Use of OpenFOAM for Wind Resource and Site Assessment Applications in Wind Energy

João Azevedo
Carlos Silva Santos

OpenFOAM 3rd Iberian Meeting

Porto, Portugal, June 12th, 2019

Contents

- PART I - THE NEED FOR CFD IN WIND ENERGY STUDIES
  1. Steps in the design of a wind farm
  2. Preliminary site identification
  3. Wind measurements
  4. Wind modelling
  5. Resource assessment
  6. Site assessment
  7. Difficulties

PART I - The Need for CFD in Wind Energy Studies (2/ 19)

MEGAJOULE
Rethinking Energy Worldwide

- industry experience
- knowledge of clients needs
- identify trends and R&D needs
- drives DEVELOPMENT

+

isep Instituto Superior de Engenharia do Porto

- CFD developers for atmospheric flows with over 15 years experience
- scientific approach
- highly qualified support
- access to other research groups & resources
- drives RESEARCH

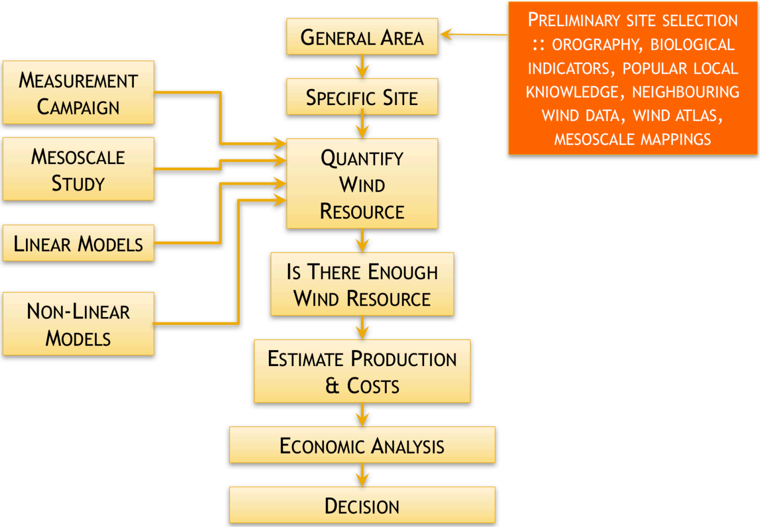
MEGAJOULE INOVAÇÃO

- Founded in 2009
- R&D branch of the Megajoule group
- Responsible for high-end services:
  - windie CFD wind modelling
  - windsight mesoscale services
  - advanced solar energy solutions

universidade de aveiro

- collaboration on meteorological services
- shared PhD supervision
- internships and MSc theses

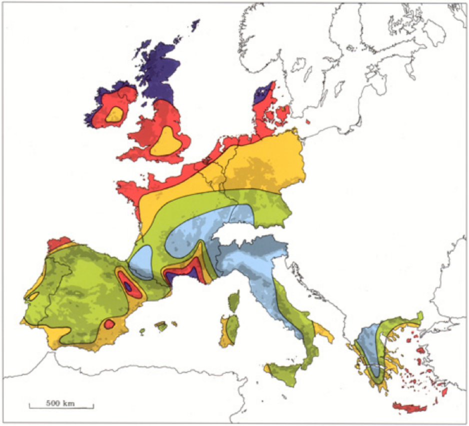
Steps in the installation of a wind farm



Analysis of topography

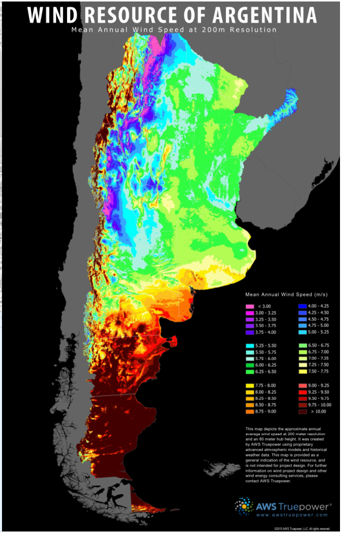
Knowledge of local population

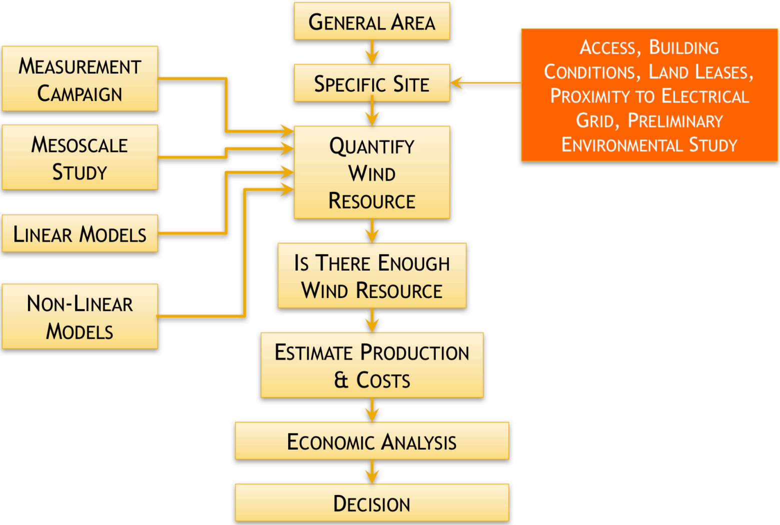
Biological indicators :: vegetation, dunes, etc.

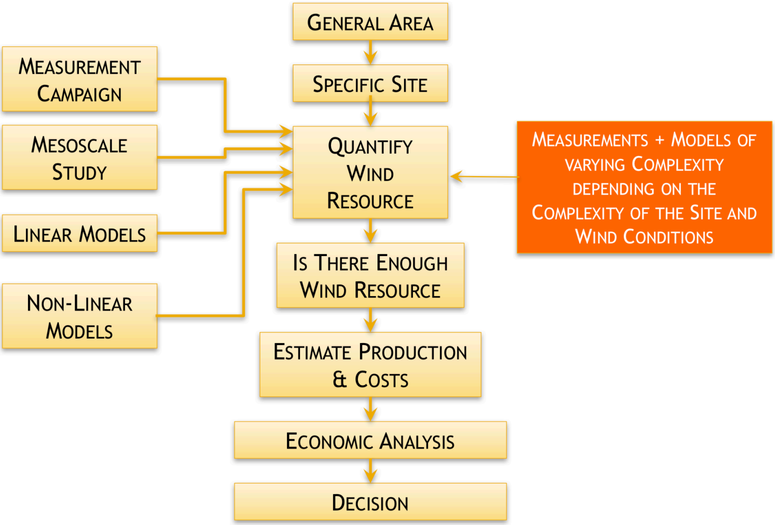


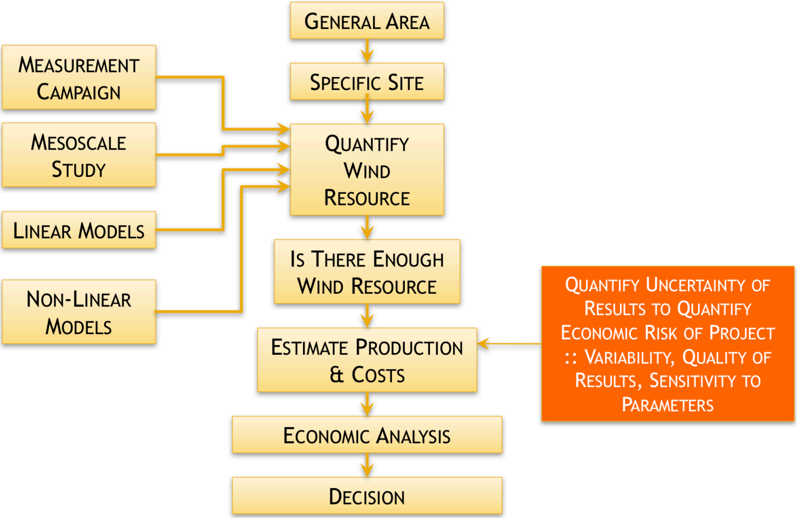
Wind resources[1] at 50 metres above ground level for five different topographic conditions										
	Sheltered terrain[2]		Open plain[3]		At a sea coast[4]		Open sea[5]		Hills and ridges[6]	
	$m\,s^{-1}$	Wm^{-2}	$m\,s^{-1}$	Wm^{-2}	$m\,s^{-1}$	Wm^{-2}	$m\,s^{-1}$	Wm^{-2}	$m\,s^{-1}$	Wm^{-2}
	> 6.0	> 250	> 7.5	> 500	> 8.5	> 700	> 9.0	> 800	> 11.5	> 1800
	5.0-6.0	150-250	6.5-7.5	300-500	7.0-8.5	400-700	8.0-9.0	600-800	10.0-11.5	1200-1800
	4.5-5.0	100-150	5.5-6.5	200-300	6.0-7.0	250-400	7.0-8.0	400-600	8.5-10.0	700-1200
	3.5-4.5	50-100	4.5-5.5	100-200	5.0-6.0	150-250	5.5-7.0	200-400	7.0- 8.5	400- 700
	< 3.5	< 50	< 4.5	< 100	< 5.0	< 150	< 5.5	< 200	< 7.0	< 400


VORTEX
ITALY
Wind Mean Speed Map
Rome
Data source ERA5 downscaled at 3km
Long-term averaged periode 2008-2017
Wind mean speed averaged at 100m height
Wind Mean Speed @ 100m - [m/s]
www.vortexfdc.com | info@vortexfdc.com | 2019 Vortex FdC. Creative Commons (by-sa 3.0)

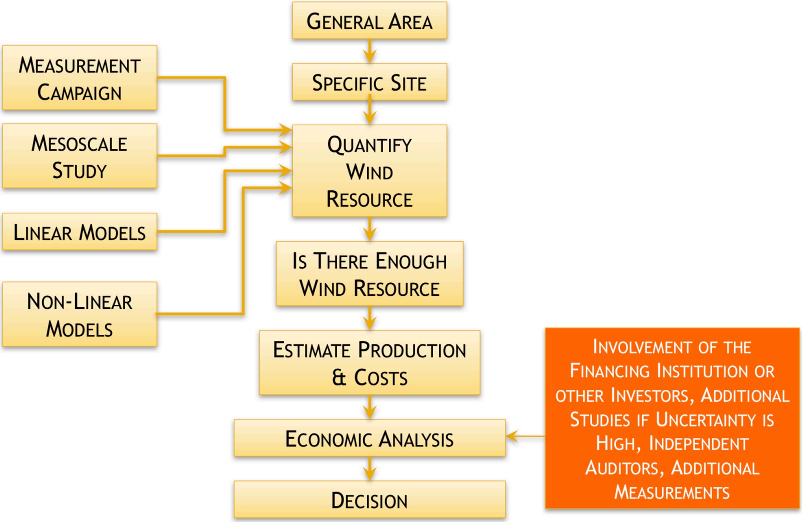

WIND RESOURCE OF ARGENTINA
Mean Annual Wind Speed at 200m Resolution
Mean Annual Wind Speed (m/s)
< 3.00
3.00 - 3.25
3.25 - 3.50
3.50 - 3.75
3.75 - 4.00
4.00 - 4.25
4.25 - 4.50
4.50 - 4.75
4.75 - 5.00
5.00 - 5.25
5.25 - 5.50
5.50 - 5.75
5.75 - 6.00
6.00 - 6.25
6.25 - 6.50
6.50 - 6.75
6.75 - 7.00
7.00 - 7.25
7.25 - 7.50
7.50 - 7.75
7.75 - 8.00
8.00 - 8.25
8.25 - 8.50
8.50 - 8.75
8.75 - 9.00
9.00 - 9.25
9.25 - 9.50
9.50 - 9.75
9.75 - 10.00
> 10.00
This map depicts the approximate annual average wind speed at 200 meter resolution and an 80 meter hub height. It was created by AWS Truepower using proprietary advanced atmospheric models and historical weather data. This map is provided as a general indication of the wind resource, and is not intended for project design. For further information on wind project design and other wind energy consulting services, please contact AWS Truepower.
AWS Truepower
www.awstruepower.com
©2015 AWS Truepower, LLC. All rights reserved.

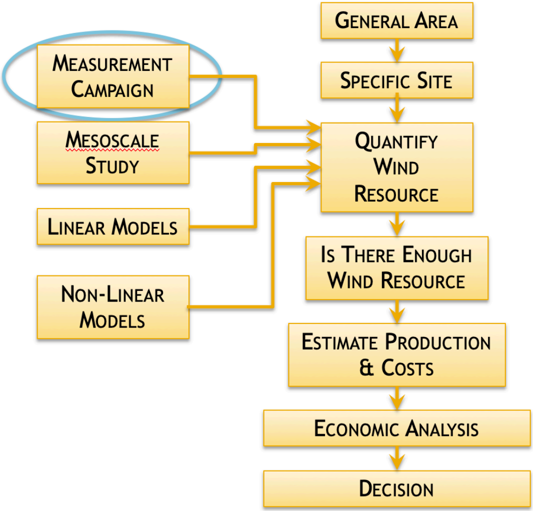

General Area
Specific Site
Access, Building Conditions, Land Leases, Proximity to Electrical Grid, Preliminary Environmental Study
Measurement Campaign
Mesoscale Study
Linear Models
Non-Linear Models
Quantify Wind Resource
Is There Enough Wind Resource
Estimate Production & Costs
Economic Analysis
Decision


General Area
Specific Site
Measurement Campaign
Mesoscale Study
Linear Models
Non-Linear Models
Quantify Wind Resource
Measurements + Models of varying complexity depending on the complexity of the site and wind conditions
Is there enough wind resource
Estimate Production & Costs
Economic Analysis
Decision


GENERAL AREA
SPECIFIC SITE
MEASUREMENT CAMPAIGN
MESOSCALE STUDY
LINEAR MODELS
NON-LINEAR MODELS
QUANTIFY WIND RESOURCE
IS THERE ENOUGH WIND RESOURCE
ESTIMATE PRODUCTION & COSTS
QUANTIFY UNCERTAINTY OF RESULTS TO QUANTIFY ECONOMIC RISK OF PROJECT :: VARIABILITY, QUALITY OF RESULTS, SENSITIVITY TO PARAMETERS
ECONOMIC ANALYSIS
DECISION

PART I - The Need for CFD in Wind Energy Studies (13/ 19)




General Area
Specific Site
Measurement Campaign
Mesoscale Study
Linear Models
Non-Linear Models
Quantify Wind Resource
Is There Enough Wind Resource
Estimate Production & Costs
Economic Analysis
Decision

Typical Measurement Campaigns :: Cup Anemometers

Typical Sampling Rate :: 1 Hz

- Standard cup anemometers and wind vanes
- Measurement campaign with at least 1 year of wind data
- Several anemometer in the same mast allow the measurement of the vertical wind profile
- Several masts with cup anemometers

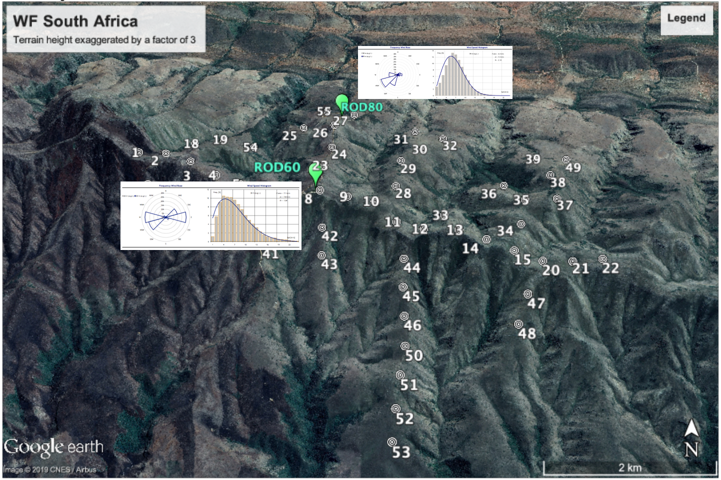
- ADVANTAGES ::

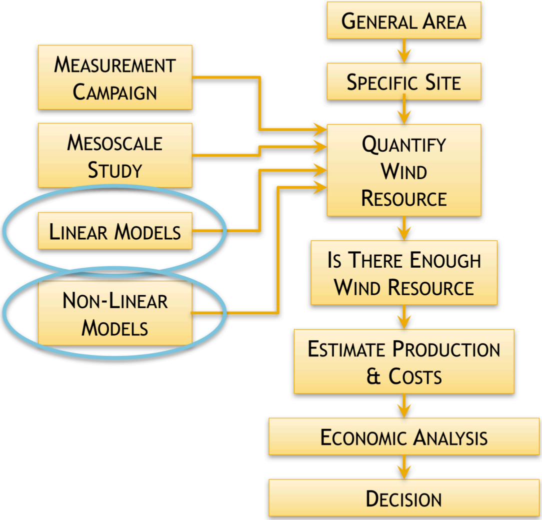
  cheap, well-spread and well-known technology
- DISADVANTAGES ::

  No measurement of the vertical wind component

  Only measure the axial component of turbulence and only with length scales higher than the sampling rate

But measurements are collected at only a few points... We need to know how conditions vary **across the whole wind farm.**




GENERAL AREA
SPECIFIC SITE
MEASUREMENT CAMPAIGN
MESOSCALE STUDY
LINEAR MODELS
NON-LINEAR MODELS
QUANTIFY WIND RESOURCE
IS THERE ENOUGH WIND RESOURCE
ESTIMATE PRODUCTION & COSTS
ECONOMIC ANALYSIS
DECISION

Wind flow modelling in Wind Energy : the beginning

Linear models :: WAsP (Wind Atlas Analysis and Application Program)

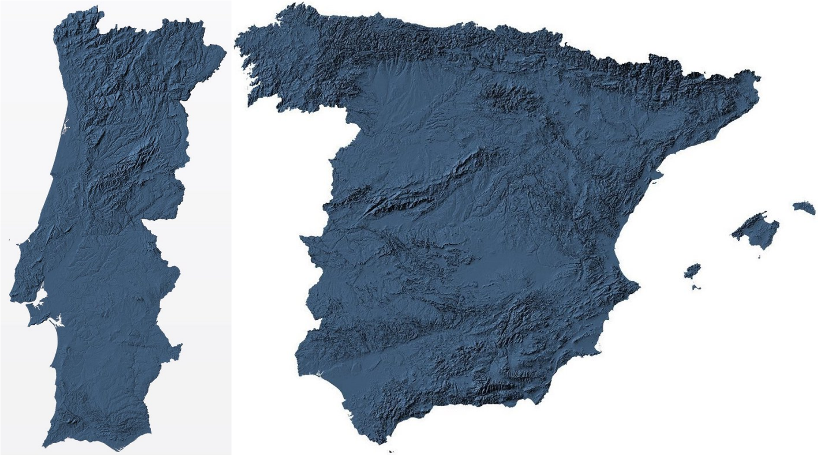
Advantages

- Widely used in the wind industry
- Quick computation and easy to use (graphical user interface, Windows)
- Use the measurements as input
- Good interaction with other known tools in wind energy (ex: WINDFARMER)

Disadvantages

- Do not work well in complex terrain
- Do not model or predict recirculation zones or turbulence
- Do not provide results for the flow inclination

Wind flow modelling in Wind Energy : when linear models are too inaccurate



Non-linear (CFD) models :: Windie™, Meteodyn®, WindSim®, etc.

Disadvantages

- Higher computational requirements and harder to use
- Driven by boundary conditions which are not always easy to specify

Advantages

- Adequate for complex terrain
- Can predict recirculation regions
- Can calculate vertical wind components
- Can model stratification (thermal) effects
- Can model the transient evolution of the wind flow

When to use CFD? :: a proposal of criteria

- Are there Obstacles inside the wind farm area?
- Are there Forests in the vicinity?
- Are there higher terrain features around?
- Is the terrain complex? (RIX $> 40\%$)

- Did measurements register high turbulence values? $> 15\%$
- Do mast data correlate poorly? $> 10\%$

- Measurement Height / Rotor height $< 2/3$
- Large distances between turbines and masts? ($>$ 3km complex & 6km flat)

- Does WAsP perform badly? $> 5\%$

END OF PART I

PART II - Methodology for the simulation of atmospheric flows over complex terrain, using OpenFOAM

- PART II - METHODOLOGY FOR THE SIMULATION OF ATMOSPHERIC FLOWS OVER COMPLEX TERRAIN USING OPENFOAM

  1. Introduction.
  2. Mesh generation.
  3. Boundary conditions for atmospheric flows.
  4. Surface roughness modelling.
  5. Turbulence models.
  6. Post processing and data extraction.
  7. Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill.

PART II - Methodology for the simulation of atmospheric flows over complex terrain, using OpenFOAM (1/2)

Methodology developed under the master thesis:

- Title: "Development of procedures for the simulation of atmospheric flows over complex terrain, using OpenFOAM".
- Orientation:
  - Prof. Fernando Aristides da Silva Ferreira de Castro (ISEP).
  - Prof. José Carlos Pereira Lopes da Costa (ISEP).
- Conclusion in November, 2013.

PART II - Methodology for the simulation of atmospheric flows over complex terrain, using OpenFOAM (2/2)

Limitations and simplification of the developed procedures:

- Steady-state, incompressible and neutrally stratified atmospheric flows.
- Absence of Coriolis force.
- Simulations limited to RaNS (*Reynolds-averaged Navier-Stokes*) k-ε and k-ω models.
- Absence of a canopy model.
- Dry air.

PART II - Mesh generation (1/10)

Preliminary tests:

- *snappyHexMesh*:
  - High computational resources.
  - Difficult to control the size and dimensions of the control volumes.
  - Non-structured mesh.
- *extrudeMesh*:
  - Lower computation resources when compared to *snappyHexMesh*.
  - Triangulation of surfaces (based on STL file).
  - Requires subsequent definition of domain boundaries.

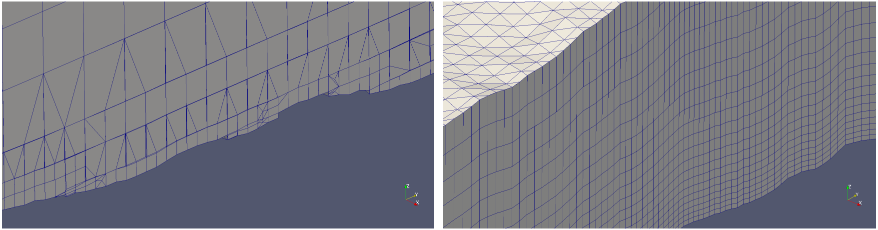


Figura 1: Preliminary testes. Left: *snappyHexMesh*. Right: *extrudeMesh*.

Developed methodology for mesh generation includes two main steps:

1. Manipulation of topographic coordinates with *gsurf* (a part of Windie™ code):
2. Translation of the topographic coordinates to a native OpenFOAM format (*write_blockMeshDict*).
   - Mesh is subsequently generated with the OpenFOAM executable *blockMesh*.

1. Manipulation of topographic coordinates with *gsurf* (a part of Windie™ code).
    - Grade the distance between coordinates, expanding from the point of interest towards periphery.
    - Define the size of computational domain and align it with a predefined direction.
    - User defines (see figure 2):
        - Coordinates of the centre of computational domain (`xcentre ycentre`).
        - Size of control volume in the centre of computational domain (`cofx cofy`).
        - Size of the computational domain (`xmin xmax ymin ymax`)
        - Flow direction (`rot`)
        - Number of coordinates in each orthogonal direction (`nig nij`)

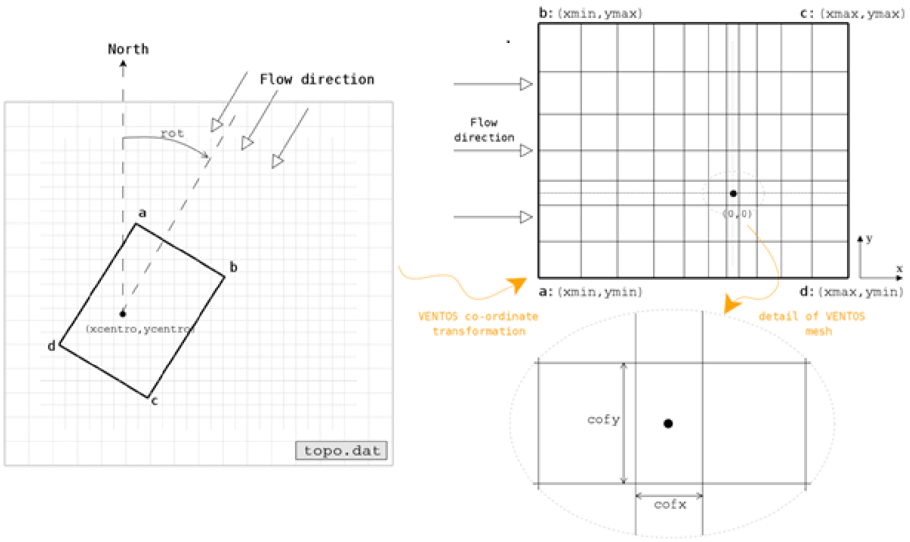


Figura 2: *gsurf* scheme

2. Translation of the topographic coordinates to a native OpenFOAM format (*write_blockMeshDict*):
    - Reads the topographic coordinates previously manipulated with *gsurf* and writes it in the OpenFOAM *blockMeshDict* format file.
    - Mesh is generated using the OpenFOAM *blockMesh* utility.
    - User defines:
        - Height of computational domain.
        - Number of cells in the vertical direction.
        - Expansion ratio R for the dimension of cells in the vertical direction.

write_ blockMeshDict procedure:

- Each 4 points of the coordinates file originate an hexahedron with bottom face shaped to ground surface z data and an horizontal top face at an user defined height.
- Upon run of OpenFOAM *blockMesh* utility, mesh is generated and hexahedrons are split in a predefined number of cells, in which each cell z dimensions expand at a ratio R in vertical direction.
- Domain boundaries are decomposed in six patches:
  - $ground$ to which is assigned the wall attribute.
  - sky, $front$ and $back$ to which is assigned the $symmetryPlane$ attribute.
  - $inlet$ and $outlet$ defined as generic $patch$.

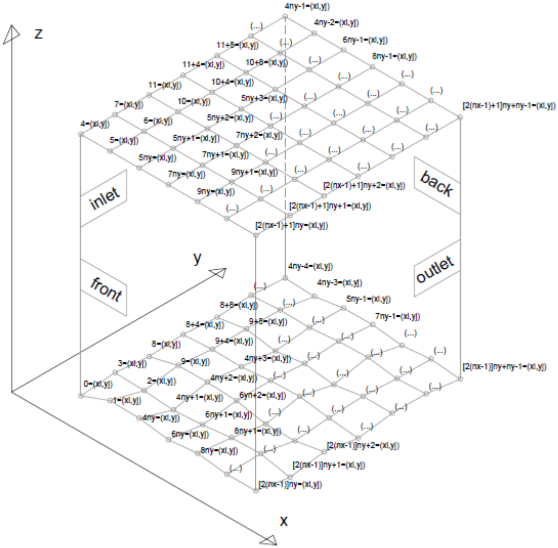


Figura 3: Vertex coding scheme

Example of a mesh generated with *write_ blockMeshDict*:

- Topography with $4000 \times 4000 \times 700$ m decomposed in $76 \times 76 \times 30$ cells.

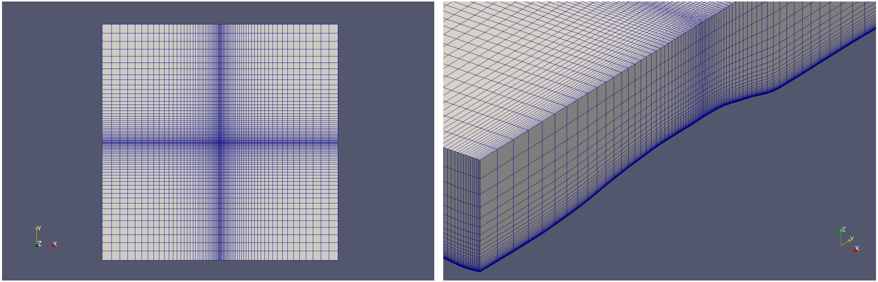
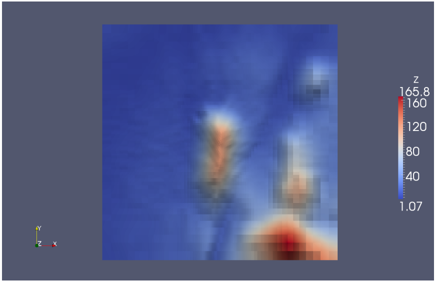


Figura 4: Mesh generated for *Askervein hill*. Left: mesh overview. Right: cross section at centre.

Example of a mesh generated with *write_blockMeshDict* (cont.):

- Topography with $4000 \times 4000 \times 700$ m decomposed in $76 \times 76 \times 30$ cells.



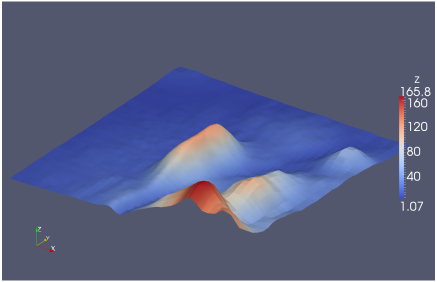


Figura 5: Mesh generated for *Askervein hill*. Ground patch magnified in z.

Important features of *write_ blockMeshDict*:

- Systematic method where the user controls the number and size of control volumes.
- Fully structured mesh.
- Much lower computational resources when compared to *snappyHexMesh*.

Possible limitations and future work:

- Investigate the behaviour of *write_ blockMeshDict* in topographies with steep slopes (e.g. cliffs).

GUIDED EXAMPLE

Mesh generation - Askervein Hill

Figura 6: Google Earth capture of Askervein Hill.

GUIDED EXAMPLE

Mesh generation - Askervein Hill

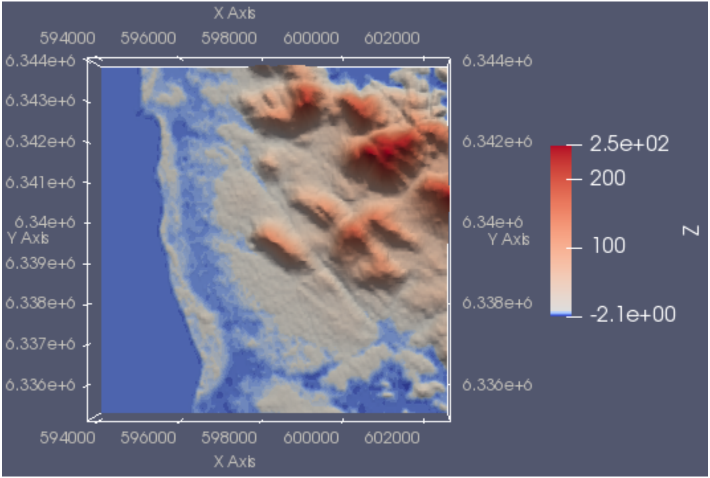


Figura 7: Askervein Hill topography.

Example files are in folder **askervein_OFI19**

1. Manipulation of topographic coordinates with *gsurf* (required files are in subfolder _*preproc/surf(windie)*):
   - Coordinates file *topo.dat* is stored in subfolder *topography*.
   - In file *preproc.cfg* are configured the following variables:

```
#GSURF
nit = 333 !x-dimension of topo data
njt = 333 !y-dimension of topo data
cofx = 10.0 !cofx
cofy = 10.0 !cofy
xmin = -2000.0 !xmin
xmax = +2000.0 !xmax
ymin = -2000.0 !ymin
ymax = +2000.0 !ymax

#MESH ROTATION
rot = 210.0 !wind direction
xcentre = 598178.0 !x-coordinate of site centre
ycentre = 6339596.0 !y-coordinate of site centre
```

PART II - Mesh generation - Guided example (4/5)

- In file *windie.cfg* are configured the following variables:

```
# GRID_DIMENSIONS
nig = 78
njg = 78
```

- Upon run of executable *gsurf* the file *topo_nrot.vtk* is generated and moved to subfolder *_preproc/surf(windie)/tecplot*.

2. Translation of the topographic coordinates to a native OpenFOAM format (required files are in subfolder *_preproc/surf(windie)*:
   - File *topo_nrot.vtk* in folder *_preproc/surf(windie)/tecplot* must be moved by the user to folder *_preproc/vtk*.
   - In file *_preprocDict* are configured the following variables:

```
# write_blockMeshDict #
sky = 700.  # Height ASL of computational domain top patch [m]
nz = 030 # Number of cells in z direction
R = 194.76 # R expansion factor
```

   - Upon run of executable *write_BlockMeshDict* the file *blockMeshDict* is generated and moved to OpenFOAM case folder *system*.
3. Running the OpenFOAM executable *blockMesh* in case parent folder the mesh is generated.

Goal:

- Develop a tool able to write `U`, `k`, `epsilon` and `omega` files based on user defined parameters.

Developed procedure (*write_bCs*):

1. Reading of $inlet$ patch geometric data.
2. Generation of idealized vertical profiles for U, k, ε and ω.
3. Calculation of wall functions parameters and first guesses for the internal field calculation process.
4. Generation of `k`, `epsilon` and `omega` files. Vertical profiles for U, k, ε e ω are stored externally and called via an `#include` instruction.

Note: boundary conditions for turbulent viscosity ν_t will be addressed on surface roughness modelling.

Reading of $inlet$ patch geometric data:

1. In file `boundary` are identified the faces that match $inlet$ patch.
2. In file `faces` are identified the points that define each of the faces.
3. In file `points` is acquired the xyz data of each point.
4. From the z coordinates of each point are built height dependent profiles for U, k, ε and ω, setting height as the distance between ground patch and the averaged z coordinate of each set of four points that define a face.
5. Profiles are then written in a raw ASCII file, being read by OpenFOAM solver following an `#include` instruction.

Inlet vertical profiles:

- U, k, ε (Castro et al., 2010):

$$u = \begin{cases} u_*/\kappa \ln(1 + z/z_0), & z < \delta \\ u_*/\kappa \ln(1 + \delta/z_0), & z \geq \delta \end{cases} \tag{1}$$

$$k = \begin{cases} u_*^2/C_\mu^{1/2}(1 - z/\delta), & z \leq 0.99\delta \\ u_*^2/(100C_\mu^{1/2}), & z > 0.99\delta \end{cases} \tag{2}$$

$$\varepsilon = \begin{cases} C_\mu^{3/4}k^{3/2}/(\kappa z), & z \leq 0.95\delta \\ C_\mu^{3/4}k^{3/2}/(0.95\delta\kappa), & z > 0.95\delta \end{cases} \tag{3}$$

- ω (Wilcox, 1988):

$$\omega = \varepsilon/(\beta^* k) \tag{4}$$

Wall functions:

- k: `kqRWallFunction`
- ε: `epsilonWallFunction`
- ω: `omegaWallFunction`

First guesses for internal field and wall functions:

- k, ε (Richards and Hoxey, 1993):

$$k = \frac{u_*^2}{C_\mu^{1/2}} \tag{5}$$

$$\varepsilon = \frac{u_*^3}{\kappa(z + z_0)} \tag{6}$$

- ω (Wilcox, 1988):

$$\omega = \varepsilon/(\beta^* k) \tag{7}$$

Adopted boundary conditions:

Patch/Field	Attribute	Value	First guess
$\boldsymbol{p}$			
ground, inlet	`zeroGradient;`	-	-
sky, front, back	`symmetryPlane;`	-	-
outlet	`fixedValue;`	`uniform 0;`	-
internal	-	-	`uniform 0;`
$\boldsymbol{U}$			
ground	`fixedValue;`	`uniform (0 0 0);`	-
sky, front, back	`symmetryPlane;`	-	-
inlet	`fixedValue;`	Eq. 1	-
outlet	`inletOutlet;`	`$internalField;`	-
internal	-	-	`uniform (0 0 0);`
$\boldsymbol{k}$, $\boldsymbol{\varepsilon}$, $\boldsymbol{\omega}$			
ground	*Wall function*	-	Eq. 5, 6, 7
sky, front, back	`symmetryPlane;`	-	-
inlet	`fixedValue;`	Eq. 2, 3, 7	-
outlet	`zeroGradient;`	-	-
internal	-	-	Eq. 5, 6, 7

PART II - Boundary conditions for atmospheric flows (6/6)

Application example:

- Serra do Cabeço da Rainha. $u_* = 0.5$ m/s; $\kappa = 0.41$; $z_0 = 0.03$ m; $\delta =$ 2500 m; $\beta^* = C_\mu =$ 0.033.

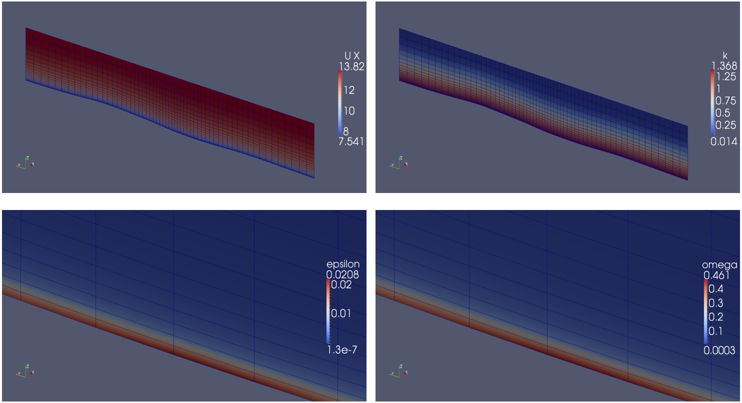


Figura 8: Boundary conditions at *inlet* patch for Serra do Cabeça da Rainha

GUIDED EXAMPLE

Boundary conditions - Askervein Hill

PART II - Boundary conditions - Guided example (2/3)

Example files are in folder **askervein_OFI19**

1. Configuration of boundary conditions (required files are in subfolder *_preproc*):
   - In file *_preprocDict* are configured the following variables:

```
# write_bCs #
zrs = 10.  # Reference height [m]
ustar = 0.607 # Friction velocity [m/s]
z0i = 0.03 # Reference aerodynamic roughness length [m]
delta = 1000.  # Height of boundary layer [m]
kv = 0.41 # vonKarman constant
cmu = 0.033 # kEpsilon model coefficient (betaStar kOmega)
```

2. Upon run of executable *write_bCs* the following files are generated:
   - *epsilon.dat*
   - *k.dat*
   - *omega.dat*
   - *U.dat*

3. Generated *.dat* files are moved to subfolder *0/dat* and called via an instruction in the respective parent file.

   Example for velocity field U file in OpenFOAM case folder 0:

```
inlet
{
type fixedValue;
value nonuniform
(#include "dat/u.dat");
}
```

PART II - Surface roughness modelling (1/4)

Goal:

- Model surface roughness (aerodynamic roughness legnth z_0) in three diferent modes:
  1. Uniform roughness.
  2. Function of topography parameters.
  3. Based on a roughness map.

Developed procedure (*write_z0*):

1. Reading of $ground$ patch geometric data (similar process to the one already performed for $inlet$ patch).
2. Generation of roughness map based on user inputed option.
3. Generation of turbulent viscosity (ν_t) `nut` file. Values of z_0 are called via an `z0 nonuniform List<scalar>` instruction in `nutkAtmRoughWallFunction` wall function.
4. Visual inspection of the roughness map through a scalar `z0` added to folder `<case>/0`.

Generation of roughness map:

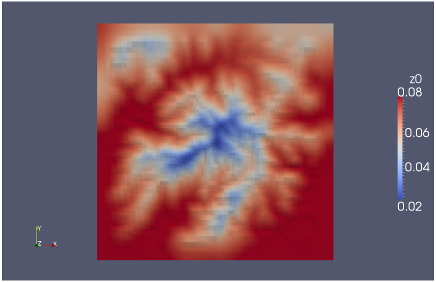
1. Uniform roughness:
   - Added the attribute `z0 uniform $z0u$` in the configuration of the wall function, in which `$z0u$` is the uniform roughness value defined by the user.
2. Function of topography parameters:
   - Function of topography parameters is defined in *write_z0* source code.
   - The value of $z0$ for each face is defined in the configuration of the wall function and called via an `z0 nonuniform List<scalar>` instruction.
3. Based on a roughness map:
   - Real surface roughness data is previously manipulated with *groug* (a part of Windie™ code) similarly to what is performed for elevation data with *gsurf*.
   - Roughness of each face is the averaged $z0$ of each of the 4 points that define the face. The value of $z0$ for each face is defined in the configuration of the wall function and called via an `z0 nonuniform List<scalar>` instruction.

Boundary conditions of turbulent viscosity (ν_t) modelled in nut file:

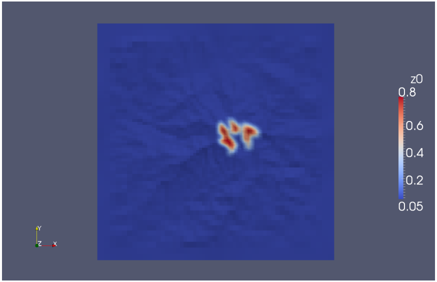
Boundary/Field	**Attribute**	**Value**	**First guess**
$\boldsymbol{\nu_t}$			
ground	`nutkAtmRoughWallFunction;`	-	`uniform 0;`
sky, front, back	`symmetryPlane;`	-	-
inlet, outlet	`calculated;`	-	`uniform 0;`
internal	-	-	`uniform 0;`

Application example:

- Serra do Cabeço da Rainha:



(a) Decrease of $z0$ based on ground elevation.



(b) Site real aerodynamic roughness length.

GUIDED EXAMPLE

Surface roughness modelling - Serra do Cabeço da Rainha

Figura 10: Google Earth capture of Serra do Cabeço da Rainha.

PART II - Surface roughness - Guided example (2/5)

Example files are in folder **cabeco-rainha_OFI19**

1. Manipulation of site roughness coordinates with *groug* (required files are in subfolder *_preproc/surf(windie)*):
    - Roughness coordinates file *roug.dat* is stored in subfolder *topography*.
    - In file *preproc.cfg* are configured the following variables (other variables already configured upon run of gsurf are also required):

```
#GROUG
nir = 3248 !x-dimension of roughness data
njr = 3256 !y-dimension of roughness data
```

    - Upon run of executable *groug* the file *rough_nrot.vtk* is generated and moved to subfolder *_preproc/surf(windie)/tecplot*.

2. Configuration of boundary conditions (required files are in sub folder *_preproc*):
   - File *rough_nrot.vtk* in folder *_preproc/surf(windie)/tecplot* must be moved by the user to folder *_preproc/vtk*.
   - In file *_preprocDict* are configured the following variables:

```
# write_z0 #
z0u = 0.03 # Uniform aerodynamic roughness length [m]
opt = 2   # 1 - Uniform roughness mode # 2 - Roughness map in
rough_nrot.vtk # 3 - Roughness defined by an equation in write_z0.f90
```

3. Upon run of executable *write_z0* the following files are generated:
   - *z0.dat*
   - *z.dat*
   - *nut*
4. Generated *.dat* files are moved to subfolder *0/dat* and called via an `#include` instruction in parent *z* and *z0* files. These files are only used for visual inspection in paraFOAM of terrain elevation (z) and aerodynamic roughness length (z0).

PART II - Surface roughness - Guided example (4/5)

4. Turbulent kinematic viscosity *nut* file is moved to OpenFOAM case folder *0*. Surface roughness is modelled in the wall function parameters acc/ to the following example:

```
ground
{
type nutkAtmRoughWallFunction;
Cmu 0.033;
kappa 0.41;
E 9.8;
z0 nonuniform List<scalar>
3481
(
5.000000E-02
5.000000E-02
()
5.000000E-02
5.000000E-02
);
value uniform 0;
}
```

5. In case the user wants a roughness map based on topography parameters (e.g. ground elevation), it can be coded in *write_z0* source code in file *write_z0.f90* (subfolder *_preproc/src/write_z0.src*) and choose opt = 3 in file *_preprocDict*.

PART II - Turbulence models (1/1)

Goal:

- Automate the configuration of the required turbulence model.

Developed procedure (*write_turbulenceProperties*):

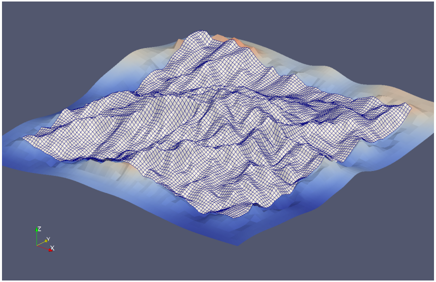
- Generation of file `turbulenceProperties` according to the user defined turbulence model and its parameters.

Considered RaNS models:

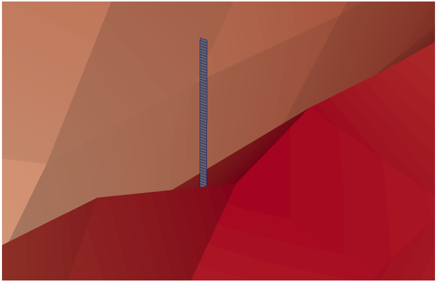
Model	**Formula / Calibration**
Standard $k-\varepsilon$	Launder and Sharma (1974)
Atmospheric $k-\varepsilon$	Beljaars et al. (1987)
Calibrated $k-\varepsilon$	Beljaars et al. (1987), Martinez (2011)
$Realisable\ k-\varepsilon$	Shih et al. (1995)
Original $RNG\ k-\varepsilon$	Yakhot and Orszag (1986)
Modified $RNG\ k-\varepsilon$	Yakhot et al. (1992)
Atmospheric $RNG\ k-\varepsilon$	Kasmi and Masson (2010)
$k-\omega$	Wilcox (1988)
$SST\ k-\omega$	Menter and Esch (2001)

Goal:

- Automate the results extraction in three diferent modes:
  1. Along a surface of constant height above ground level [figure (a)].
  2. Along a vertical mast at predefined coordinates [figure (b)].

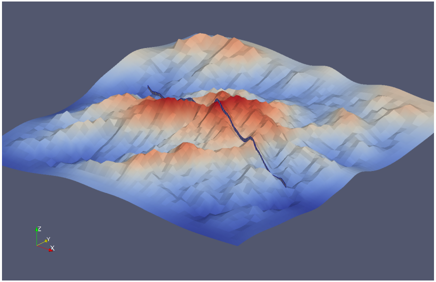


(a)

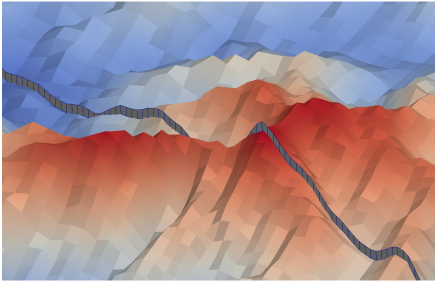


(b)

3. Along a line at a constant height above ground level [figures (a) and (b)].



(a)



(b)

Developed procedure (*write_sample*):

1. Reading of $ground$ patch geometric data (similar process to the one already performed for roughness modelling).
2. Generation of vectors and matrix with the xy component of each coordinate.
3. Bilinear interpolation to calculate the coordinates z component.
4. Configuration of `sampleDict` file for the extraction of the required results for the generated coordinates.
5. Results are extracted recurring to the OpenFOAM instruction `postProcess -func sampleDict -latestTime`.

GUIDED EXAMPLE

Post processing and data extraction - Askervein Hill

Example files are in folder **askervein_OFI19**

1. Configuration of post processing and data extraction (required files are in subfolder *_preproc*):
   - In file *_preprocDict* are configured the following variables:

```
# write_agl #
wd = 210.  # Wind direction [ž]
ang = 223.  # Angle to inspect (from North - before gsurf) [ž]
zagl = 10.  # Height above ground level of line agl [m]
npaglvec = 100 # Size of vector that defines line agl
agloffsetx = 0.  # Offset in x direction of line agl [m]
agloffsety = 0.  # Offset in y direction of line agl [m]
npvprfl = 100 # Size of vectors that define vertical profiles vprfl#
zmast = 300.  # Height of vertical profiles above ground level [m]
npaglsurf = 100 # Size of matrix aglsurf (npaglsurf x npaglsurf)
bound = 0 # Bounding of aglsurf:  0 - Manual; 1 - Auto
xmin = -1000.  # x and y minimum values of aglsurf [m]
xmax = +1000.  # x and y maximum values of aglsurf [m]
nmasts = 2 # Quantity of vertical profiles (masts)
mastcoords:
0.  0.
-63.  -393.
```

2. Upon run of executable *write_sample* the following files are generated and moved to subfolder *_postproc/dat*:
   - *aglsurf.dat* (coordinates of surface above ground level mode 1).
   - *vprfl1.dat* (coordinates of mast 1 mode 2).
   - *vprfl2.dat* (coordinates of mast 2 mode 2).
   - *agl.dat* (coordinates of line above ground level mode 3).
   - *al.dat* (coordinates ground level of line agl mode 3).
3. Upon run of executable *write_sample* the following files are also generated and moved to subfolder *_postproc/vtk* for visual inspection in paraFoam:
   - *aglsurf.vtk* (coordinates of surface above ground level mode 1).
   - *vprfl1.vtk* (coordinates of mast 1 mode 2).
   - *vprfl2.vtk* (coordinates of mast 2 mode 2).
   - *agl.vtk* (coordinates of line above ground level mode 3).

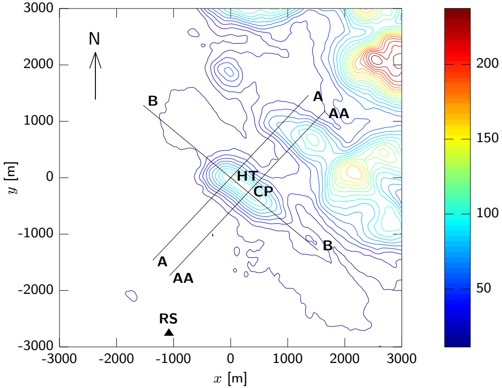
PART II - Post processing - Guided example (4/4)

4. Extraction of results for the points in *.dat* files require a configuration of *sampleDict* file acc/ to the supplied example.
5. Running the OpenFOAM executable *postProcess -func sampleDict -latestTime* in case parent folder the the results are extracted and stored in subfolder *postprocessing/sampleDict*.

PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (1/10)

Askervein Hill (located in the Outer Hebrides of Scotland):

- Nearly elliptical shape in plan view (1 km in minor axis and 2 km in major axis).
- Maximum height in HT of 116 m above surrounding terrain.



PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (2/10)

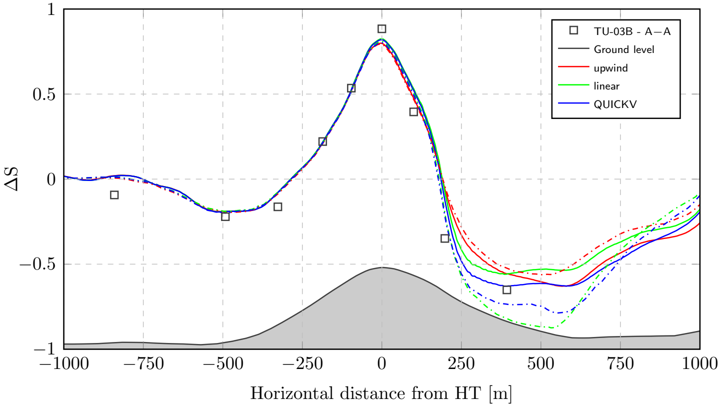
Goals:

- Evaluation of the merit of the developed procedures.
- Evaluation of the sensibility of the solution to different:
  - Convection schemes.
  - Mesh dimension.
  - Turbulence models.
  - Roughness conditions (not shown here).

PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (3/10)

Results for different convection schemes in velocity field U:

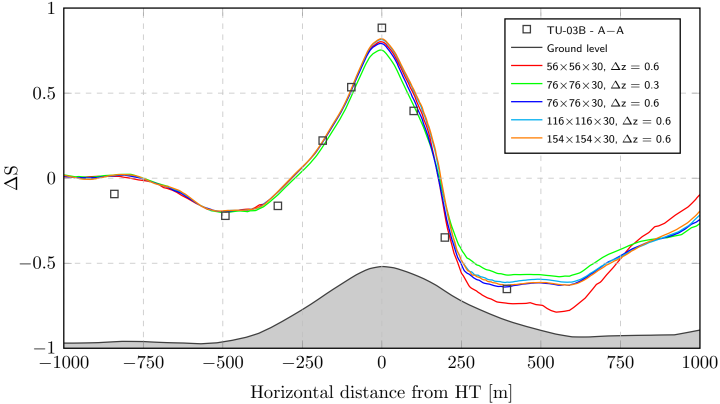
- *Speed-up* along line A−A 10 m above ground level:
  - Atmospheric $k-\varepsilon$; uniform roughness: $z_0 = 0.03$ m.
  - Two mesh sizes: $56 \times 56 \times 30$ (− · −); $154 \times 154 \times 30$ (——).



PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (4/10)

Results for different mesh sizes:

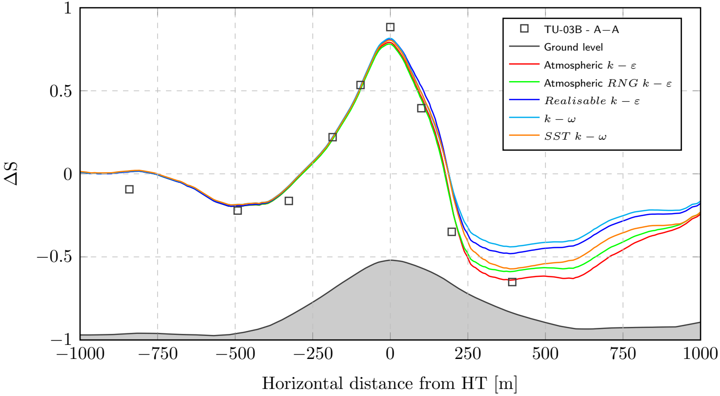
- *Speed-up* along line A−A 10 m above ground level:
  - Convection scheme *QUICKV* for velocity field U.
  - Atmospheric $k-\varepsilon$; uniform roughness: $z_0 = 0.03$ m.



PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (7/10)

Results for different turbulence models:

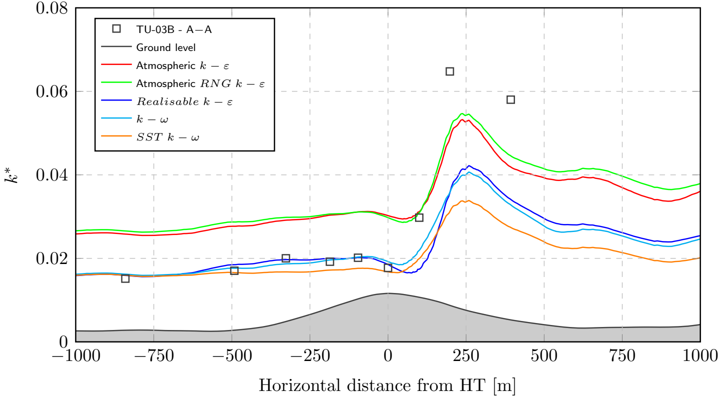
- *Speed-up* along line A−A 10 m above ground level:
  - Convection scheme *QUICKV* for velocity field U.
  - Mesh size: $76 \times 76 \times 30$; uniform roughness: $z_0 = 0.03$ m.



PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (8/10)

Results for different turbulence models (cont.):

- k^* along line A−A 10 m above ground level:
  - Convection scheme *QUICKV* for velocity field U.
  - Mesh size: $76 \times 76 \times 30$; uniform roughness: $z_0 = 0.03$ m.



PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (9/10)

Computational parameters:

- Mainstream laptop with a `Intel Core i5 M430 @ 2.27 GHz` processor without parallelization.
- Convergence criteria of the solution:
  - Pressure field: 1×10^{-2}.
  - Remaining fields: 1×10^{-3}.

CPU time for mesh generation:

- From 5 to 270 s.

CPU time for solution (with solver *simpleFoam*) :

- From 205 to 2175 s.

PART II - Example of the simulation of a neutrally stratified ABL flow over the Askervein Hill (10/10)

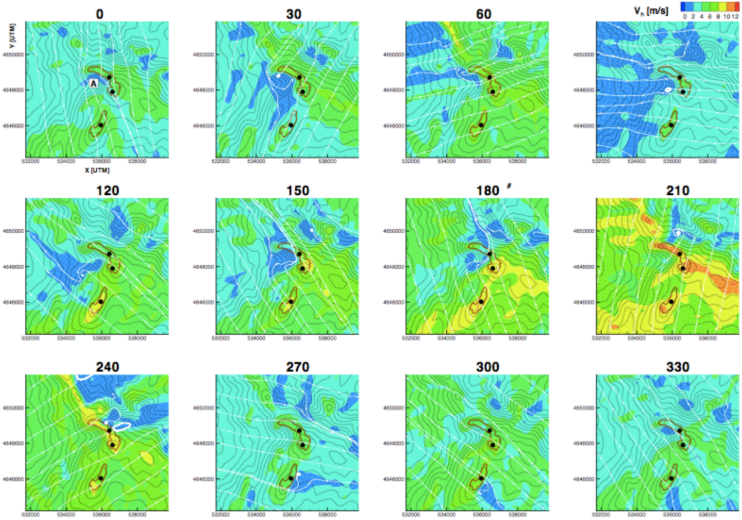
Main conclusion from the performed simulations:

- Good agreement of *speed-up* in hill top and downstream.
- Good agreement of k^* in hill top and upstream and grossly predicted at the downstream.
- Sensitivity analysis for the tested parameters (convection schemes, mesh size, roughness conditions and turbulence models) considered to be globally in line with the simulations presented in the reviewed bibliographic sources.

END OF PART II

- PART III - BRINGING IT ALL TOGETHER
  1. Synthesis: CFD results as a transfer function.
  2. Resource assessment results.
  3. Site assessment results.
  4. Conclusions and Future Work.

PART III - Bringing It All Together (2/ 13)



[
Simulação 0°

Simulação 30°

Simulação ...

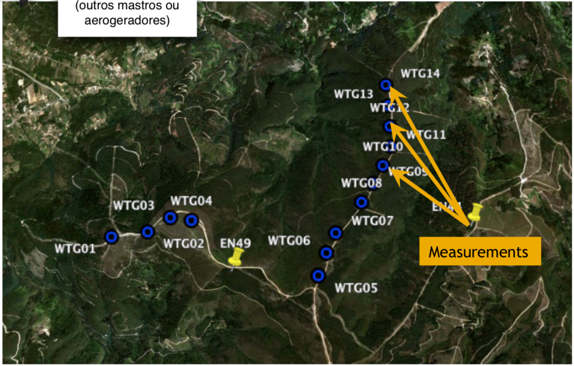
Simulação 330°
]

+

Coordenadas do Mastro de Referência e dos pontos de interesse

→

Matriz de Transferência entre Mastro de Medição e pontos de interesse (outros mastros ou aerogeradores)



Simulação 0º

Simulação 30º

Simulação ...

Simulação 330º

+

Coordenadas do Mastro de Referência e dos pontos de interesse

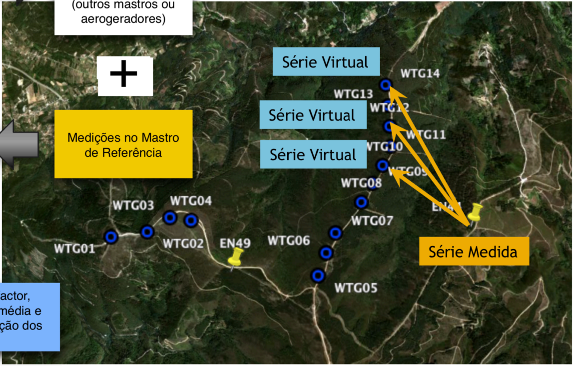
Matriz de Transferência entre Mastro de Medição e pontos de interesse (outros mastros ou aerogeradores)

+

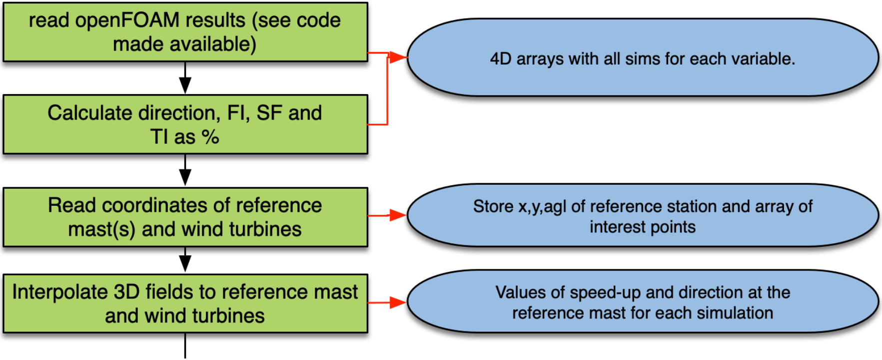
Medições no Mastro de Referência

Série Virtual nos **aerogeradores**

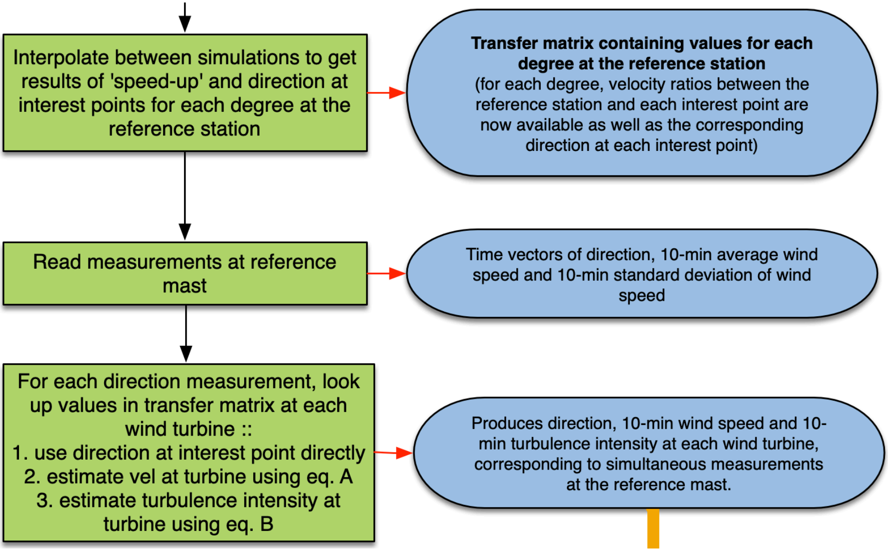
Permite cálculo da turbulência, shear factor, inclinação do escoamento, velocidades média e mínima e parâmetros de Weibull na posição dos aerogeradores



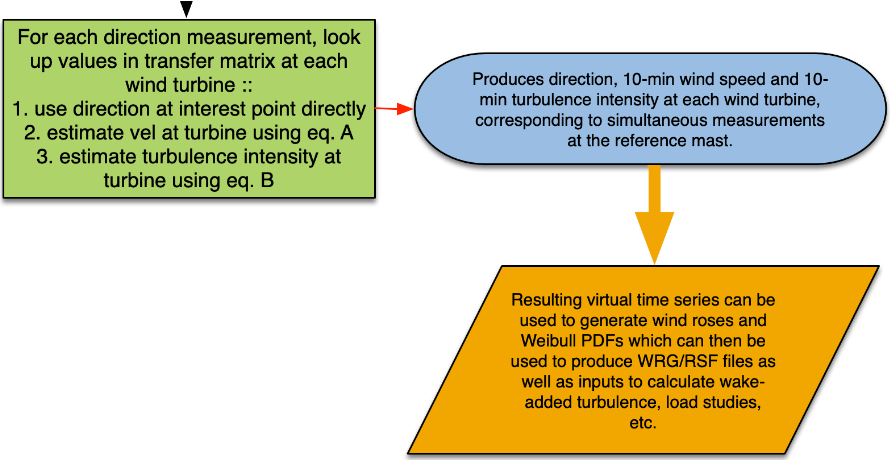
Synthesis flowchart (1/3)



Synthesis flowchart (2/3)



Synthesis flowchart (3/3)



PART III - Bringing It All Together (8/ 13)

The synthesis equations - velocity & turbulence

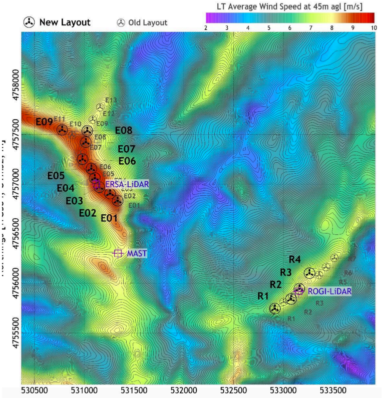
$$V'_{turb} = V_{meas,mast} \frac{V_{CFD,turb}}{V_{CFD,mast}}$$

$$I'_{turb} = I_{CFD,turb} + \Delta I_{turb} \frac{V_{CFD,mast}}{V_{CFD,turb}}$$

REFERENCE MAST		TARGET TURBINE		TRANSFER FUNC
DIRECTION	VH	DIRECTION	VH	V_WTG1 / V_MAST
0	3.152	350.2	3.773	**1.197**
2	3.105	353.1	3.806	**1.226**
4	3.059	356.3	3.839	**1.255**
6	3.013	359.6	3.872	**1.285**
8	2.966	3.1	3.905	**1.317**
10	2.920	6.7	3.939	**1.349**
12	2.874	10.5	3.972	**1.382**
14	2.827	14.4	4.005	**1.417**
16	2.781	18.3	4.038	**1.452**
18	2.735	22.2	4.071	**1.488**
20	2.689	26.0	4.097	**1.524**
22	2.656	27.5	4.009	**1.509**
24	2.622	29.1	3.921	**1.495**
26	2.589	30.7	3.832	**1.480**
28	2.556	32.4	3.744	**1.465**
30	2.522	34.2	3.656	**1.450**
32	2.489	36.0	3.568	**1.434**
34	2.456	38.0	3.480	**1.417**

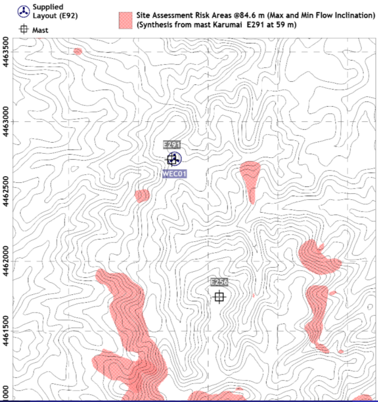
Where to put the turbines: RSF/WRG files

- Virtual time series are processed to calculate Weibull PDF (global and per sector, as well as frequency of each wind direction)
- Production of RSF files for AEP calculation
- Production of WRG for resource mapping and layout optimisation



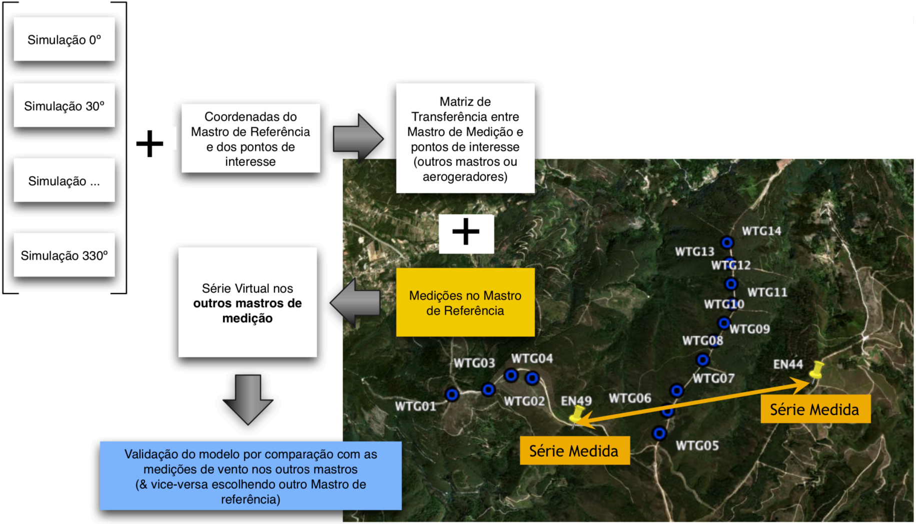
Where NOT to put the turbines: Site Assessment quantities

- avoid areas of high turbulence intensity
- high wind shear
- high flow inclination
- excessive wind speed and gusts must also be considered



PART III - Bringing It All Together (12/ 13)

Cross-predictions to assess model performance and quantify uncertainty



Conclusions & Future Work

Other topics to explore:

- inclusion of stratification
- boundary conditions from mesoscale models instead of idealized profiles
- inclusion of OpenFOAM forest canopy model (Lopes da Costa 2006)

- Feel free to make use of the tools supplied to run through the guided examples.
- Additional maps and mast data supplied for you to design a wind farm and determine wind conditions at the turbines.

Thank you for your attention.