ONSHORE WIND FARM MODELLING

M. Avila[a]**, A. Folch**[a]**, G. Houzeaux**[a]**, B. Eguzkitza**[a]**, L. Prieto**[b] **and D. Cabezón**[c]

[a]*CASE-Environmental Simulations, Barcelona Supercomputing Center, Nexus II Building, Jordi Girona 29, 08034-Barcelona, Spain http://www.bsc.es*

[b]*Iberdrola Renovables, Spain, http://http://www.iberdrola.es*

[c]*Centro Nacional de Energías Renovables (CENER), Spain, http://www.cener.com*

Abstract. We present a Computational Fluid Dynamics (CFD) modeling strategy for onshore wind farms aimed at predicting and optimizing the production of farms using a CFD model that includes meteorological data assimilation, complex terrain and wind turbine effects. The model involves the solution of the Reynolds-Averaged Navier-Stokes (RANS) equations together with a κ-ε turbulence model specially designed for the Atmospheric Boundary Layer (ABL). The model involves automatic meshing and generation of boundary conditions with atmospheric boundary layer shape for the entering wind flow. As the integration of the model up to the ground surface is still not viable for complex terrains, a specific law of the wall including roughness effects is implemented. The wake effects and the aerodynamic behavior of the wind turbines are described using the actuator disk model, upon which a volumetric force is included in the momentum equations. The placement of the wind turbines and a mesh refinement for the near wakes is done by means of a Chimera method. The model is implemented in Alya, a High Performance Computing (HPC) multi physics parallel solver based on finite elements and developed at Barcelona Supercomputing Center.

1 INTRODUCTION

Numerical modeling of wind farms is a crucial aspect in terms of both wind farm design and management. Modeling of wind farms must consider all aspects affecting surface layer atmospheric flow such as topographic variations, heterogeneities in the roughness of the terrain, or the downwind wake effects of wind turbines. During the last years, the growth in computational capacity has allowed to implement modeling strategies based on Computational Fluid Dynamics (CFD) involving the full resolution of the turbulent flow equations within the Atmospheric Boundary Layer (ABL) using both Reynolds-averaged Navier-Stokes (RANS) and Large Eddy Simulations (LES) approaches (for a recent review see e.g. Sanderse et al., 2011).

Here we present the modeling strategy under development within the research Spanish national project "Modelo de viento basado en código abierto de CFD OpenFOAM y técnicas de supercomputación aplicadas" (Proyecto INNPACTO IPT-2011-1693-920000, 2011-2014) involving three partners: Iberdrola Renovables S.A., the National Center for Renewable Energies (CENER) and the Barcelona Supercomputing Center-Centro Nacional de Supercomputación (BSC-CNS). The main goal of the project is to port the state-of-the-art of onshore wind farm modeling to High Performance Computing (HPC) applied to industry.

The project working plan is divided in two stages. During a first phase, already concluded, we have constrained to high resolution wind modeling in complex terrain by solving the RANS equations coupled with a κ-ε length-limited turbulence model. During the second stage the effect of wind turbines is added making use of different disk actuator models. The proposed modeling strategy, yet under development, involves: i) an automatic pre-process with mesh generation and terrain data assimilation, ii) the numerical solution of the governing equations in the context of HPC using Alya, a BSC-CNS home-made code based on Finite Elements and, iii) the post-process of results using Google Earth overlays and an automatic evaluation of speed-ups of the wind turbines. This papers describes the methodology, modeling strategy and preliminary results of this still on-going project.

2 MODELING STRATEGY

2.1 Pre-process

Mesh generation and terrain data assimilation is a necessary pre-process step in classical numerical simulation algorithm. First of all we have written a structured mesh generator for hexahedral elements in complex terrain that handles most industry standard formats for terrain elevation and surface roughness and exports the resulting mesh to several standard formats. The code (WindMesh) is used to generate a "background mesh" containing the assimilated topography at high resolution (typical grid sizes range from few to tens of meters in the horizontal) and has the following characteristics:

- At ground level there exist three differentiated zones named FARM, TRANSITION, and BUFFER (Fig. 1). The external BUFFER zone is flat and made up with regular elements, the TRANSITION zone has elements of variable size and the innermost FARM zone contains finer regular elements. The external BUFFER zone can be rectangular or circular and is designed to accommodate the inflow conditions. The TRANSITION and FARM zones contain topography and roughness surfaces.

- The vertical distribution of elements can be linear or follow a geometric progression in order to increase resolution near the ground surface (Fig. 1f). In turn, in cases with high

topographic gradients, the vertical distribution of nodes can be modified using an elliptic smoothing in order to have ortogonality and prevent elemental distortion.

- Terrain elevation and roughness contours can be read from MAP or STL format files, *i.e.* can be imported from industry standard commercial codes like WAsP (Wind Atlas Analysis and Application Program). Both types of contours are interpolated using a Delauney triangulation and the resulting topography and roughness surfaces can optionally be smoothed and rotated to align with the wind attack angle.

- WindMesh outputs results in several standard formats, including a kmz file to visualize the emplacement and mesh properties using GoogleEarth (Fig. 1a).

- Preprocessing of tracking points representing the positions of the wind turbines. This facilitates the subsequent post-process of wind results at the turbine locations.

WindMesh generates a background structured mesh. Wind turbines and a downstream higher-resolution zone (to capture wake effects) are overimposed to the background mesh using a chimera method as explained in section 2.3.

2.2 ABL wind field modeling

2.2.1 Governing equations

In this section, we briefly discuss the governing equations for turbulent flow in the Atmospheric Boundary Layer (ABL). The equations of motion that need to be solved are the Averaged Reynolds Navier-Stokes coupled with the $k - \varepsilon$ turbulent model equations in a domain $\Omega \subset \Re^3$:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot ((\nu + \nu_t) \nabla^s \boldsymbol{u}) + \nabla p + 2\boldsymbol{\omega} \times \boldsymbol{u} = \boldsymbol{f} \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2}$$

$$\frac{\partial k}{\partial t} + \boldsymbol{u} \cdot \nabla k - \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \cdot \nabla k \right) + \varepsilon - P_k = 0 \tag{3}$$

$$\frac{\partial \varepsilon}{\partial t} + \boldsymbol{u} \cdot \nabla \varepsilon - \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \cdot \nabla \varepsilon \right) + C_{\varepsilon 2} \frac{\varepsilon^2}{k} - C_{\varepsilon 1} \frac{\varepsilon}{k} P_k = 0 \tag{4}$$

$$\nu_t = C_\mu k^2 / \varepsilon \tag{5}$$

$$P_k = 2\nu_t \nabla^s \boldsymbol{u} : \nabla^s \boldsymbol{u} \tag{6}$$

where equations (1) and (2) are the momentum and mass conservation equations. Here $\boldsymbol{u}$ is the mean velocity field, p is the mean pressure, $\boldsymbol{f}$ is the vector of body forces, ν is the kinematic viscosity of the fluid, and ν_t is the eddy viscosity that needs to be modelled. The vector $\boldsymbol{\omega}$ is the velocity of rotation of the frame of reference ($|\boldsymbol{\omega}| = 7.292 \times 10^{-5}$ Rad/s in our particular case) and $2\boldsymbol{\omega} \times \boldsymbol{u}$ is the Coriolis force. The k-ε model is given by equations (3) and (4) where the two turbulent quantities are the turbulent kinetic energy k and dissipation rate ε. Here P_k is the production term of kinetic energy, given by expression (6), and the eddy viscosity is modeled in terms of k and ε by expression (5). The k-ε model constants C_μ, $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k and σ_ε are set to its standard values (Launder and Sharma, 1974).
Detering and Etling (1985) concluded that the standard k-ε model when applied to the atmospheric boundary layer, yields a very deep boundary layer, large mixing length (and hence eddy-viscosity) in the upper boundary layer, and large friction velocity when compared with

observations. These deficiencies can all be traced to inability of the $k - \varepsilon$ model in its standard form to recognize some finite upper limit to the mixing length. There exist different mixing length limitations that can be imposed to the standard $k - \varepsilon$ model consisting on increasing the production term in the ε equation (4). In this work this is done by replacing coefficient $C_{\varepsilon 1}$ by $C'_{\varepsilon 1}$ as proposed by Apsley and Castro (1997):

$$C'_{\varepsilon 1} = C_{\varepsilon 1} + (C_{\varepsilon 2} - C_{\varepsilon 1})\frac{l_m}{l_{max}} \tag{7}$$

where l_m is the mixing length, given by $l_m = \frac{C_\mu^{3/4} k^{3/2}}{\varepsilon}$, and $l_{\max}$ is the maximum limited mixing length, which is a model input to be given.

The boundary conditions for the Navier-Stokes problem (1) - (2) are:

$$\boldsymbol{u} = \boldsymbol{u_{in}} \quad \text{on} \quad \Gamma_{in} \tag{8}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{0} \quad \text{on} \quad \Gamma_{out} \tag{9}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0, \quad \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{g_1} = \boldsymbol{t_1}, \quad \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{g_2} = \boldsymbol{t_2} \quad \text{on} \quad \Gamma_w \tag{10}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{n}$ is the unit exterior normal to $\partial\Omega$. The boundary $\partial\Omega$ is split into three sets of disjoint components Γ_{in}, Γ_{out} and Γ_w, being respectively the inflow part of the boundary (where $\boldsymbol{u} \cdot \boldsymbol{n} \leq 0$), the outflow part of the boundary (where $\boldsymbol{u} \cdot \boldsymbol{n} > 0$), and the surface terrain boundary. In this latter part we prescribe mixed conditions: normal zero velocity and the tangent stresses, where the vectors $\boldsymbol{g_1}$ and $\boldsymbol{g_1}$ span the space tangent to Γ_w. In the ABL a specific law of the wall including roughness effects needs to be prescribed over Γ_w, where the components of the shear stress vectors $\boldsymbol{t_1}$, $\boldsymbol{t_2}$ are expressed in terms of the velocity. The shear stress on the boundary is given by:

$$\boldsymbol{t} = -\frac{\rho u_*^2}{|\boldsymbol{u}|}\boldsymbol{u} \tag{11}$$

where ρ is the fluid (air) density and u_* is the friction velocity, determined by the following wall law for atmospheric boundary layers:

$$|\boldsymbol{u}(z = \delta_w)| = \frac{u_*}{\kappa} log\left(\frac{\delta_w + z_0}{z_0}\right) \tag{12}$$

where z_0 is the terrain roughness and δ_w is the distance from the wall at which the velocity is evaluated. The constant κ is the Von Karman constant, taken as $\kappa = 0.41$.

Proper boundary conditions have to be added also to the $k - \varepsilon$ system of equations (3) - (4). For the case in which the velocity is prescribed (*i.e.* on Γ_{in}), k and ε are also prescribed as the solution of an idealized horizontally-homogeneous one-dimensional problem over a flat plane, wherein the mean velocity $\boldsymbol{u_{in}} = (u, v, 0)$ is function of the vertical coordinate z alone. In this one-dimensional problem the same Coriolis force, fluid properties, and maximum mixing length of the 3D problem are considered. On the outflow boundary (*i.e.* on Γ_{out}), where zero traction is imposed, the boundary conditions for $k - \varepsilon$ are:

$$\frac{\partial k}{\partial n} = 0, \qquad \frac{\partial \varepsilon}{\partial n} = 0, \tag{13}$$

where $\partial/\partial n$ is the normal derivative on the boundary of Ω. Finally, on the terrain boundary (*i.e.* on Γ_w) boundary conditions for k and ε including roughness effects are imposed at a distance

δ_w from the wall as:

$$k(z = \delta_w) \;=\; \frac{u_*}{\sqrt{C_\mu}} \tag{14}$$

$$\varepsilon(z = \delta_w) \;=\; \frac{u_*^3}{\kappa(z_0 + \delta_w)} \tag{15}$$

When a wall law is prescribed for the velocity, u_* is determined from (12). However, when the non slip condition $\boldsymbol{u} = \boldsymbol{0}$ is imposed on Γ_w the wall distance is taken as $\delta_w = 0$ in Eq. (15) and u_* is determined from

$$u_* = \sqrt{\mid \nu \frac{\partial \boldsymbol{u}}{\partial n} \mid} \tag{16}$$

2.2.2 Numerical algorithm

The equations described previously are discretized in space using the finite element method. It is well-known that Galerkin formulations can lack stability for the Navier-Stokes equations for three main reasons. The first reason is related to the compatibility of the finite element spaces for velocity and pressure, which have to satisfy the so-called LBB condition. The second reason is attributed to the relative importance of the viscous and convective effects. Finally, the third one appears when the Coriolis force becomes important with respect to viscous effects. We circumvent these instabilities using a stabilized finite element formulation based on the variational multiscale method (VMM) (Hughes, 1995), using the same interpolation for pressure and velocity spaces. The Navier Stokes equations are discretized using the algebraical subgrid scale model (ASGS), as described in Codina (2001).

Though mathematical results exist ensuring the well-posedness of the k-ε equations, the strong nonlinearities may interact with discretization errors in such a way as to instabilize computations. A typical behavior of unstable computations involves the loss of positivity of k or ε. This changes the sign of several terms in the equations, with disastrous effects. To avoid such loss of positivity we use a linearization scheme that preserves positivity in the numerical schemme, described in Codina and Soto (1999). The k-ε equations are discretized using finite element methods, with stabilization based on the VMM, and adding shock capturing terms for crosswind dissipation.

2.3 The effects of wind turbines

2.3.1 The actuator disk concept

The effects of the rotor are modeled using the actuator disk concept (Sanderse et al., 2011). In this approximation, the momentum needed to put in motion the wind turbine is extracted from the Navier-Stokes equations. The real rotor is approximated by a permeable thin cylinder of equivalent area A inside which the total linear momentum sink is distributed uniformly within a volume V. The rotor model used in this work is based on the 1-dimensional axial momentum theory for a uniformly loaded rotor and non-rotating flow in which the change of momentum is only due to pressure differences across the actuator disc. The expression for the module of the force sink is:

$$F = \frac{1}{2}\rho A C_t U_\infty^2 \quad [N]. \tag{17}$$

where the force F is expressed in terms of the upstream velocity module $U_\infty [m/s]$ and the rotor thrust coefficient C_t. This coefficient is rotor dependent and determined experimentally. In

order to take into account the actuator disc, the momentum equation (1) is modified as:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \left(\left(\nu + \nu_t \right) \nabla^s \boldsymbol{u} \right) + \nabla p + 2\boldsymbol{\omega} \times \boldsymbol{u} = \boldsymbol{f} + \delta_V F / (\rho V) \boldsymbol{n}, \qquad (18)$$

where δ_V is the delta Dirac function with support in the cylinder V and $\boldsymbol{n}$ is the cylinder unit normal pointing towards the incoming flow.

When implementing the actuator disk in a finite element code one faces two difficulties. The first one is the geometrical representation of the rotor inside the mesh. The second one concerns the accuracy of the solution in the wakes of the rotor. Regarding this last point, accurate calculations in the neighboring of the rotor are essential for a proper estimation of the wind farm power. This requires mesh refinement in the wake, which is computationally tricky when treating with structured or Cartesian meshes. As far as the geometrical representation of the cylinder is concerned, we have several options. On the one hand, the mesh can be locally adapted to represent the disk in which the momentum is extracted. This technique requires to adapt the mesh for each configuration and is quite difficult to implement for structured and Cartesian meshes. In fact, one direction should necessarily be aligned with the cylinder. On the other hand, Chimera or overset techniques offer a nice alternative to face the two aforementioned difficulties.

2.3.2 Chimera method

The Chimera Method was originally developed in Steger et al. (1983); Benek et al. (1985); Steger and Benek (1987) to simplify the construction of computational meshes about complex geometries. This is achieved by breaking the geometries into components and generating independent meshes for each subdomain, which gives a great flexibility on the choice of the type of element, their orientation and local mesh refinement. The components are further coupled by imposing transmission conditions (typically done using domain decomposition methods, DDM) from one mesh to the other to obtain a global solution. The idea of the Chimera method in the present context is to create a structured background mesh of the order of kilometers including the terrain, and place inside it mesh patches containing the rotors as illustrated in Figure 2. The Chimera Method can be seen as a preprocess technique, named hole cutting, plus a domain decomposition method to couple overlapping and non-conforming meshes. The hole cutting consists in removing the elements of the background mesh located inside the patch mesh. It is essentially a pre-process step. Once the hole is created, we are left with two unconnected slightly overlapping meshes, or subdomains, with apparent interfaces. The coupling between the subdomains is usually achieved via transmission conditions to impose both the continuities of the unknown and it flux across the subdomain interfaces. There are many possibilities to achieve these continuities using DDM. An extensive bibliography of DDM can be found in Quarteroni and Valli (1999). Traditionally, the coupling has been mainly achieved in an iterative way, leading to a family of DDM referred to as *iteration-by-subdomain* methods. To circumvent the inherent non-linearity of *iteration-by-subdomain* methods, we have developed an alternative coupling strategy. It consists in connecting the two independent meshes by creating some new elements, called *extension elements*. These extension elements connect the nodes of one subdomain with the nodes of the adjacent subdomain. In Figure 3, we illustrate the process of constructing the extension elements from the patch to the background. Extension elements are also necessary in the other way, that is to connect the background interface to the patch. The creation of extension elements consists in the following process:

- Identify the interface nodes.

- For each interface node, identify the surrounding nodes in the adjacent subdomain.

- From the interface node and using the surrounding nodes, create the best elements (using some quality criterion).

The extension elements are created to connect the interface nodes to the nodes of the adjacent subdomain by forming a global shape function with compact support for the interface nodes. In the element loop, only the equation for the corresponding interface nodes should be assembled when dealing with the extension elements of these nodes. In 3-dimensional cases, the process is not straighforward mainly due to the restriction for closing the extension. Figure 3 shows an example of extension elements where two subdomains are coupled: the patch mesh containing the rotor, composed of non-strucutured tetrahedra; the background mesh containing the terrain. In this case, the extension elements for the patch interface are tetrahedra and for the background interface they are pyramids.

2.4 Alya parallel solver

The equations have been implemented in the Alya parallel solver, a HPC code developed at Barcelona Supercomputing Center that is able to run with thousands of processors with an optimal scalability. The parallelization in Alya is based on a sub-structuring technique, using a Master-Slave strategy. A first step is carried out to partition the original mesh and distribute the corresponding geometrical and simulation data to the slaves. From the Master's and Slaves' point of view, it consists of:

- Master: reads the mesh, creates the mesh partition, sends each sub-mesh and simulation data to the corresponding slaves.

- Slaves: receive their sub-mesh. The partition has been carried out element-wise so the slaves only share interface nodes.

One important point of these preprocess steps is the scheduling of the communication strategy. All along the simulation, neighboring slaves will have to exchange data on their interfaces. The order of the communications is of real significance to obtain a good scalability. At this point, the Master does no longer have any mesh dependent variable, as they are distributed among the slaves. As a second step, the simulation can continue as follows:

- Master: is in charge only of output tasks like convergence, time steps, etc.

- Slaves:

    - Assembly: they assemble their local matrices via an element loop.
    - Algebraic solvers: the slaves perform their local matrix-vector products and dot products (basic operations of iterative solvers). The matrix-vector product is assembled on the interfaces between subdomains using the MPI function *MPI_SendRecv*. The dot products are assembled using the MPI function *MPI_AllReduce*.

Figure 4 illustrates in a simplified way the flowchart of the parallel execution of Alya. The communications are drawn with dotted lines to outline the two main kinds of communication present in the solver. For further information on the parallelization of Alya, see Houzeaux et al. (2009).

3 PRELIMINARY RESULTS

Here we show some preliminary results of the project (first year), in particular of wind modeling in complex terrain. As explained, the effect of wind turbines is still under implementation in the Alya code.

3.1 Leipzig wind profile

The implementation of the boundary conditions for wind model for the ABL has been tested using the problem considered in Apsley and Castro (1997). In this problem an idealized horizontally-homogeneous, atmospheric boundary layer over a flat plane is considered, wherein the mean velocity $\boldsymbol{U} = (u, v, 0)$ is function of the vertical coordinate z alone. The aim of the present example is to show the difference of the obtained results when using the standard k-ε model and when adding the limited mixing length correction (7). Comparisons between the obtained results and experimental values is done.
The problem is defined by a computational height of $z \in [0, 3000]$ (in m), a wind velocity at the top prescribed to $u = 17.5$ m/s and aligned with the geostrophic wind ($\alpha = 0^o$), and a constant roughness length of $z_0 = 0.3$ m. The Coriolis force corresponds to that of a latitude 45 degrees N, and the maximum mixing length in the correction model is taken as $l_{max} = 36$ m. Figures (5) and (6) compare the obtained velocities, eddy viscosity, angle of turning of the wind and mixing length profiles, using the standard $k - \varepsilon$ model and the modifications introduced by Apsley and Castro (1997). The ABL model that better adjusts to experimental measurements is the limited length scale model. The standard $k - \varepsilon$ model predicts too large mixing length and therefore kinematic viscosity. The results obtained using the mixing length correction are clearly observed to be much more accurate than if no using any correction.

3.2 Wind in complex terrain

We have modeled wind field in several wind farms considering 16 characteristic inflow wind directions. Figure 7 shows results for a domain of 20×20km in the horizontal and 8km in the vertical meshed with 3M hexahedra. Element resolutions range from 20m in the inner farm zone to 300m in the buffer zone. In the vertical start from 0.5m in the first layer and follow a geometrical distribution.

REFERENCES

Apsley D. and Castro I. A limited-length-scale k-ϵ model for the neutral and stably-stratified atmospheric boundary layer stationary incompressible flows. *Boundary-Layer Meteorology*, 83:75–98, 1997.

Benek J., Buning P., and Steger J. A 3-d chimera grid embedding technique. 1985. *AIAA Paper*, pages 85–1523, 1985.

Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190(20–21):2681–706, 2001.

Codina R. and Soto O. Finite element implementation of two-equation and algebraic stress turbulence models for steady incompressible flows. *IJNMF*, 30:309–334, 1999.

Detering H. and Etling D. Application of the $e-\epsilon$ turbulence model to the atmospheric boundary layer. *boundary-Layer Meteorology.*, 33:113–133, 1985.

Houzeaux G., Vázquez M., Aubry R., and Cela J. A massively parallel fractional step solver for incompressible flows. *J. Comput. Phys.*, 228(17):6316–6332, 2009.

Hughes T. Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering*, 127:387–401, 1995.

Launder B. and Sharma B. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Lett. Heat Mass Transf.*, 1:131–138, 1974.

Quarteroni A. and Valli A. *Domain decomposition methods for partial differential equations.* Oxford University Press, USA, 1999.

Sanderse B., van der Pijl S., and Koren B. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind Energ.*, 14:799–819, 2011.

Steger J. and Benek J. On the use of composite grid schemes in computational aerodynamics. *Comp. Meth. Appl. Mech. Eng.*, 64:301–320, 1987.

Steger J., Dougherty F., and Benek J. A chimera grid scheme. *Advances in Grid GEneration*, 5:59–69, 1983.
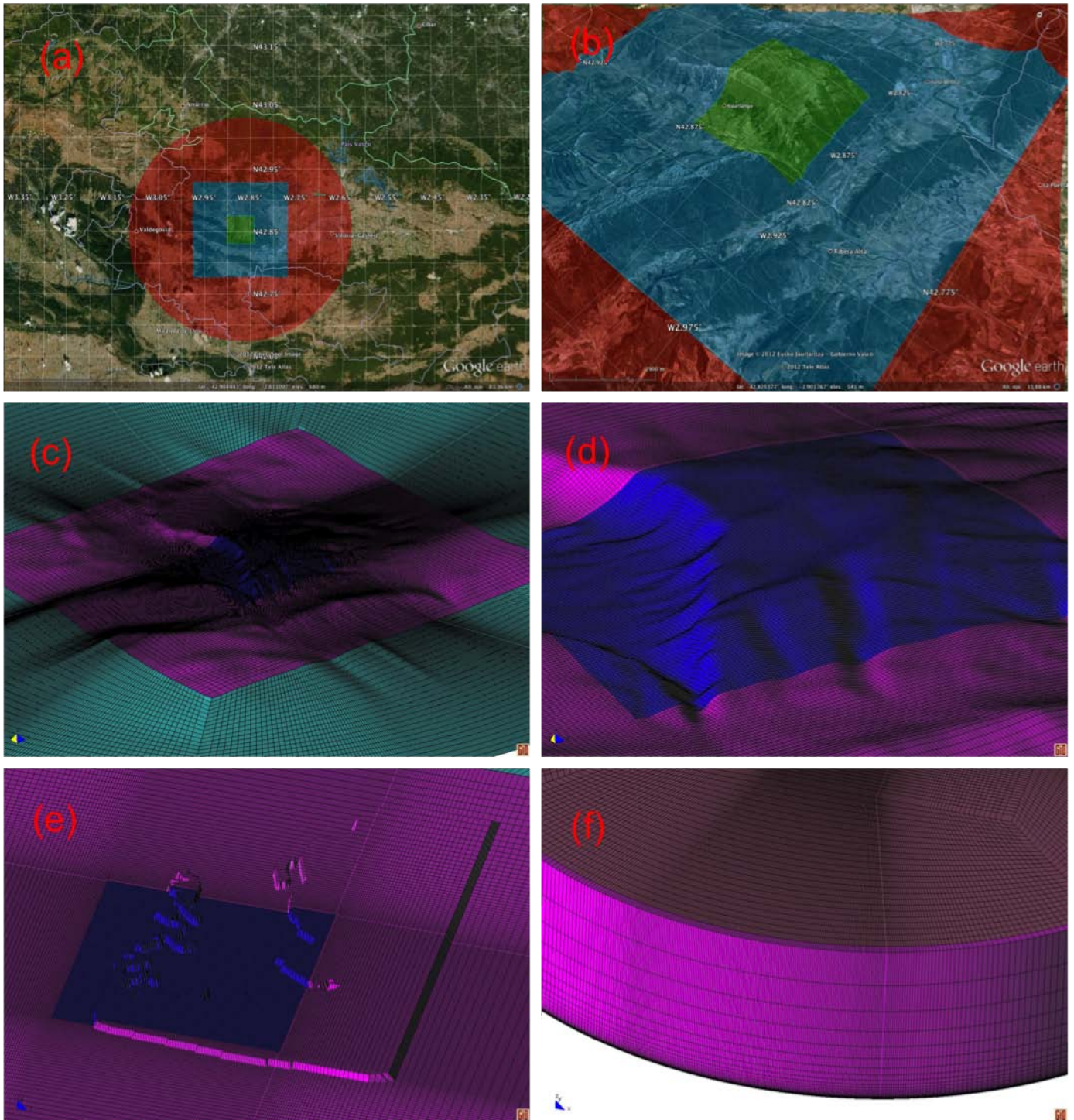
Figure 1: Example of WindMesh output. (a) and (b) show the buffer (red), transition (blue), and farm (green) zones as seen in GoogleEarth. (c) y (d) surface mesh at the different zones. Topography has been smoothed. (e) roughness contours interpolated from a map file. (f) detail of the 3D mesh.
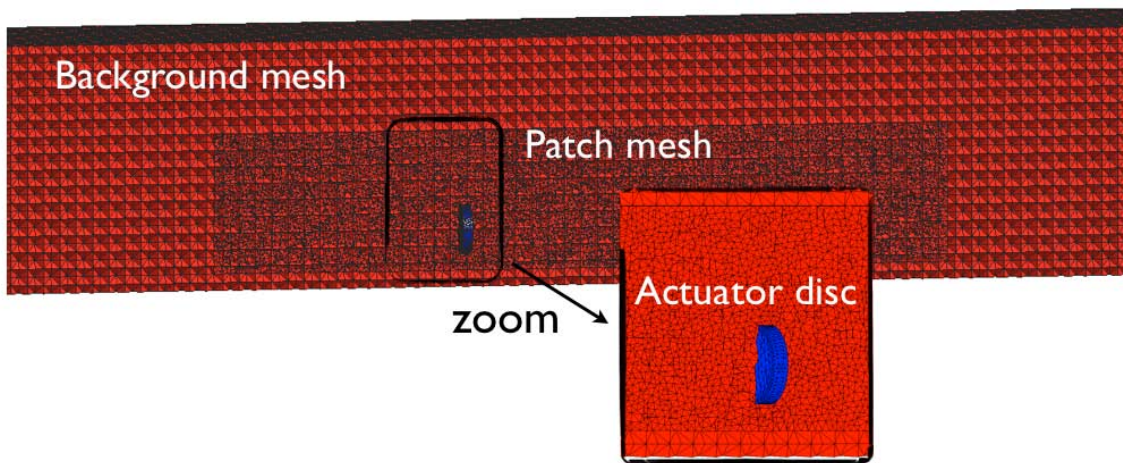
Figure 2: Example to illustrate the Chimera concept in wind farm modeling. A patch mesh contains the actuator disc, represented by a body-fitted mesh, and eventually local refinement dowstream. Note that the interior of the disk is also solved so that the mesh covers both the exterior and interior parts of the rotor.

Figure 3: Example of the extension elements used to connect patch and background meshes. A brick patch meshed with tetrahedral (pyramids) elements contains the actuator disk (shown in blue) and interfaces with a background structured mesh of hexahedral elements. For clarity, the inset shows a simpler 2D example of extension elements.
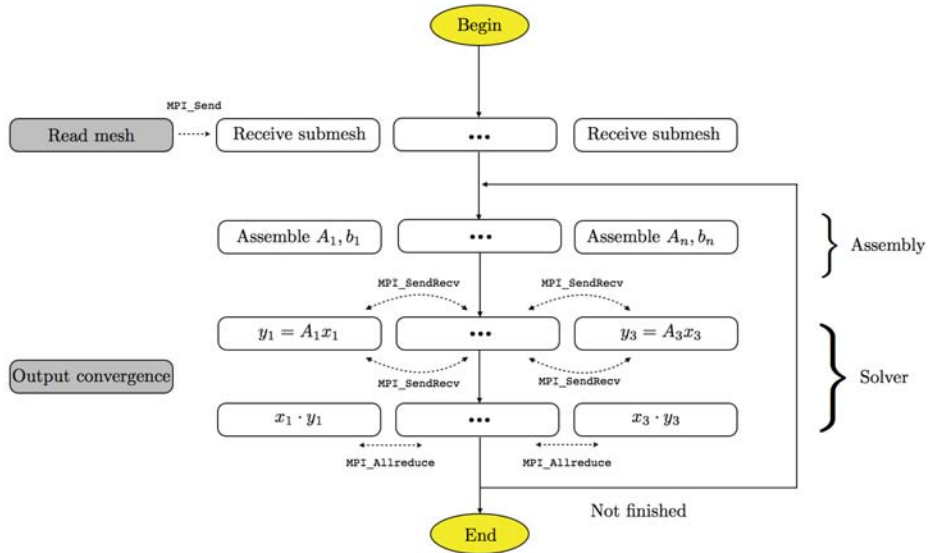
Figure 4: Flowchart of Alya parallel execution using $n + 1$ processes (n subdomains).
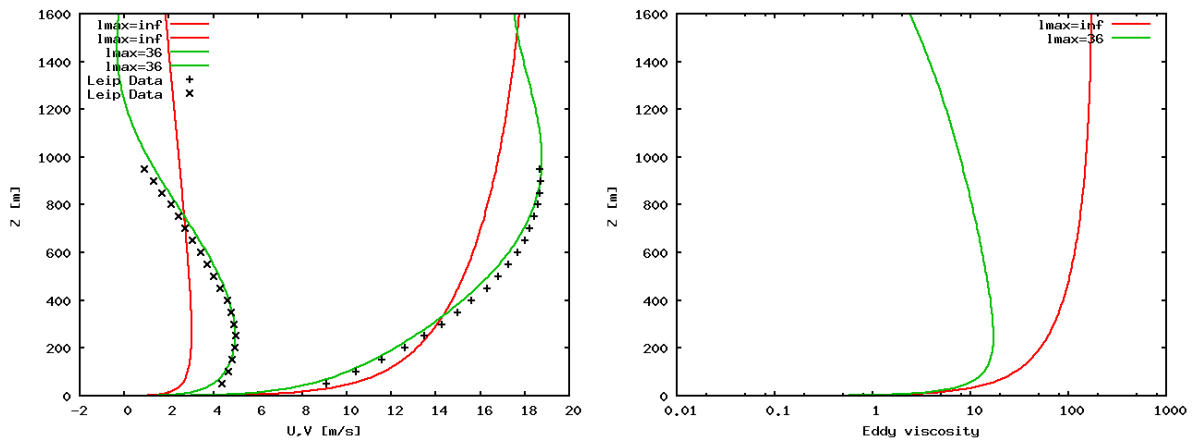

Figure 5: Vertical profiles of wind velocity components (left) and eddy viscosity (right) using the standard k-ε model (i.e. $l_{max} \to \infty$) and the ABL k-ε model modified by limiting the mixing length as explained in section 2.2.1 ($l_{max} = 36$m in this case). Note how the second option improves substantially the fits to the experimental Leipzig profile data, indicated by crosses.
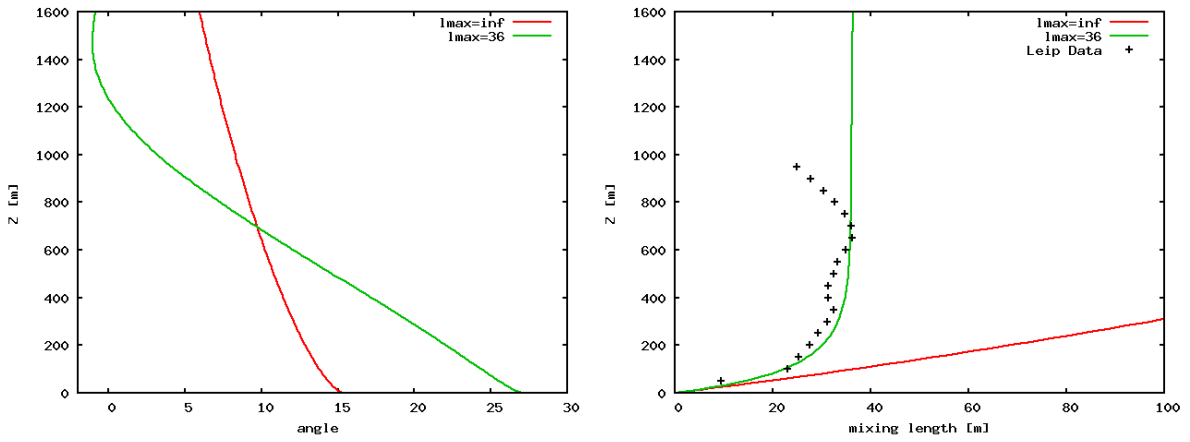
Figure 6: Vertical profiles of wind turning angle ($\alpha = 0^o$ at $z = 3000$m) and mixing length profiles using the standard k-ε model (ι.e. $l_{max} \to \infty$) and the ABL k-ε model modified by limiting the mixing length as explained in section 2.2.1 ($l_{max} = 36$m in this case). Experimental measurements showed a surface wind turned by 26.1^o from the geostrophic and a friction velocity of $u_* = 0.65$ m/s. The standard $k-\varepsilon$ predicts too large friction velocity $u_* = 0.81$ m/s whereas the limited length model predicts $u_* = 0.676$ m/s. The surface wind is turned only 15° when using the standard k-ε model and 27° when using the mixing length correction, in excellent agreement with measurements.
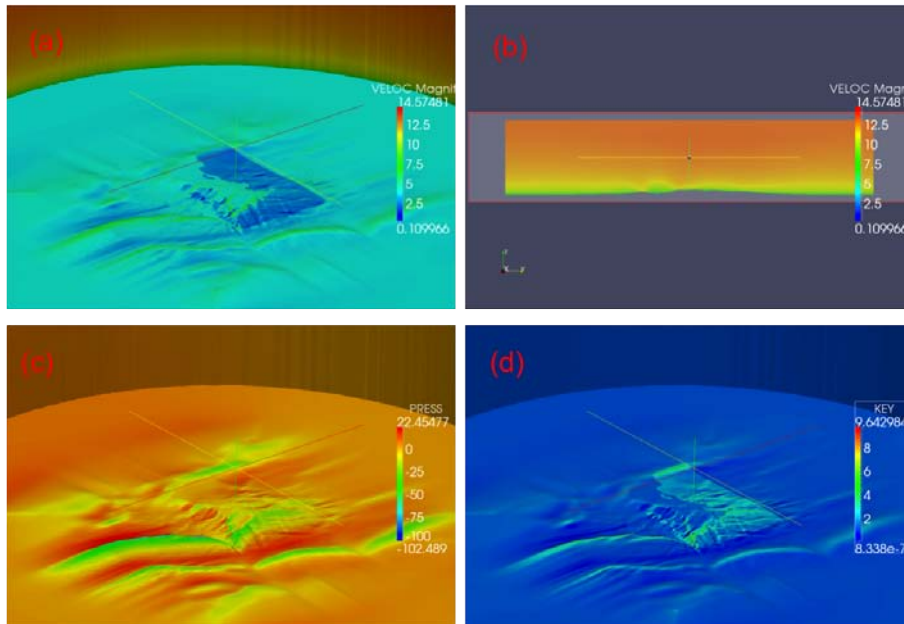

Figure 7: Results for ABL simulations of a wind farm in Spain. (a) wind speed at surface. (b) Wind velocity in a vertical cut along the inflow wind direction. (c) pressure at surface. (d) turbulent kinetic energy at surface.