcal{L}^2$ norm on the ideal element, $\|1\|_{E^I}$ is the measure of the ideal element, and d is the dimension of the element ($d = 2$ for planar and surface meshes, and $d = 3$ for volumetric meshes). In particular, the quality of an element is the inverse of the distortion:

$$q := \frac{1}{\eta} \in [0, 1], \tag{17}$$

which gives us a value in $[0, 1]$, being 0 an invalid configuration, and 1 the desired one. For the surface case, we consider as ideal the equilateral triangle with the desired size. In Figure 3(a) we show the initial surface mesh, coloring the elements with respect to their quality. It can be observed that non-regular lower quality elements are present in areas with big slopes of the topography.

To optimize the nodes on the topography [36, 37], we use the surface parameterization to rewrite the distortion in Eq. (16) for an element E with nodes $\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}$, in terms of the parametric coordinates of the nodes $\mathbf{x}_1, \ldots, \mathbf{x}_{n_p}$ as:

$$\begin{aligned}\eta^{\boldsymbol{\varphi}}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_p}) :=& \eta(\boldsymbol{\varphi}(\mathbf{x}_1), \ldots, \boldsymbol{\varphi}(\mathbf{x}_{n_p})) \\ =& \eta(\mathbf{z}_1, \ldots, \mathbf{z}_{n_p}).\end{aligned}$$

In the optimization process it is found the location of the nodes on the parametric plane Π_Σ such that provide minimal elemental distortion (maximum quality) of the surface elements in the least squares sense. In particular, it is sought $\{\mathbf{x}_1^*, \ldots, \mathbf{x}_{n_{N_s}}^*\} \subset \Pi_\Sigma$ such that:

$$\{\mathbf{x}_1^*, \ldots, \mathbf{x}_{n_{N_s}}^*\} = \underset{\mathbf{x}_1, \ldots, \mathbf{x}_{n_{N_s}} \in \Pi_\Sigma}{\operatorname{argmin}} \sum_{e=1}^{n_{E_s}} \left(\eta^{\boldsymbol{\varphi}}(\mathbf{x}_{e_1}, \ldots, \mathbf{x}_{e_{n_p}})\right)^2, \tag{18}$$

where n_{N_s} is the number of surface nodes, n_{E_s} is the number of surface elements, e_i denotes the global node id of the ith node of element e, and $\eta_e^{\boldsymbol{\varphi}}$ denotes the distortion of element e. We highlight that to deal with inverted elements ($\det \mathbf{D}\boldsymbol{\phi} \leq 0$), and specially to untangle meshes in the optimization procedure, the regularization of the determinant $\det \mathbf{D}\boldsymbol{\phi}$ presented in [38, 39] is used. In particular, $\det \mathbf{D}\boldsymbol{\phi}$ in Eq. (16) is replaced by $r(\det \mathbf{D}\boldsymbol{\phi})$, where

$$r(x) = \frac{1}{2}\left(x + \sqrt{x^2 + 4\delta^2}\right), \tag{19}$$

where δ is a numerical parameter that we determine following the approach presented in [40].

Once the optimal location in the parametric space $\{\mathbf{x}_1^*, \ldots, \mathbf{x}_{n_{N_s}}^*\}$ is obtained, the nodes are mapped back to the topography by means of the parameterization as $\{\mathbf{z}_1^*, \ldots, \mathbf{z}_{n_{N_s}}^*\} = \{\boldsymbol{\varphi}(\mathbf{x}_1^*), \ldots, \boldsymbol{\varphi}(\mathbf{x}_{n_{N_s}}^*)\}$. For the optimization process, we use the input piece-wise linear parameterization in order to ensure that the final location of the mesh nodes is on the input topography.

The topography mesh generation process for the Bolund hill (Denmark) is shown in Figure 3. In addition, in Table 1 we illustrate the mesh quality statistics resulting from the detailed procedure. Figure 3(a) shows the initial distorted surface mesh, which has a minimum quality of 0.17. In contrast, Figure 3(b) shows the optimized mesh, with a minimum quality that has been increased to 0.50, and where it can be observed that the distorted elements from the initial mesh have become almost regular all over the domain. We highlight that the quality of the surface mesh is of major importance for the volume mesh generation. The surface mesh defines the boundary of the volume mesh, and therefore, an invalid (or low-quality) surface mesh will derive in a low-quality (or invalid) volume mesh.

Table 1: Shape quality statistics (minimum, maximum, mean and standard deviation) for the triangle meshes presented in Figure 3.

Mesh	Min.Q.	Max.Q.	Mean Q.	Std
Fig. 3(a)	0.17	1.00	0.89	0.12
Fig. 3(b)	0.50	1.00	0.96	0.05

5. VOLUME MESH: ATMOSPHERIC BOUNDARY LAYER MESH GENERATION

The Atmospheric Boundary Layer mesh is generated in a domain enclosed by the topography, a planar ceiling at the desired height (user input defaulted as 2km over the highest topography point), and an elliptic lateral wall. The elliptic lateral boundary is defined by extruding vertically the boundary of the 2D domain, see Figure 1, up to the desired height.

This volumetric domain is meshed following Algorithm 2. First, in Line 2, the triangle surface mesh of the topography is swept (extruded) to generate a structured prismatic mesh close to the ground. This process is de-

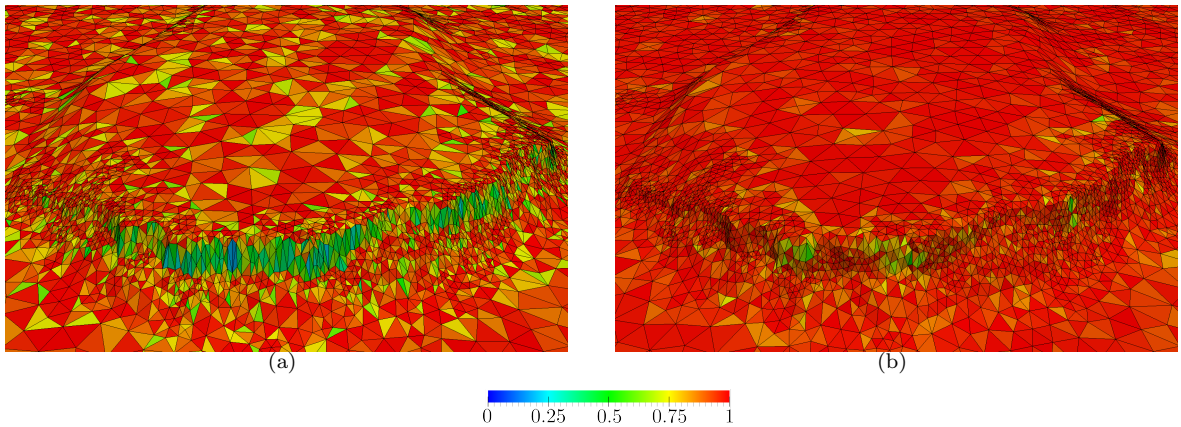


Figure 3: Topography mesh generation for the Bolund hill: (a) initial distorted surface mesh, and (b) optimized high-quality surface mesh. The surface elements elements are colored according to their quality, Eq. (17).

tailed in Section 5.1. As input for the prismatic meshing process, it is required to provide the initial height of the elements (h_0), the growing ratio (r), a maximum elemental height for the boundary layer (h_1), and the height until which the boundary layer extends (z_{BL}). Herein, the growth factor of the boundary layer is chosen in the interval $[1.05, 1.25]$, and the anisotropy in the first layer is set between the order of 1/100 or 1/1000, depending on the CFD case and the region of the domain. In addition, the maximum height of the structured boundary layer region is set by default at the 20% of the total ABL (400 meters over 2km), see [1, 2, 3, 4, 5]. If the elements become isotropic before reaching the height of the structured region, the sweeping is continued keeping constant the size along the vertical direction.

Once the prismatic mesh is finished, in Line 3 an unstructured tetrahedral mesh is generated to fill the rest of the domain with tetrahedra: from the last prismatic layer to the flat ceiling located at height z_{top}. The generation of the tetrahedral mesh relies on TetGen [41], to which a surface mesh is provided together with an element size field. The element size of the triangles from the prismatic mesh are assigned with their own size, while the elements in the top ceiling are assigned with the top ceiling size given by the user (h_2). The elements on the elliptical lateral boundary are assigned with a smooth linear size gradation from the prismatic mesh to the top ceiling height. The prismatic layer together with the tetrahedral mesh compose an hybrid mesh, Line 4, and following, the resulting hybrid mesh is optimized, see Line 5. The generation of the unstructured mesh and the optimization of the hybrid mesh are detailed in Section 5.2.

Algorithm 2 Generation of the hybrid ABL mesh

Input: Surface mesh M_Σ, Initial Size SBL h_0, Final Size SBL h_1, Growing Ratio r, Height SBL z_{BL}, Top Ceiling z_{top}, Top Ceiling Size h_2
Output: Mesh M

```
1: function MeshABL(M_Σ,h_0,h_1,r,z_BL,z_top)
2:     M_pri ← MeshSBL(M_Σ,h_0,h_1,r,z_BL)
3:     M_tet ← GenerateTetMesh(M_pri,z_top,h_2)
4:     M ← GenerateHybridMesh(M_pri,M_tet)
5:     M ← OptimizeQuality(M)
6: end function
```

5.1 Structured prismatic meshing using an optimization-based sweeping

Given a initial triangle surface mesh generated with the approach presented in Section 4, the volume mesh of the close to the surface region is generated by means of an iterative sweeping procedure that requires two main steps to compute each new sweeping layer. First, given a layer of triangle elements, a new layer of triangles is generated. These two triangle meshes determine the new prismatic layer by connecting each node on the initial layer with the corresponding swept node in the extruded layer. Second, once a new layer of prisms has been generated, it is performed a local optimization of the low quality elements that have been generated up to this point.

Given a layer of triangle elements, we generate a new layer of prisms by means of sweeping each node computing a new extruding length and an extruding direction. The current extrusion length is computed in a standard manner using a geometrical law based on a user input growing ratio, herein defaulted as 1.15. As input for the user it is also asked a maximum element size (defaulted as the isotropic configuration of

the swept element). Once this size is reached or if the element becomes isotropic, the extruding length is kept constant and not further increased with the growing ratio.

To determine the extruding direction, we compute the pseudo-normal [42, 43] of the nodes adjacent to each swept node. In particular, given a node $\mathbf{z}$ with n_l neighboring nodes $\{\mathbf{z}_1, \ldots, \mathbf{z}_{n_l}\}$, the pseudo-normal $\mathbf{n_z}$ is defined as

$$\begin{aligned}\mathbf{n_z} :=& \frac{\sum_{i=1}^{n_l} \mathbf{z}_i \times \mathbf{z}_{i+1}}{\|\sum_{i=1}^{n_l} \mathbf{z}_i \times \mathbf{z}_{i+1}\|} \\ =& \frac{\sum_{i=1}^{n_l} (\mathbf{z}_i - \mathbf{z}) \times (\mathbf{z}_{i+1} - \mathbf{z})}{\|\sum_{i=1}^{n_l} (\mathbf{z}_i - \mathbf{z}) \times (\mathbf{z}_{i+1} - \mathbf{z})\|},\end{aligned}$$

where we consider $\mathbf{z}_{n_l+1} \equiv \mathbf{z}_1$. The pseudo-normal is used to maximize the orthogonality of the extruded layer with respect to the previous one. The main property of the pseudo-normal $\mathbf{n_z}$ is that it defines the plane that maximizes the area of the projection of the polygon defined by $\{\mathbf{z}_1, \ldots, \mathbf{z}_{n_l}\}$, see [42, 43], reducing the chances of generating invalid extruded elements. We highlight that the pseudo-normal is blended with the vertical direction in order to enforce that the mesh grows towards the ceiling and that it gets to the top orthogonally to the planar ceiling.

It must be remarked that although the use of the pseudo-normal reduces the possiblity of obtaining inverted or low quality elements, this is still not guaranteed in the extruding process. Thus, it is required to assess the quality of the mesh during the procedure to ensure its validity and, if possible, improve its quality with a similar approach to the one used for the surface mesh in Section 4.2. To compute the quality of a given prismatic element, an ideal element is required. We set the ideal of each prism in terms of the best prism that we would get if no topography was present. That is, the ideal element of a given physical prism is defined as the initial surface element extruded orthogonally with the computed extrusion length at the current layer. This is indeed the element that we would desire to generate, since it is extruded orthogonally and has the best configuration on the surface (it has been optimized for the target topography in Section 4.2). Once we have set the ideal prism, we use Eq. (16) to compute the distortion of the elements and assess its validity.

Once a new prismatic layer is generated, it is optimized to improve its configuration with respect to the defined ideal elements. Recall that although it has been extruded using the best projection plane, the constraint of coming from a non-planar mesh prevents the mesh from being optimal. In particular, only a mesh with a planar topography would be optimal with respect to the ideal configuration. Therefore, before generating a new layer of elements, the current layer is optimized. Since the mesh is still being generated and taking into account the computational efficiency of the process, during the sweeping process we only optimize the lowest quality elements on the mesh. The quality threshold for this local optimization is defaulted in our applications as 0.2. To allow those elements to have room for improvement, several layers of neighboring nodes are included in the optimization process. The rest of nodes are kept fixed and are not optimized. In all the examples of this work, two levels of neighboring nodes are considered. In particular, following a Gauss-Seidel approach, we solve for each free node the non-linear minimization problem:

$$\mathbf{z}_i = \underset{\mathbf{z}_i \in \mathbb{R}^3}{\operatorname{argmin}} \sum_{e=1}^{n_E^i} \eta_{\hat{e}}^2(\mathbf{z}_{e_1}, \ldots, \mathbf{z}_{e_{n_p}}), \qquad (20)$$

where $\{\hat{e}\}_{e=1,\ldots,n_E^i}$ denotes the set of adjacent elements to node $\mathbf{z}_i$, $\hat{e}$ denotes the global id of the e-th adjacent element of node $\mathbf{z}_i$, $\eta_{\hat{e}}$ is the distortion of the e-th element, and e_i corresponds to the global node id of the ith node of the eth neighbor element.

Table 2: Shape quality statistics (minimum, maximum, mean and standard deviation) for the meshes presented in Figure 4.

Mesh	Min.Q.	Max.Q.	Mean Q.	Std
Fig. 4(a)	0.10	1.00	0.98	0.10
Fig. 4(b)	0.10	1.00	0.85	0.23
Fig. 4(c)	0.20	1.00	0.86	0.19

5.2 Unstructured mesh generation and hybrid mesh optimization

Once the sweeping process is finalized, the unstructured tetrahedral mesher TetGen [41] is used to generate a volume mesh that fills the rest of the domain up to the desired height. To do so, the target volume is defined in terms of a planar ceiling, an elliptical surface that encloses all the perimeter of the initial surface mesh, and the top boundary of the prismatic mesh. Each triangle of the boundary that encloses the volume is assigned with an element size field that determines the size of the tetrahedral mesh. In particular, the triangles from the swept prismatic mesh are assigned a size that ensures a smooth matching between the prismatic and the tetrahedral mesh. The triangles of the planar top ceiling are assigned the element size chosen by the user. Finally, the triangles from the elliptic lateral are assigned an element size computed using a linear blend between the size at the swept surface and the size at the planar top ceiling.

Once all the volume mesh has been generated, only the prismatic elements have been optimized in terms of the

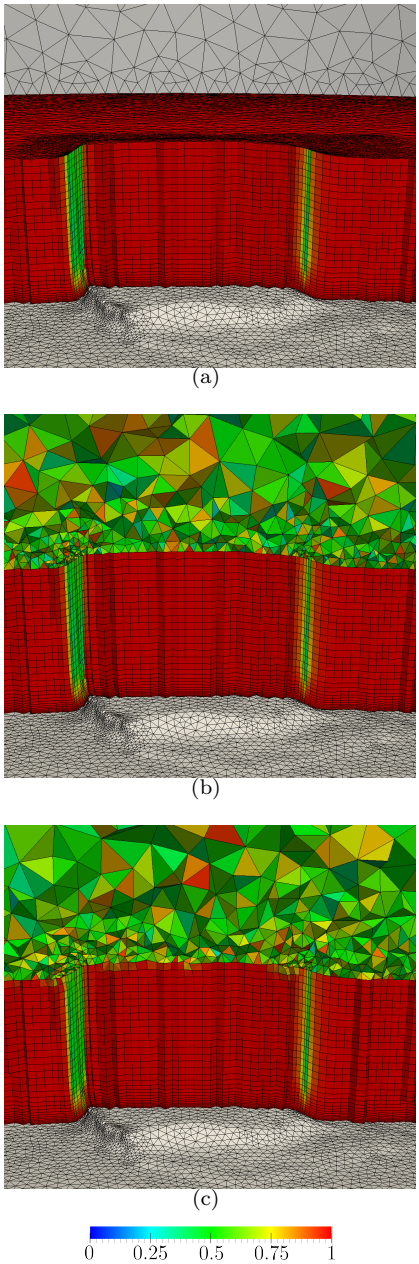


Figure 4: Main meshing steps to generate the mesh of the Atmospheric Boundary Layer for the Bolund hill: (a) prismatic mesh, (b) hybrid prismatic-tetrahedral mesh, and (c) globally optimized hybrid mesh. The elements are colored according to their quality.

quality presented in Eq. (17), and only locally in the sweeping process to improve the previous configuration before a new surface sweep. Therefore, the mesh is still not globally optimal in terms of the defined quality (distortion) measure. Thus, after the topology of the final mesh has been set, we perform a final mesh optimization to compute the coordinates of the mesh nodes that deliver minimal distortion. To do so, we optimize all the nodes except the topography surface ones. The nodes on the topography are kept fixed since they are already optimal in the parameterized topography geometry. The volume nodes are free to move in $\mathbb{R}^3$, whereas the rest of boundary nodes are allowed to move on their boundary surfaces.

In particular, we find the coordinates of the nodes that do not belong to the topography that minimize the mesh distortion in the least-squares sense [34], by optimizing in a Gauss-Seidel approach the objective function

$$f(\mathbf{z}_1, \ldots, \mathbf{z}_n) = \sum_{e=1}^{n_E} \eta_e^2(\mathbf{z}_{e_1}, \ldots, \mathbf{z}_{e_{n_p}}), \qquad (21)$$

where n_N is the number of nodes of the mesh, n_E is the number of elements in the mesh, η_e is the distortion of the e-th element, and e_i corresponds to the global node id of the ith node of element e.

Figure 4 illustrates the generated hybrid volume mesh from the topography surface mesh illustrated in Figure 3(b). Figure 4(a) shows the first step of the process, the prismatic mesh obtaining by sweeping the topography surface mesh. Figure 4(b) shows the hybrid mesh obtained after meshing with tetrahedra the domain between the top triangles mesh from the prismatic sweep mesh and the top ceiling mesh. Finally in Figure 4(c) we illustrate the final hybrid mesh obtained after the performed quality optimization. We highlight that the domain has been designed to clearly illustrate both the topography adaptation, the prismatic mesh and the unstructured mesh in the same image. That is, the sizes of the farm, transition and buffer areas, and the region featuring the boundary layer are not realistic for an ABL flow simulation. A complete Bolund scenario is meshed in Section 6.1. Table 2 details the quality of the meshes presented in Figure 4. The minimum quality of the final mesh, Figure 4(c), is 0.2. The minimum is obtained in prismatic mesh in the region extruded close to the almost vertical wall from the geometry, where the mesh has been more refined. We highlight that without the final mesh optimization the minimum quality of the mesh is 0.1, presenting a significant quality improvement due to the presented optimization process. In addition, if no volumetric optimization is present during the extruding procedure (Sec. 5.1), the mesh becomes invalid during the sweeping process and the mesh cannot be generated. This issue stresses the importance of the quality optimization framework, specially in regions where the topography features high gradients.

6. RESULTS

In this section, we present two examples to illustrate the capabilities of the presented hybrid mesh generation approach adapted to the topography. First, in Section 6.1 it is performed an analysis of the meshes obtained using the hybrid process in contrast with the standard semi-structured hexahedral meshers for ABL flows. In particular, the same mesh sizing parameters are used to analyze the main differences between the semi-structured and the unstructured approaches. Second, in Section 6.2 the meshing process is illustrated for the topography encircling the Badaia wind farm, located in Spain.

In Section 6.2, the generated mesh is used to solve the Reynolds Averaged Navier-Stokes (RANS) equations with a k-ε turbulence model adapted to the Atmospheric Boundary Layer [44] and featuring Coriolis effects. The model was presented and developed in [10, 45] and implemented in the finite-element multiphysics parallel solver Alya [46, 47]. However, the proposed meshing algorithm has been implemented in the external model-independent pre-process code WindMesh [10, 9, 16, 48]. As a result, meshes generated with this utility can be used to simulate both with RANS or Large-Eddy Simulation (LES) turbulence models and, in addition, are also valid for solvers based on other numerical methods such as Finite Volumes. The simulation illustrated in Figure 8 has been run in the supercomputer MareNostrum4 [49] using 512 cores.

For the different examples, we detail the computational time of the mesh generation process performed on a MacBook Pro with one dual-core Intel Core i7 CPU, a clock frequency of 3.0 GHz, and a total memory of 16 GBytes. In Section 6.1, the computational time to generate the semi-structured mesh in Figure 5(a) is 9 seconds, and the time to generate the hybrid mesh in Figure 5(b) is 17s. In Section 6.2, the CPU time to generate the hybrid mesh in Figure 7 is 246 seconds.

6.1 Bolund peninsula

The objective of this work is to propose a new mesher to simulate ABL flows featuring Coriollis effects in complex topographies. In this context, the motivation of this section is to present a simple example to illustrate the advantages of using the proposed approach against a standard semi-structured ABL mesher in terms of the resulting element count and the flexibility to attain the desired element sizes in the domain. We highlight that standard meshers are devoted for problems without Coriollis. In those cases there is a dominant large-scale wind direction at the top ceiling, and this is exploited by means of generating semi-structured hexahedral meshes aligned with this inflow direction. However, in cases featuring complex topographies, the direction of the wind close to the ground is not dominated by a particular direction. As previously highlighted, this problem is increased when Coriollis effects are present, where even without topography the wind direction changes depending on the height with respect to the ground. Thus, in contrast with standard meshers, we have developed a specific-purpose mesher for complex topographies and simulations featuring Coriollis effects.

The comparison of this section is performed with the aim to illustrate the advantages of the meshing approach proposed in this work for the specific-purpose for which it has been developed. One of the objectives of the hybrid mesher is to reduce the degrees of freedom required to discretize a complex topography and the Atmospheric Boundary Layer in simulations featuring Coriollis effects. Hence, we will measure the reduction of degrees of freedom of the proposed mesher in contrast with the standard approach [6, 7, 8, 9, 16, 10, 11]. However, we highlight that semi-structured meshers have the advantage of being aligned with the wind inflow direction in problems without Coriollis effects or without complex topographies.

To perform this comparison, we use our in-house semi-structured code WindMesh [9, 16], that generates a semi-structured cross-type mesh, see Figure 5(a), which constitutes the standard in the field of ABL flow simulation [6, 7, 8, 9, 10, 11, 16]. As previously highlighted, these semi-structured meshers are devoted to generate an hexahedral mesh aligned with the inflow direction. Thus a structured quadrilateral mesh is generated in the interest regions, and then an elliptical domain is meshed using a semi-structured approach increasing the mesh size towards the exterior of the domain. Next, this mesh is extruded to discretize the boundary layer, see [9, 16] for further details.

Figure 5 illustrates the obtained meshes. Figure 5(a) shows the generated semi-structured mesh, while Figure 5(b) we illustrate the hybrid mesh generated with exactly the same parameters than the semi-structured one. That is, the imposed mesh sizing is exactly the same. In this particular test case, featuring a small domain with a small topographic scenario around Bolund hill (a small peninsula in Denmark), the input sizes are: 5 meters in the farm region and 25 in the buffer region. Regarding the vertical discretization, the initial cell size is 0.5 and the growing ratio is 1.15. We highlight that in this example, we do not adapt the hybrid mesh to the topography, to perform a fair comparison with the semi-structured approach. However, the adaptive procedure can also be exploited to locate the degrees of freedom where they are really required

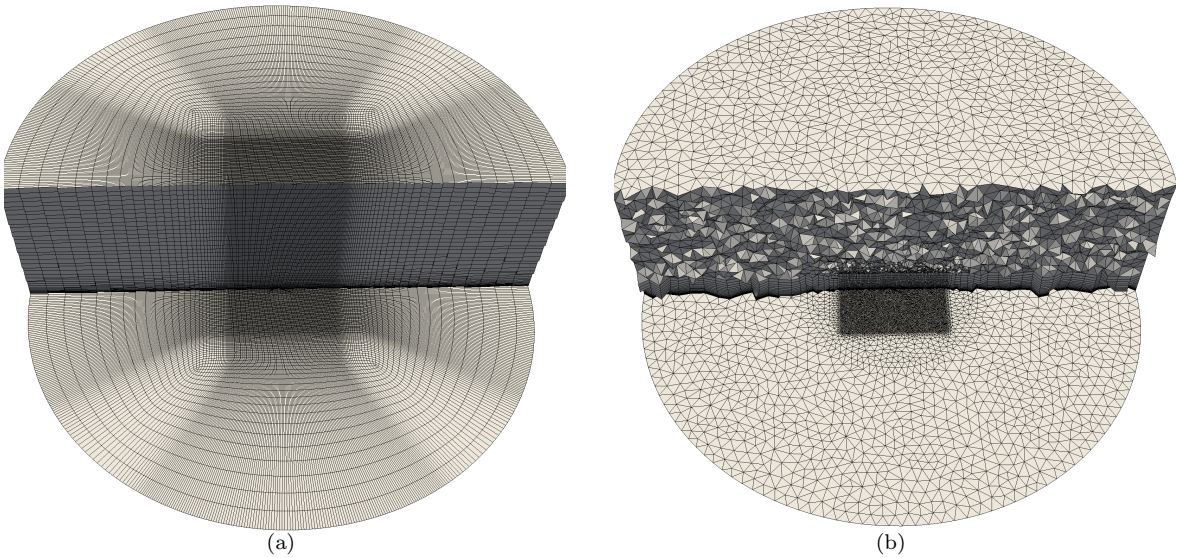


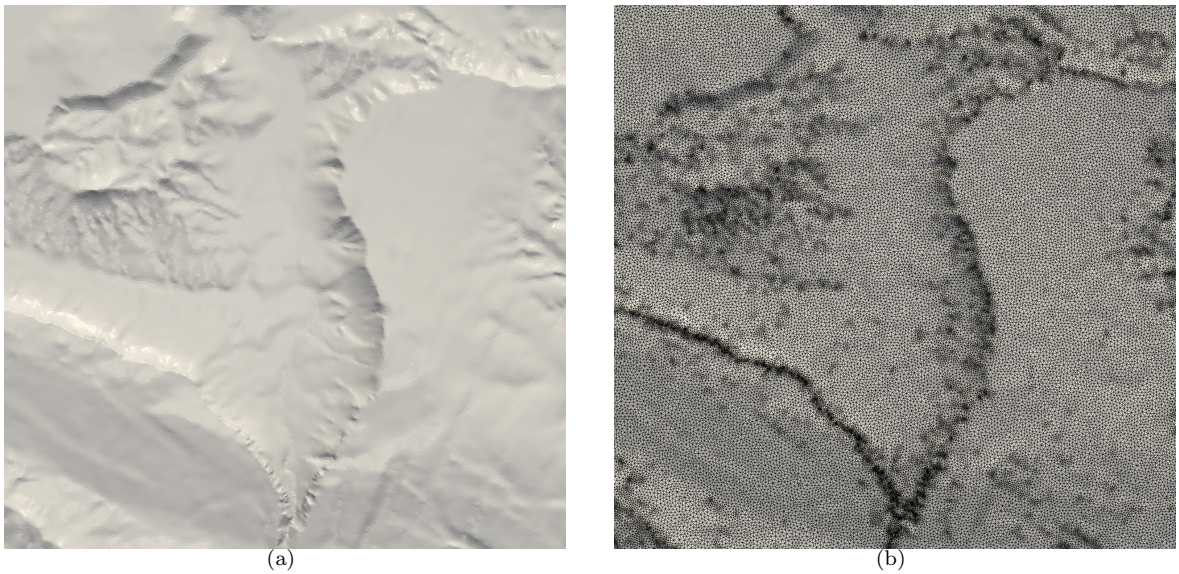
Figure 5: Meshes generated on Bolund hill: (a) standard semi-structured mesh and (b) hybrid mesh.

and reduce even more the total element count.

Regarding the number of elements and nodes of the generated meshes, the hexahedral mesh in Figure 5(a) is composed by 1407780 (1.41M) nodes and 1362756 (1.36M) elements. In contrast, the meshes in Figure 5(b) is composed by 268698 (0.27M) nodes and 767444 (0.77M) elements, from which 405680 (0.41M) are prisms and 361764 (0.36M) tetrahedra. In particular, we obtain a mesh with half of the elements and a fifth of the number of nodes but attaining the same required resolution than in the structured approach. Recall that the computational cost of a given simulation depends on the number of nodes, since the nodes determine the number of unknowns to solve. Thus, a fifth of the number of nodes is a significant reduction of the computational cost and of the size of the matrices involved in the solvers, which have a dimension of the order of the number of nodes. However, we have the additional advantage of being able to adapt the mesh to ensure that the element size on the surface is the desired and to capture the curvature of the geometry.

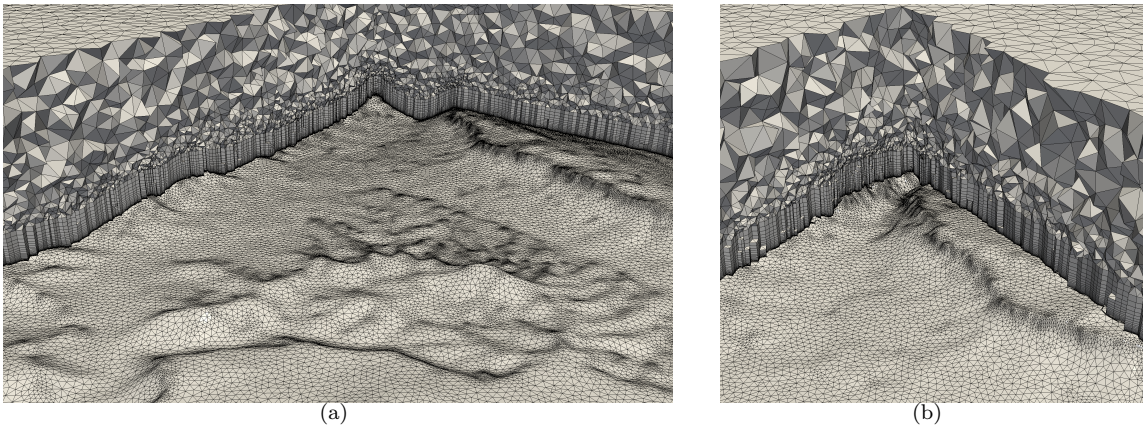
The difference in the element count derives from the flexibility of the new approach to attain the mesh sizes prescribed by the user in the different regions. In contrast, the semi-structured approach extends the high-resolution areas towards the rest of the domain, both in the surface mesh and in the volume mesh when extruding the boundary layer. It must be highlighted again that, in applications with a dominant wind direction and not introducing the Coriolis effect on the RANS equations, the semi-structured mesh presents a clear advantage, which is the alignment of the elements with the flow direction. However, we are interested on onshore problems featuring Coriolis, and thus the combination of the effect of complex topographic scenarios and a wind direction that changes according to the height reduces the advantages from a semi-structured approach for our objectives. Thus, in our application, without a unique wind direction, the hybrid approach reduces to one fifth the degrees of freedom of the simulation and simultaneously we reduce the complexity of generating a different mesh for each inflow direction.

Alternatively to the unstructured hybrid approach presented in this work, an unstructured quadrilateral surface mesh and an unstructured hexahedral mesh could also be an alternative to the standard semi-structured approach. However, the existence of mature Delaunay-based meshers such as Triangle [32] and TetGen [41], has allowed us in this work to focus on dealing with the input topography data, the geometry re-parameterization, the adaptation process and the boundary layer meshing. Together with the lack of a unique wind direction, we have favored a fully unstructured triangle approach on the surface that is adapted to the topography features, relying on existent mature technology provided by Triangle. Regarding the volume mesh, we have exploited the vertical structure of the SBL and generated first a prismatic region. Finally, we have generated an unstructured tetrahedaral mesh to allow a size gradation in all the directions out of the SBL, relying on the mature technology provided by TetGen.



(a) (b)

Figure 6: (a) Badaia topography, and (b) adapted surface mesh.



(a) (b)

Figure 7: ABL hybrid mesh on Badaia topography: (a) overview and (b) close-up.

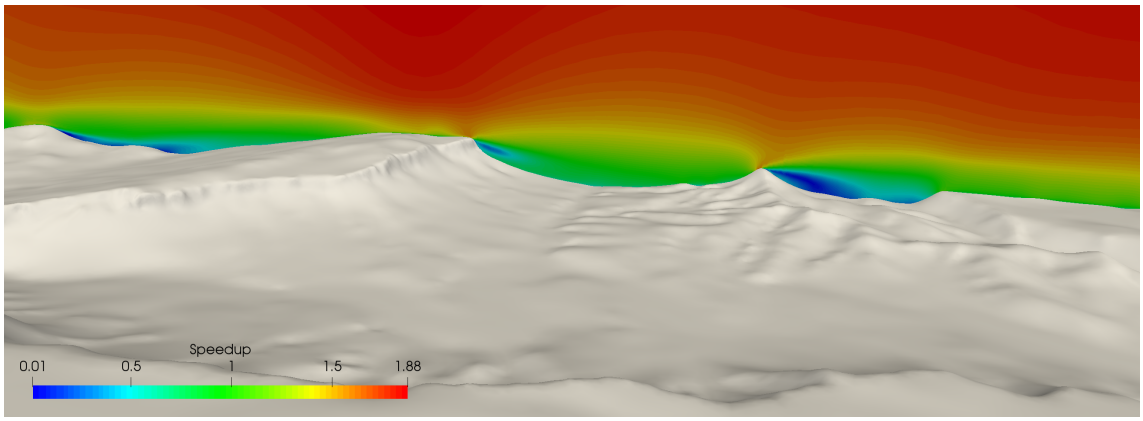


Figure 8: Velocity speedup with respect to a reference point upwind over Badaia topography.

6.2 Badaia topography

In this section, we present the adapted mesh generated on the Badaia topographic scenario, Figure 6(a), located in Spain. The main objective is to illustrate both the adaptive surface technique and the hybrid volume mesher in a real scenario. This topography features a valley surrounded by several plateaus and mountains with orographic steps up to 700 meters. The minimum height of the topography is 388 meters over the sea level, and the maximum 1098 meters. The area of the target meshed domain is of 20x20km^2, and the top ceiling of the mesh is located at 2km over the highest topographic point.

The surface meshing procedure generates a triangle surface mesh with elements featuring edges of at most 75 meters, which are allowed to be refined up to 10 meters to capture the geometry curvature. The generated surface mesh is presented in Figure 6(b). The initial mesh is composed by 41531 nodes and 82379 elements. The mesher performs 4 cycles of the refinement procedure presented in Section 4.1. The final surface mesh is composed by 62469 nodes and 124172 elements. After the adaptive procedure the obtained mesh has a minimum elemental quality of 0.2, which is improved up to 0.24 after performing the optimization procedure presented in Section 4.2.

In Figure 7 two different slices of the volume mesh are presented. The generated volumetric mesh is composed by 2.2M nodes and 5.6M elements, from which 3.7M are prisms and 1.9M are tetrahedra. The mean quality of the generated mesh is 0.84 and the minimum is 0.08. After the mesh optimization, the minimum mesh quality is improved up to 0.11, whereas the mean remains is 0.84. In Figure 8 we illustrate the applicability of the generated meshes for simulation. In particular, we illustrate the wind velocity speedup with respect to a reference point upwind. The wind inflow direction is from left to right of the domain.

Although out of the scope of this paper, in the future we would like to perform an actual study of the benefits of the topography adapted mesh on the solution of the ABL flow. The fact that the mesh can be adapted to the topography may allow us to reduce the degrees of freedom, augmenting the mesh size in large flat regions, and increasing the resolution around the features of the topography that require it.

7. CONCLUDING REMARKS AND FUTURE WORK

In this work, we have presented a new meshing strategy to simulate ABL flows on onshore topographical scenarios. The first contribution is an adaptive procedure to attain the desired mesh size on the topography surface and to capture the curvature of the geometry. Since topographic geometries are discrete (point clouds, contour lines or triangle meshes in contrast to CAD geometries), to compute the required metrics to perform the mesh adaptivity, a new approach to compute local smooth parameterizations of the geometry is presented. In particular, the geometry is locally approximated by a high-order polynomial of the desired degree (by default 3) that allows to compute a local continuous first and second order derivatives. Using those derivatives, we compute the metric of the tangent space and a curvature metric. These metrics allow us to refine the elements of the mesh to attain the desired edge length of the surface mesh and to discretize the curvature of the geometry with the desired accuracy.

The second contribution is the new hybrid volume mesh generation procedure. On the first $10 - 20\%$ of the domain a prismatic mesh is generated to capture the Surface Boundary Layer flow. On the rest of the domain, the flexibility of tetrahedra is exploited to avoid extending the fine mesh on the interest region to the rest of the domain. The result is an hybrid mesh that resolves the Surface Boundary Layer using prisms, and then discretizes the rest of the domain with the desired resolution using tetrahedra. The tetrahedral elements allow imposing a smooth size transition between the finer element size of the prismatic layer and the coarser element size desired in the top planar ceiling.

To illustrate the advantages in terms of the number of generated degrees of freedom, in Section 6.1 a standard semi-structured mesh is compared to the new strategy. In addition, Section 6.2 illustrates the applicability of the mesher to discretize complex topographical scenarios producing valid meshes for simulation. We discretize the Badaia topography with the hybrid approach and we use the generated mesh to perform a RANS simulation using the model presented in [45, 10].

In the near future, we will perform a detailed analysis of the advantages and disadvantages of the use of an hybrid approach to simulate ABL flows against the standard semi-structured approach, both in terms of the simulation accuracy and the number of degrees of freedom of the simulation. In addition, we would like to study the possibility to use the topography adaptation procedure to reduce the degrees of freedom of the mesh, by means of increasing the mesh size in flat areas and only requiring a finer mesh size around the topography features. Regarding the application of the mesher for wind farm assessment, we would also like to explore using the generated hybrid mesh to discretize wind farms modeled with the actuator disc model with a similar approach to the one presented in [9, 16].

ACKNOWLEDGMENTS

This work has been partially funded by the EU H2020 *Energy oriented Center of Excellence (EoCoE) for computer applications*, the *New European Wind Atlas (NEWA)* and the *High Performance Computing for Energy (HPC4E)* projects. We thank Iberdrola Renovables [50] for their collaboration and for providing the topography data.

References

[1] Köpp F., Schwiesow R., Werner C. "Remote measurements of boundary-layer wind profiles using a CW Doppler lidar." *Journal of climate and applied meteorology*, vol. 23, no. 1, 148–154, 1984

[2] Garratt J.R. "The atmospheric boundary layer." *Earth-Science Reviews*, vol. 37, no. 1-2, 89–134, 1994

[3] Blocken B., Stathopoulos T., Carmeliet J. "CFD simulation of the atmospheric boundary layer: wall function problems." *Atmospheric environment*, vol. 41, no. 2, 238–252, 2007

[4] Wizelius T. *Developing wind power projects: theory and practice*. Earthscan, 2007

[5] Stull R.B. *An introduction to boundary layer meteorology*, vol. 13. Springer Science & Business Media, 2012

[6] Sørensen N.N. "HypGrid2D a 2-D mesh generator." Tech. rep., Denmark. Forskningscenter Risoe. Risoe-R; No. 1035(EN), 1998

[7] Stefan I., Mikkelsen R., Sørensen N.N., Henningson D. "Validation of methods using EllipSys3D." Tech. rep., Engelska, 2008

[8] Weller H. "OpenFOAM." https://openfoam.org/, 2018

[9] Gargallo-Peiró A., Avila M., Owen H., Prieto L., Folch A. "Mesh Generation for Atmospheric Boundary Layer Simulation in Wind Farm Design and Management." *Procedia Engineering*, vol. 124, 239–251, 2015

[10] Avila M., Gargallo-Peiró A., Folch A. "A CFD framework for offshore and onshore wind farm simulation." *Journal of Physics: Conference Series*, vol. 854, no. 1, 012002, 2017

[11] B. Bichet T.F., Alexandre P. "Comparison of two CFD tools against measurements on complex terrain." *1st Symposium OpenFOAM dedicated to Wind Energy*, pp. 1–19. 2013

[12] Haltiner G.J., Williams R.T. *Numerical prediction and dynamic meteorology*. Wiley, 1980

[13] Arakawa A., Konor C.S. "Vertical differencing of the primitive equations based on the Charney-Phillips grid in hybrid σ-p vertical coordinates." *Month. Weather Rev.*, vol. 124, no. 3, 511–528, 1996

[14] Michelsen J.A. *Block structured Multigrid solution of 2D and 3D elliptic PDE's*. Department of Fluid Mechanics, Technical University of Denmark, 1994

[15] Marras S. *Variational multiscale stabilization of finite and spectral elements for dry and moist atmospheric problems*. Ph.D. thesis, Universitat Politècnica de Catalunya, 2012

[16] Gargallo-Peiró A., Avila M., Owen H., Prieto-Godino L., Folch A. "Mesh generation, sizing and convergence for onshore and offshore wind farm Atmospheric Boundary Layer flow simulation with actuator discs." *Journal of Computational Physics*, vol. 375, 209–227, 2018

[17] Montero G., Montenegro R., Escobar J. "A 3-D diagnostic model for wind field adjustment." *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 74, 249–261, 1998

[18] Montero G., Rodrıguez E., Montenegro R., Escobar J.M., González-Yuste J.M. "Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model." *Advances in Engineering Software*, vol. 36, no. 1, 3–10, 2005

[19] Escobar J.M., Montero G., Montenegro R., Rodríguez E. "An algebraic method for smoothing surface triangulations on a local parametric space." *Int. J. Numer. Meth. Eng.*, vol. 66, no. 4, 740–760, 2006

[20] Montenegro R., Montero G., Escobar J.M., Rodríguez E., González-Yuste J.M. "Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains." P. Sloot, A. Hoekstra, C. Tan, J. Dongarra, editors, *Computational Science ICCS 2002*, vol. 2329 of *Lecture Notes in Computer Science*, pp. 335–344. Springer Berlin Heidelberg, 2002

[21] Behrens J., Rakowsky N., Hiller W., Handorf D., Läuter M., Päpke J., Dethloff K. "amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation." *Ocean modelling*, vol. 10, no. 1-2, 171–183, 2005

[22] Oliver A., Rodríguez E., Escobar J., Montero G., Hortal M., Calvo J., Cascón J., Montenegro R. "Wind Forecasting Based on the HARMONIE Model and Adaptive Finite Elements." *Pure and Applied Geophysics*, vol. 172, no. 1, 109–120, 2015

[23] Gargallo-Peiró A., Folch A., Roca X. "Representing Urban Geometries for Unstructured Mesh Generation." *Procedia Engineering*, vol. 163, 175–185, 2016. 25th International Meshing Roundtable

[24] Frey P.J. "About Surface Remeshing." *Proc. 9th Int. Meshing Roundtable*, 2000

[25] Frey P., Borouchaki H. "Surface meshing using a geometric error estimate." *International journal for numerical methods in engineering*, vol. 58, no. 2, 227–245, 2003

[26] Frey P., George P. *Mesh Generation: Application to Finite Elements*. Wiley, 2008

[27] Peraire J., Vahdati M., Morgan K., Zienkiewicz O.C. "Adaptive remeshing for compressible flow computations." *J. Comput. Phys.*, vol. 72, no. 2, 449—466, 1987

[28] Frey P.J., Alauzet F. "Anisotropic mesh adaptation for CFD computations." *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, 5068–5082, 2005

[29] Loseille A., Alauzet F. "Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework." *Proc. 18th Int. Meshing Roundtable*, pp. 575–594. Springer, 2009

[30] Loseille A., Alauzet F. "Continuous mesh framework part I: well-posed continuous interpolation error." *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, 38–60, 2011

[31] Loseille A., Alauzet F. "Continuous mesh framework part II: validations and applications." *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, 61–86, 2011

[32] Shewchuk J.R. "Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator." *Applied computational geometry towards geometric engineering*, pp. 203–222. Springer, 1996

[33] Knupp P.M. "Algebraic mesh quality metrics for unstructured initial meshes." *Finite Elem. Anal. Des.*, vol. 39, no. 3, 217–241, 2003

[34] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J. "Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes." *Int. J. Numer. Meth. Eng.*, vol. 103, 342–363, 2015

[35] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J. "Distortion and quality measures for validating and generating high-order tetrahedral meshes." *Eng. Comput.*, vol. 31, 423–437, 2015

[36] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J. "A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization." *Int. J. Numer. Meth. Eng.*, vol. 106, no. 13, 1100–1130, 2015

[37] Gargallo-Peiró A., Roca Navarro X., Peraire J., Sarrate J. "High-order mesh generation on CAD geometries." *Adaptive Modeling and Simulation 2013*, pp. 301–312. 2013

[38] Garanzha V., Kaporin I. "Regularization of the Barrier Variational Method." *Computational mathematics and mathematical physics*, vol. 39, no. 9, 1426–1440, 1999

[39] Escobar J.M., Rodríguez E., Montenegro R., Montero G., González-Yuste J.M. "Simultaneous untangling and smoothing of tetrahedral meshes." *Comput. Meth. Appl. Mech. Eng.*, vol. 192, no. 25, 2775–2787, 2003

[40] Gargallo-Peiró A., Roca X., Sarrate J. "A surface mesh smoothing and untangling method independent of the CAD parameterization." *Comput. Mech.*, vol. 53, no. 4, 587–609, 2014

[41] Si H. "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator." *ACM Trans. Math. Softw.*, vol. 41, no. 2, 11:1–11:36, 2015

[42] Roca X. *Paving the path towards automatic hexahedral mesh generation*. Ph.D. thesis, Universitat Politècnica de Catalunya, 2009

[43] Roca X., Sarrate J. "An automatic and general least-squares projection procedure for sweep meshing." *Eng. Comput.*, vol. 26, no. 4, 391–406, 2010

[44] Apsley D., Casrto I. "A limited-length-scale k-ε model for the neutral and stably-stratified Atmospheric Boundary Layer." *Bound.-Lay. Meteorol.*, vol. 83, no. 1, 75–98, 1997

[45] Avila M., Folch A., Houzeaux G., Eguzkitza B., Prieto L., Cabezón D. "A Parallel CFD Model for Wind Farms." *Procedia Computer Science*, vol. 18, no. 0, 2157 – 2166, 2013. 2013 International Conference on Computational Science

[46] Houzeaux G., Vázquez M., Aubry R., Cela J. "A massively parallel fractional step solver for incompressible flows." *J. Comput. Phys.*, vol. 228, no. 17, 6316 – 6332, 2009

[47] Vázquez M., Houzeaux G., Koric S., Artigues A., Aguado-Sierra J., Arís R., Mira D., Calmet H., Cucchietti F., Owen H., Taha A., Burness E., Cela J., Valero M. "Alya: Multiphysics engineering simulation toward exascale." *Journal of Computational Science*, vol. 14, 15–27, 2016

[48] Gargallo-Peiró A., Houzeaux G., Roca X. "Subdividing triangular and quadrilateral meshes in parallel to approximate curved geometries." *Procedia Engineering*, vol. 203, 310–322, 2017

[49] Barcelona Supercomputing Center. "MareNostrum 3." `https://www.bsc.es/marenostrum`, 2018

[50] Dpt. E.R. "Iberdrola Renovables." `http://www.iberdrolarenovablesenergia.com`, 2001