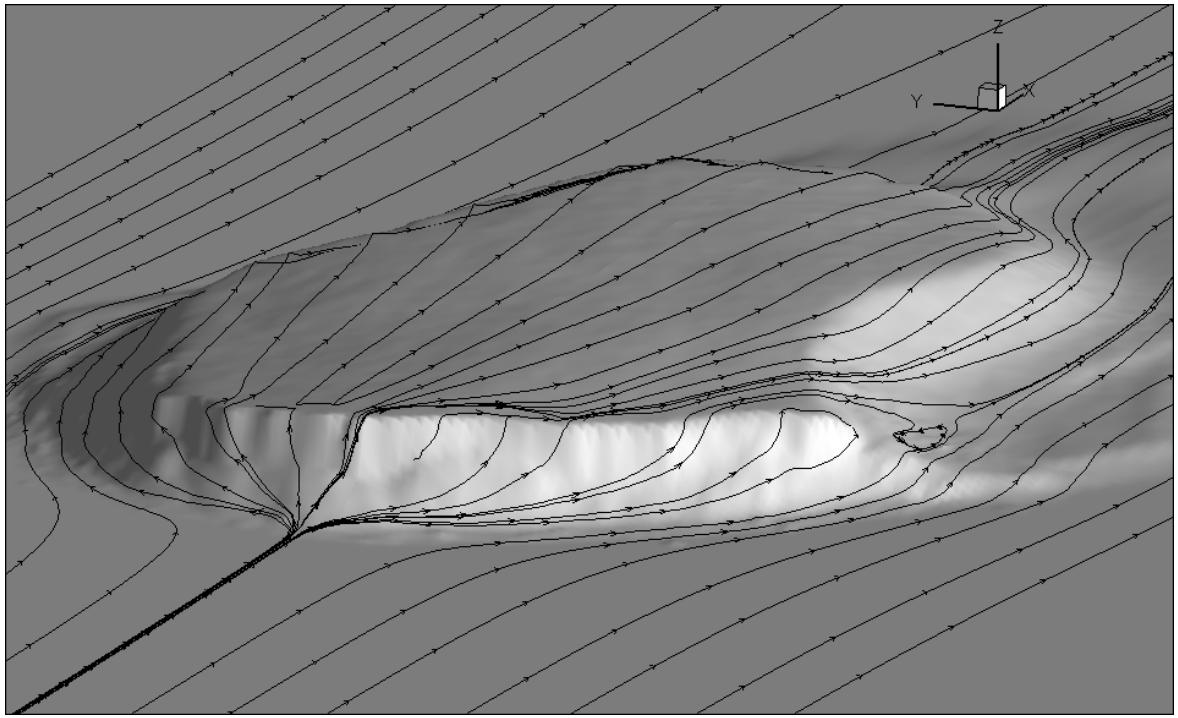
RANS CFD simulations of flow around Bolund

Niels N. Sørensen and Andreas Bechmann



Risø DTU
National Laboratory for Sustainable Energy

Outline

- CFD solver for terrain flow
  - Turbulence modeling
  - Boundary conditions
  - Roughness
- Computational domain and surface discretization
- Brief overview of present simulations, number of points and computing time
- Verification of the simulations, Convergence and Grid Convergence.
- Problems when comparing with measurements.
- Terrain resolution.
- A few examples of the results
- Conclusion

Components of a CFD methodology

- Preprocessor
  - Geometry processor (CAD)
  - Grid Generation
  - Specification of Boundary Conditions
  - Roughness treatment
- CFD solver
  - Accurate
  - Efficient solver
  - Versatile
- Postprocessor
  - 3D graphics
  - Extraction of velocities, turbulence etc in predefined points

Components of a CFD solver

The basic idea is to take the partial differential equations describing the fluid flow, transform them into a set of algebraic equations, and solve these using a numerical method on a computer.

Typical components of a CFD code are listed below:

- Mathematical Model
  - Turbulence Modeling
- Coordinate and basis vector systems
- Discretization Method, space and time
- Solution Method
- Computational Grid

Turbulence Modeling

- Direct Numerical Simulation (DNS)

- Filtered Equations
  - Large Eddy Simulation (LES)

- Time Averaged Equations, Reynolds Averaged Navier-Stokes(RANS)
  - Algebraic Models (e.g. Baldwin-Lomax)
  - One Equations Models (e.g. Spalart-Allmaras, Baldwin-Barth)
  - Two Equation Models (e.g. k-ω, k-ε)
  - Reynolds Stress Models

- Hybrid Models
  - Detached Eddy Models

Boundary conditions (Inlet and outlet conditions)

Inflow boundary conditions for Atmospheric flows:

Log-law profiles for the velocities and turbulent quantities.

$$U(z) = \frac{u_\tau}{\kappa} \ln\left(\frac{z}{z_0}\right) \quad , \quad \mu_t = \rho \kappa u_\tau z \ ,$$

$$\epsilon(z) = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa z} \quad , \quad k(z) = \frac{u_\tau^2}{\sqrt{C_\mu}} \ .$$

$$C_{\epsilon 1} = C_{\epsilon 2} - \frac{\kappa}{C_\mu^{\frac{1}{2}} \sigma_\epsilon} \ .$$

Outflow boundary conditions:

Fully developed flow is assumed in the mesh direction normal to the outlet.

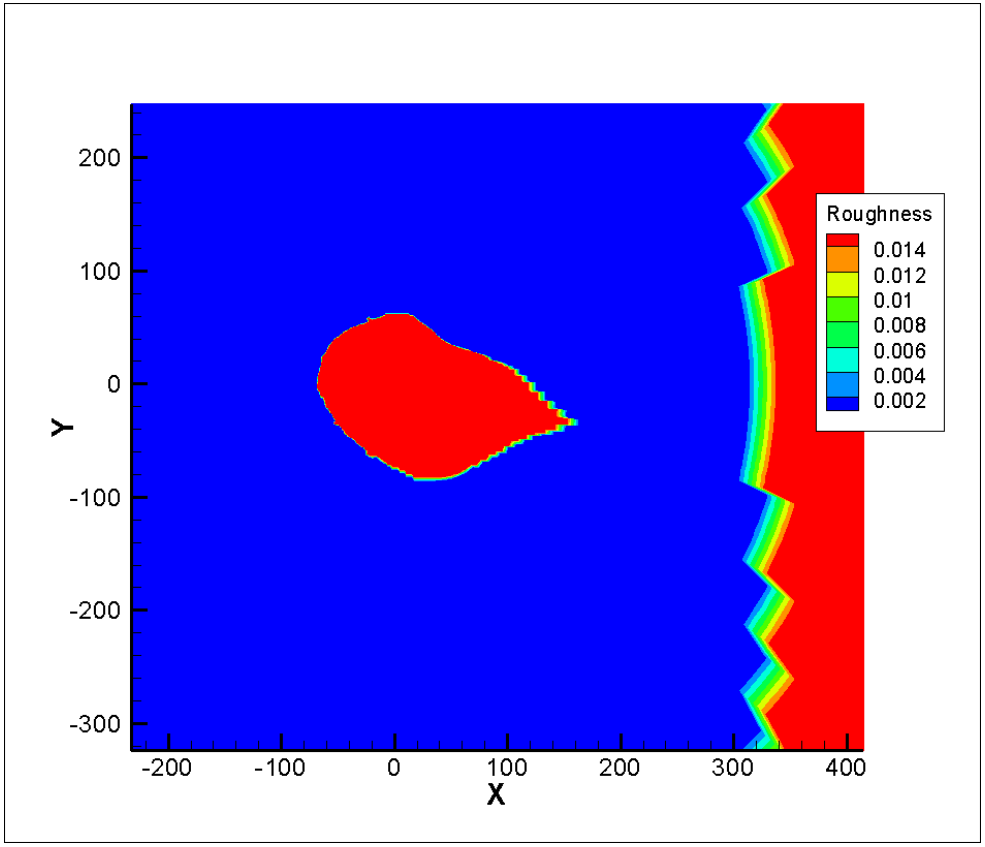
Boundary conditions (Wall)

Wall boundary conditions are given by the log-law

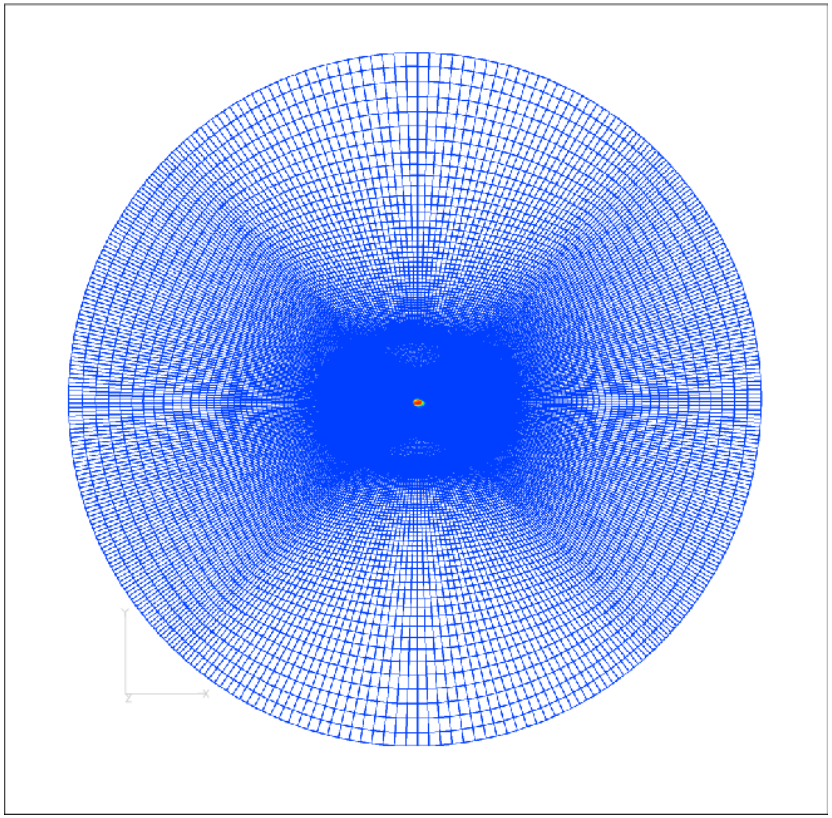
- The velocity boundary conditions are implemented through the friction at the wall.
- The implementation assures that flow separation can be handled by evaluating the friction velocity from the turbulent kinetic energy.
- The computational grid is placed on top of the roughness elements, and the actual roughness heights are ignored in connection with the grid generation.
- The TKE boundary condition is an equilibrium between production and dissipation, implemented through a von Neumann condition and specifying the production term from the equilibrium between production and dissipation.
- The epsilon equation is abandoned at the wall, and instead the value of the dissipation is specified according to the equilibrium between production and dissipation.

Roughness Maps

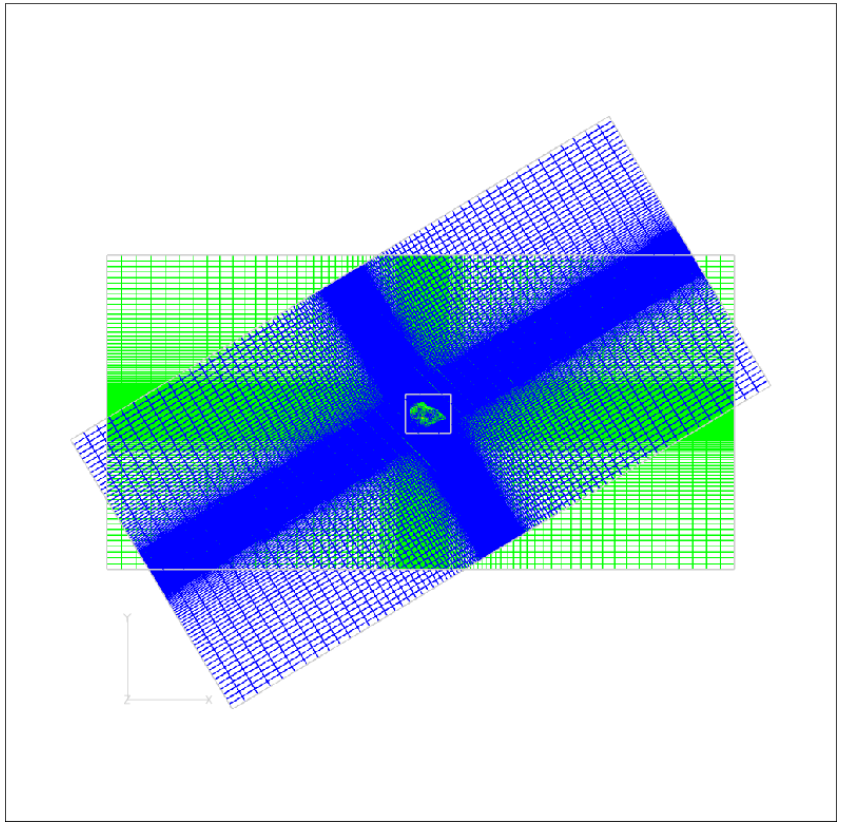
- In our case, and maybe in most cases the local roughness is determined based on the (x,y) coordinats. This may be a problem for roughness shifts along vertical slopes.



Computational Domain (1)



One domain for all comp.



A dedicated mesh for each direction

Computational Domain (2)

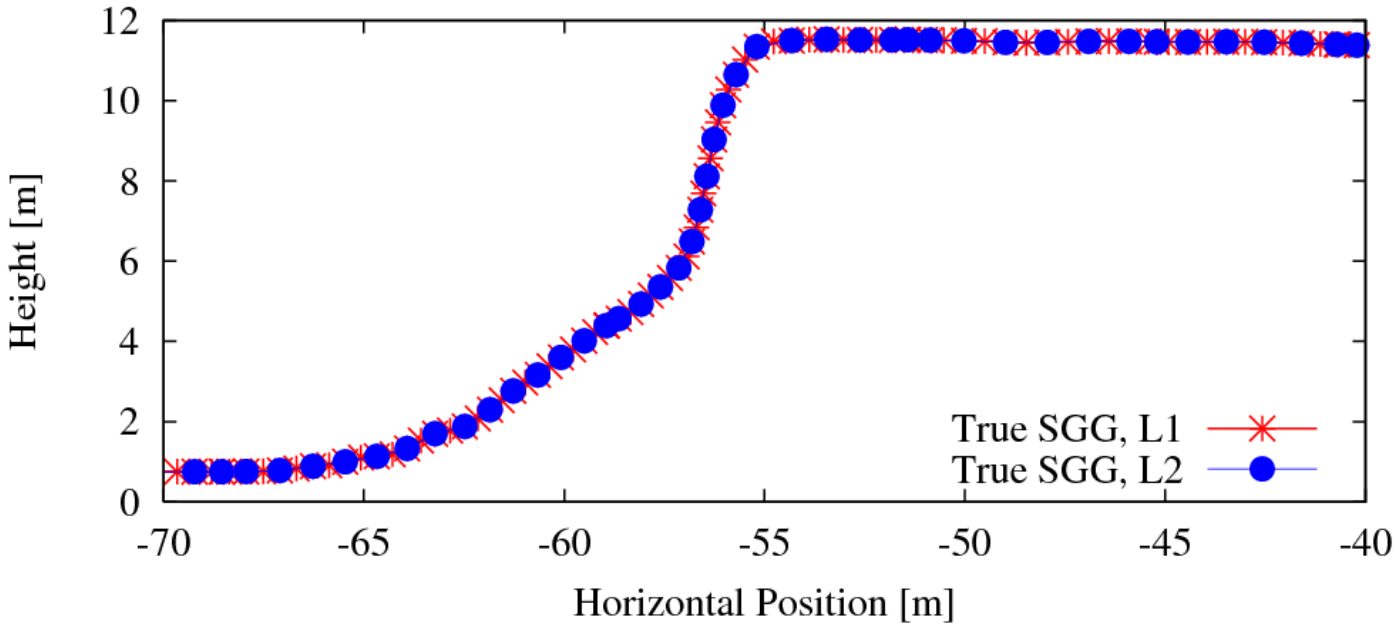
- Typically we have a problem of where to specify boundary conditions, especially inflow.
  - For Bolund this is not an issue
- Solutions?
  - Make a very large domain specify simple conditions at inlet

    Expensive or requires a zooming grid
  - Obtain the inflow conditions from external means
    - Measured values
    - Nested computations
    - Mesoscale model

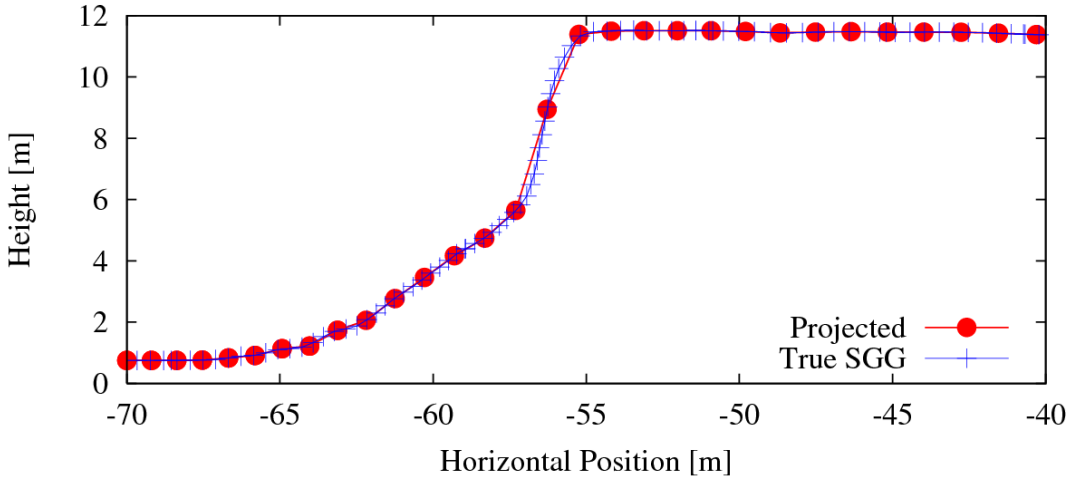
    Often the measurements or computations will not have sufficient resolution.
- For domains dedicated to specific flow directions symmetry conditions are often used at the side ‘walls’. This may make them useless for studies with slightly different flow direction.

Surface Resolution (1)

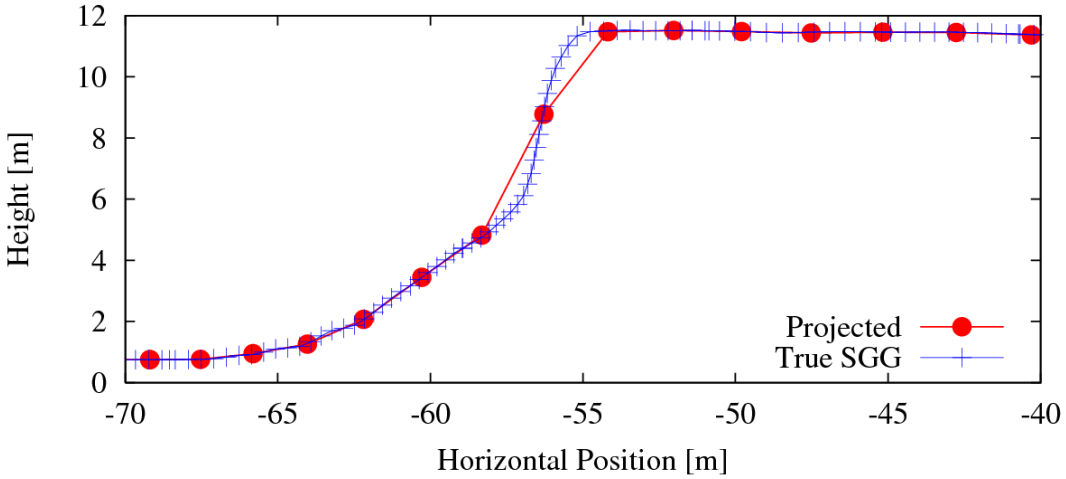
- Using a true surface grid generation on the terrain surface, will allow good resolution of steep gradients. As seen below this allows good resolution even on level 2 and 3 grids.



Surface resolution (2)
The problem of projected grids

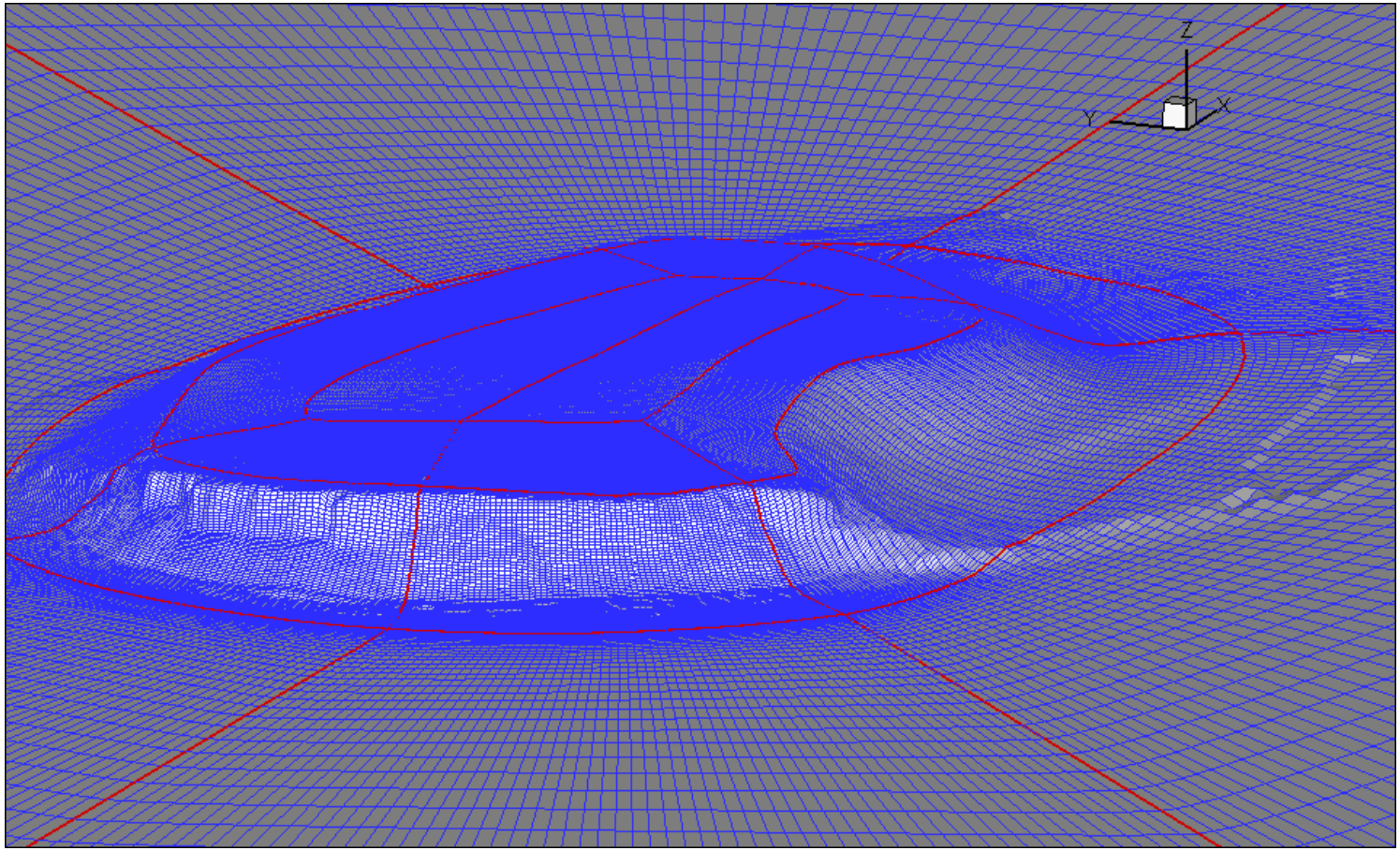


Using just simple projection of a 2D surface grid onto the terrain, will naturally lead to coarse cells at steep slops in the terrain.

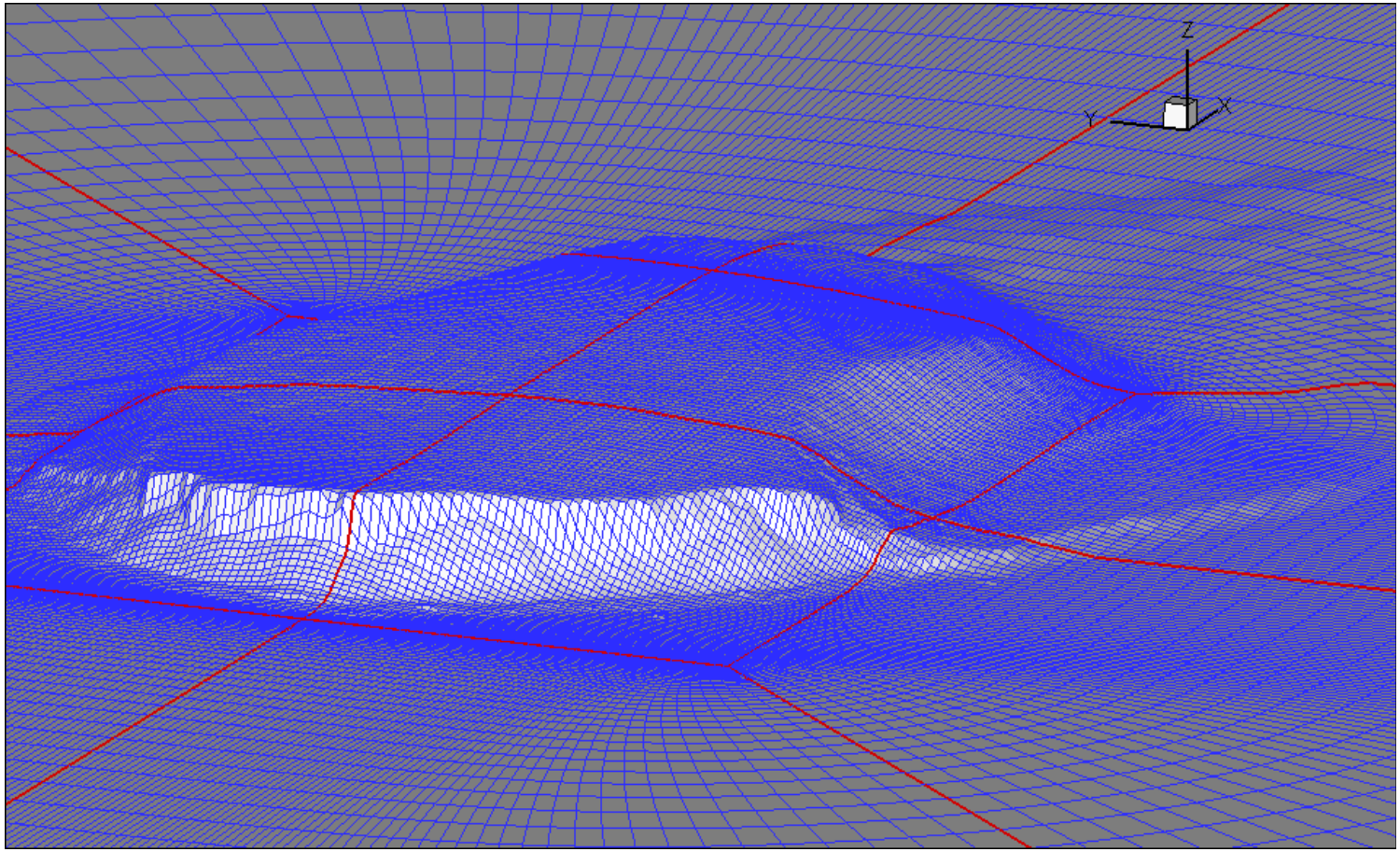


The grids are not well suited for grid convergence studies.

Surface Grid (1)



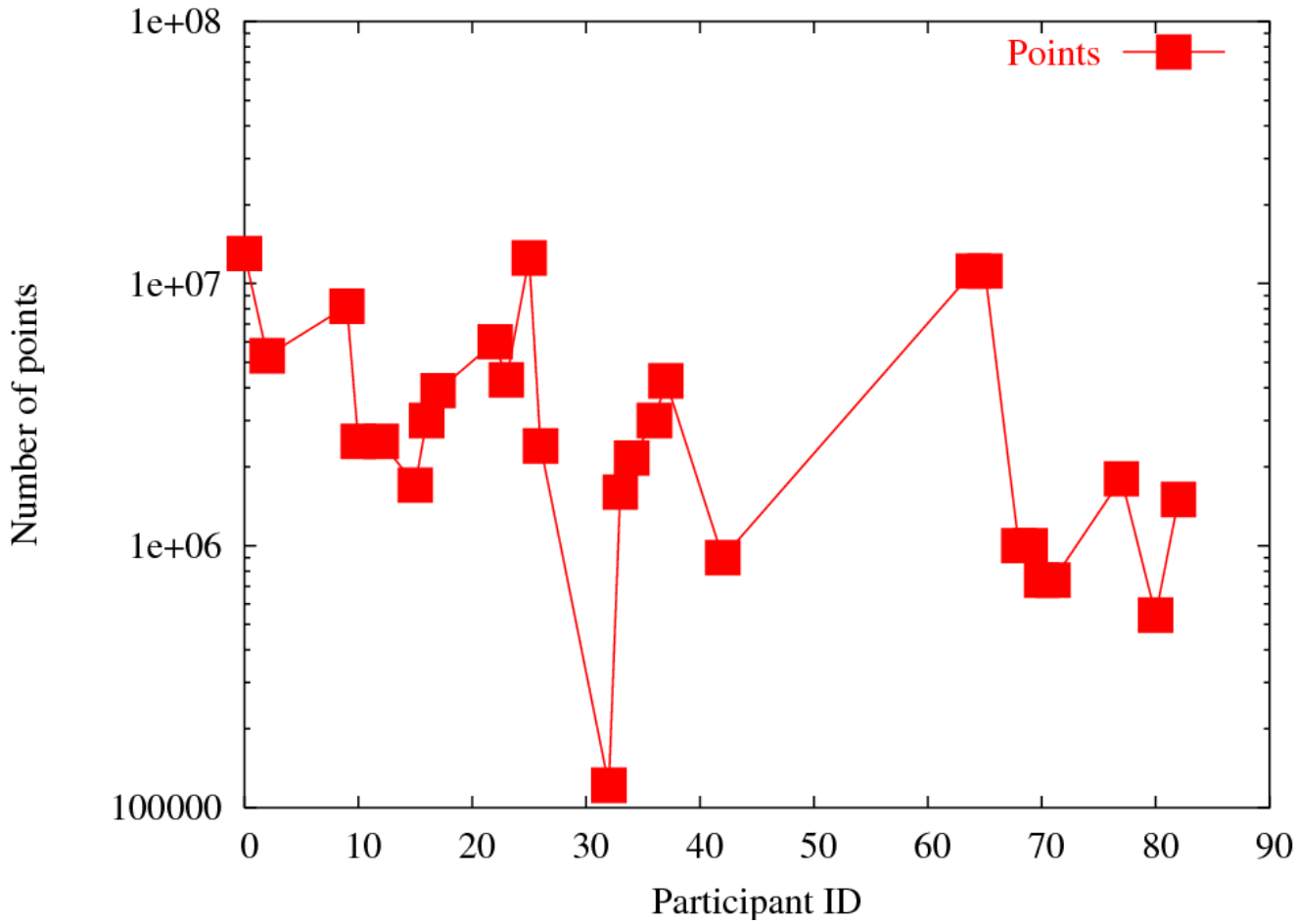
Surface Grid (2)



Is the digital terrain correct

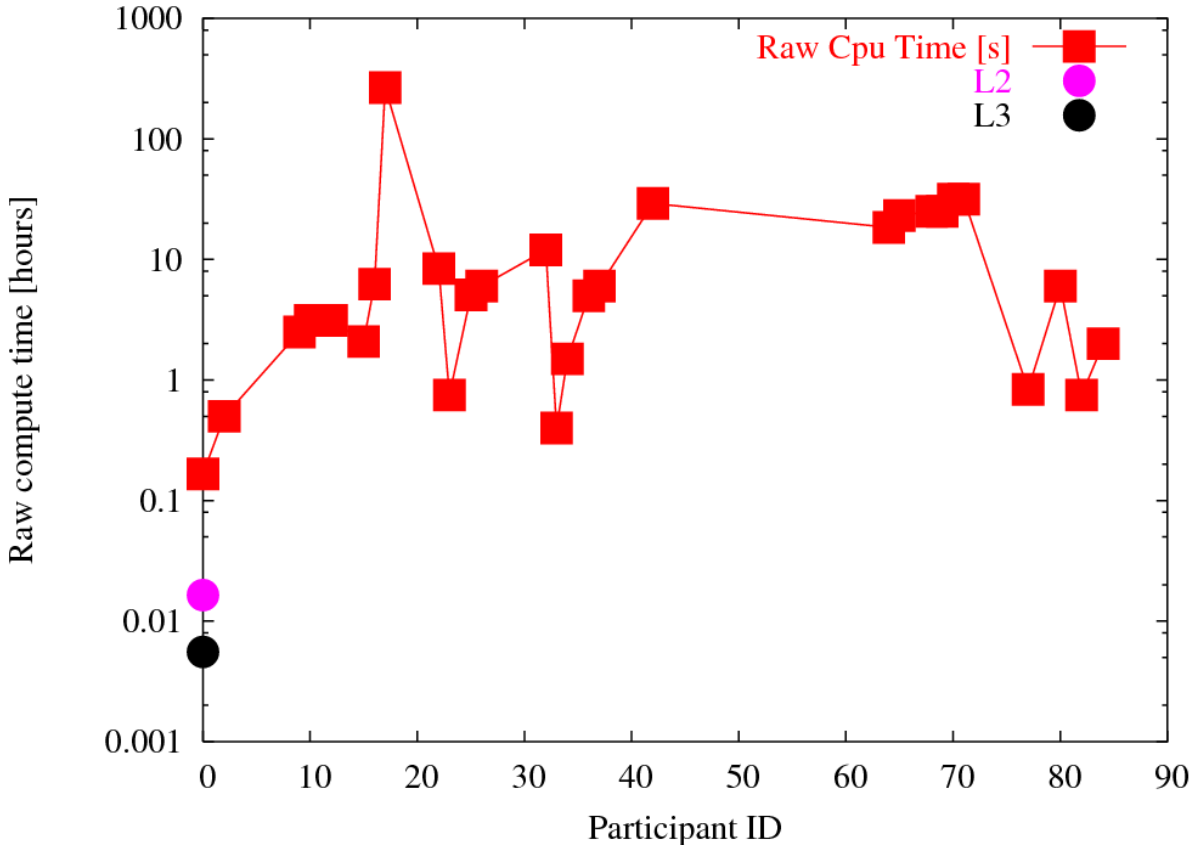
- And how good is the roughness estimation?

Number of points used for the simulation

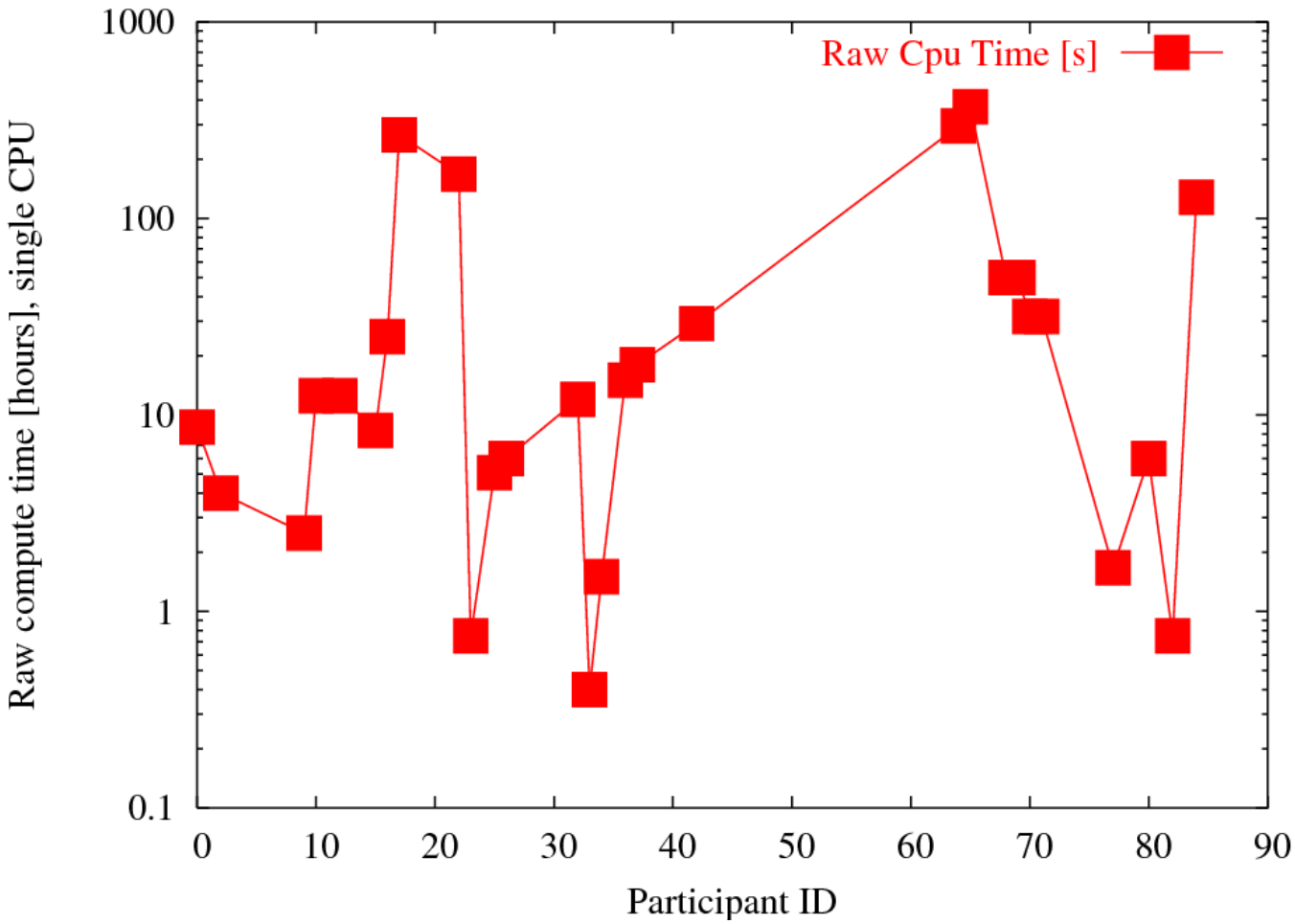


Turnaround time

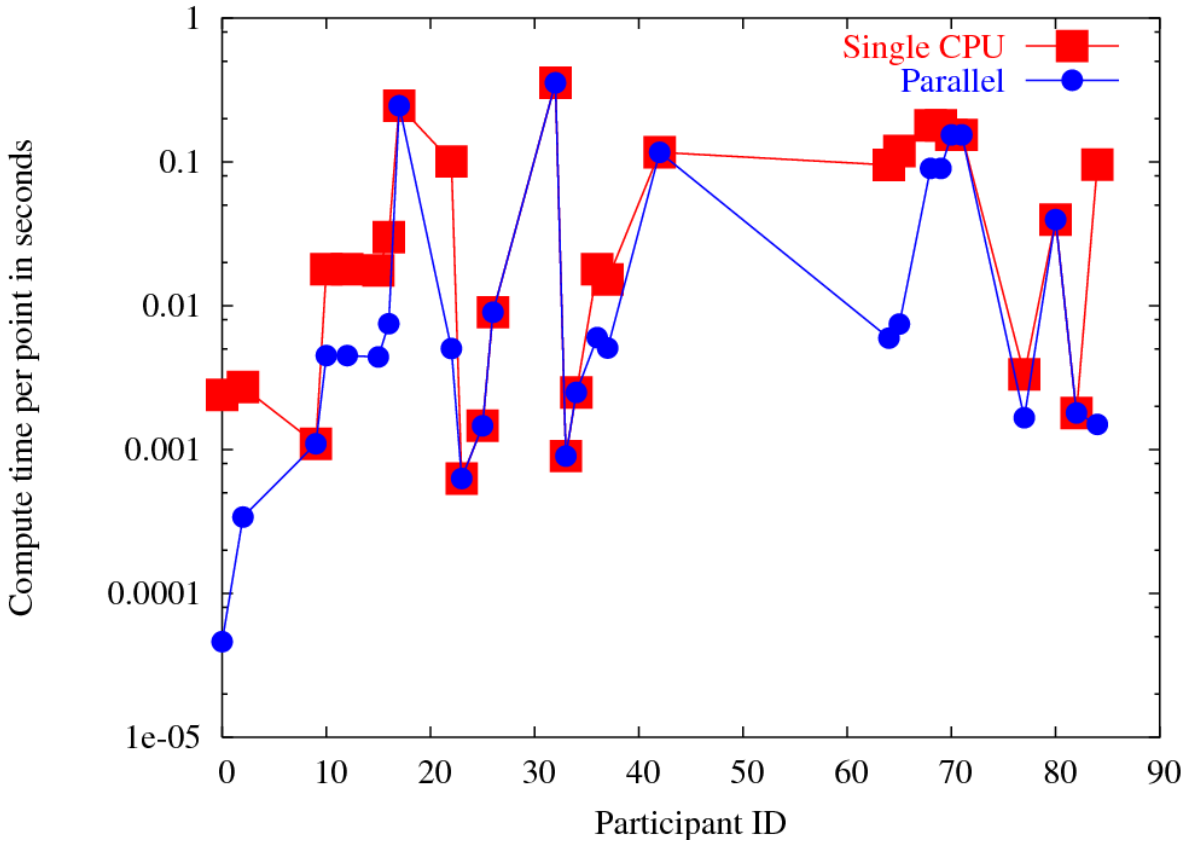
Computing time may be an issue compared to linear models



Computing time on a single Cpu



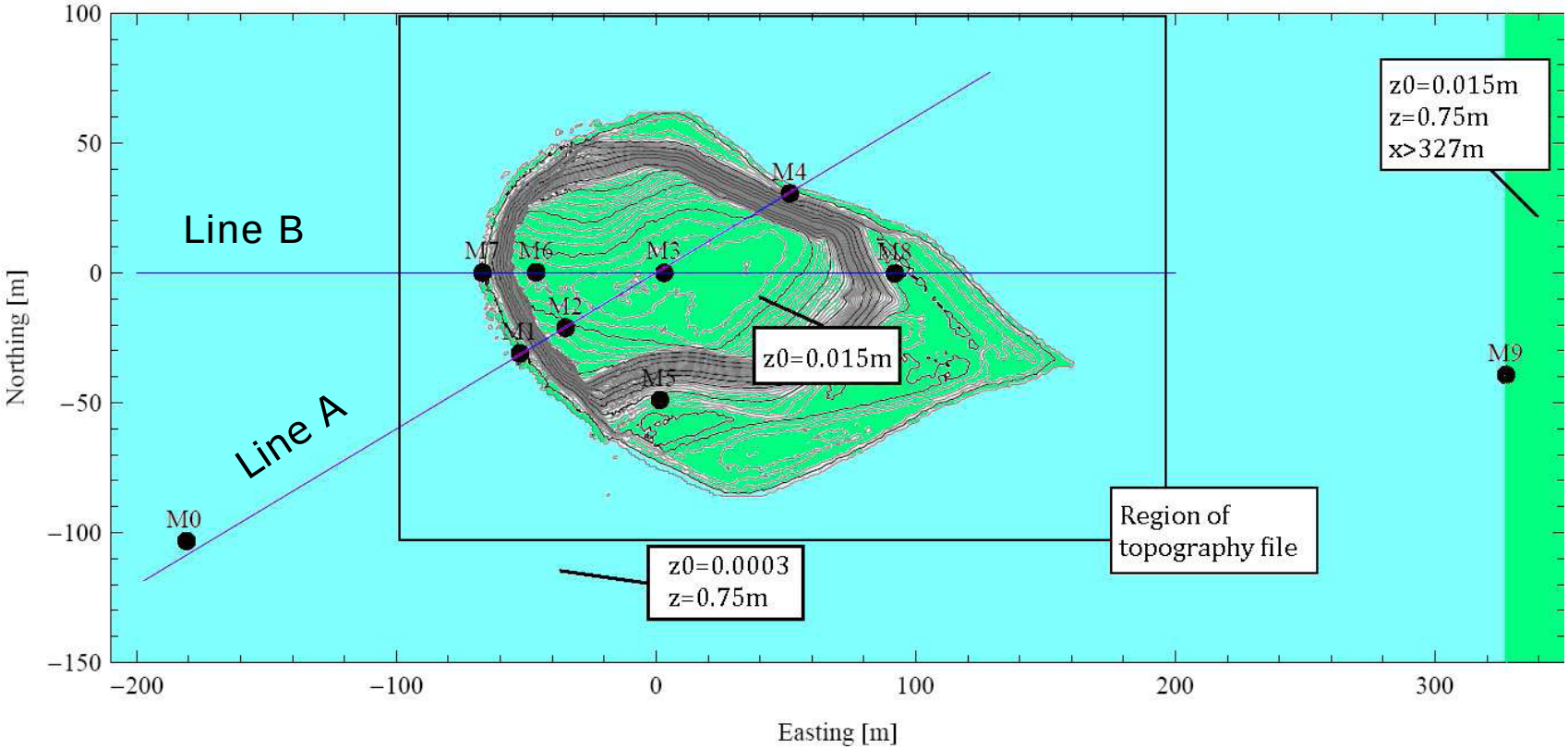
Computing Time



There is a speed difference of around 50-500 between the fastest and slowest.

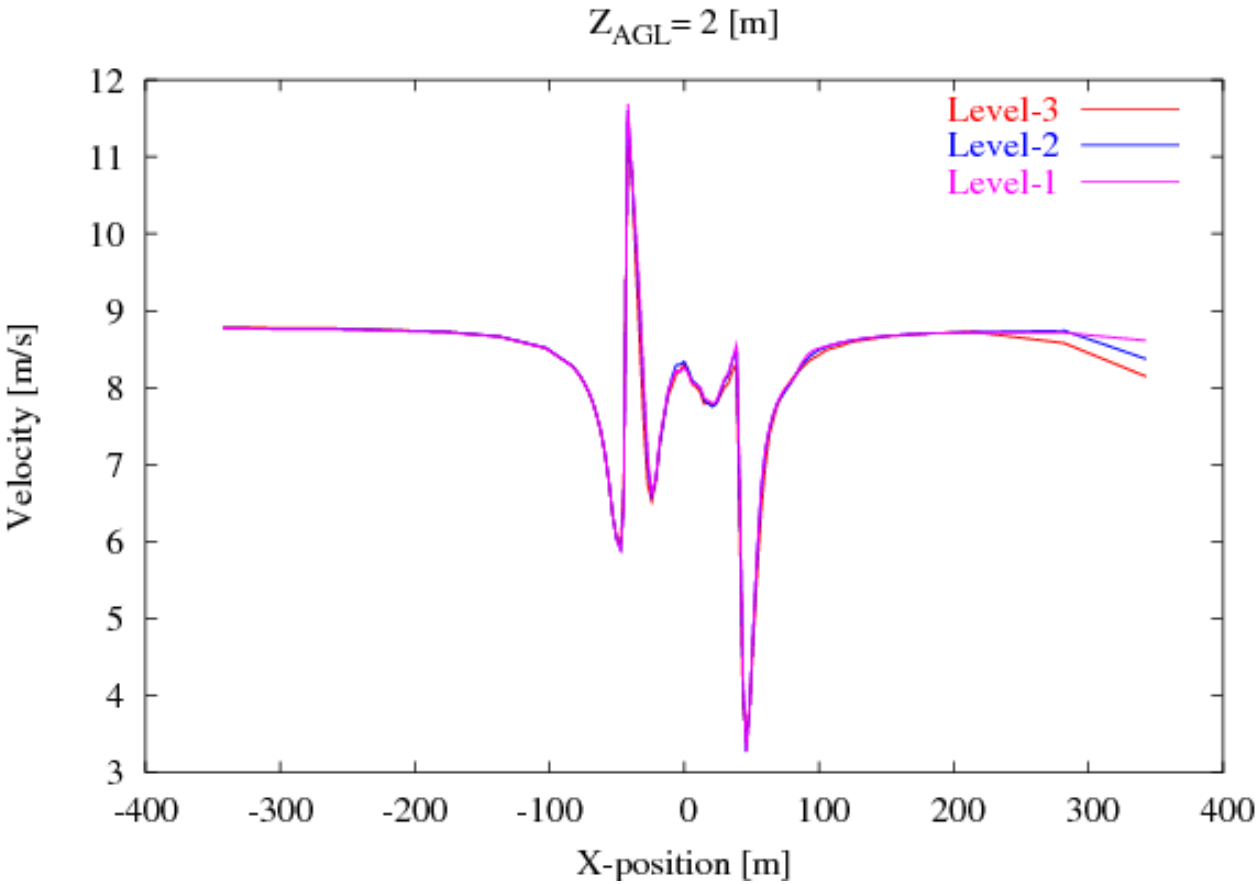
Including the parallel runs the difference is more than 5000

Setup of the masts



Grid Convergence (1)

- Flow direction 270 degrees, computations along line B.

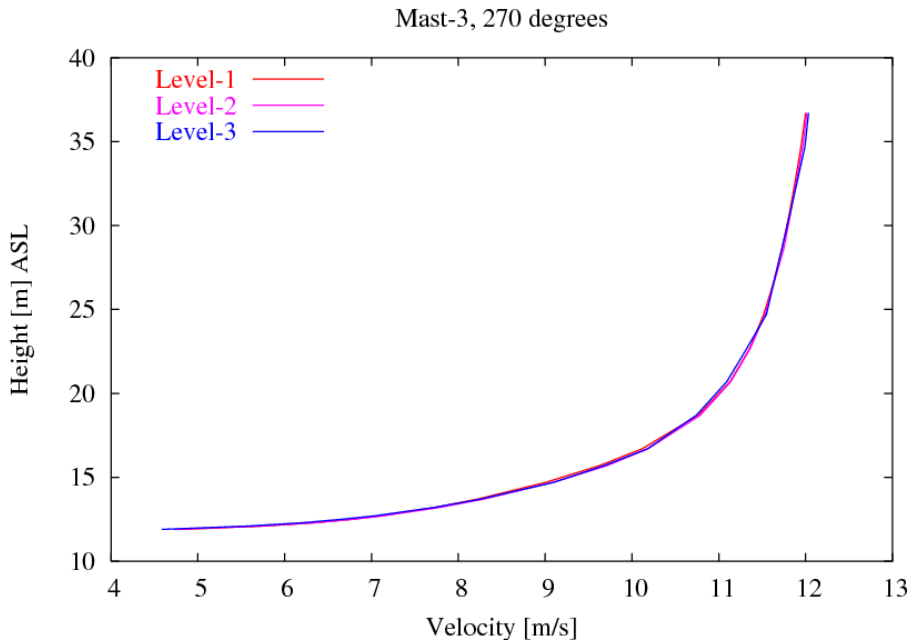


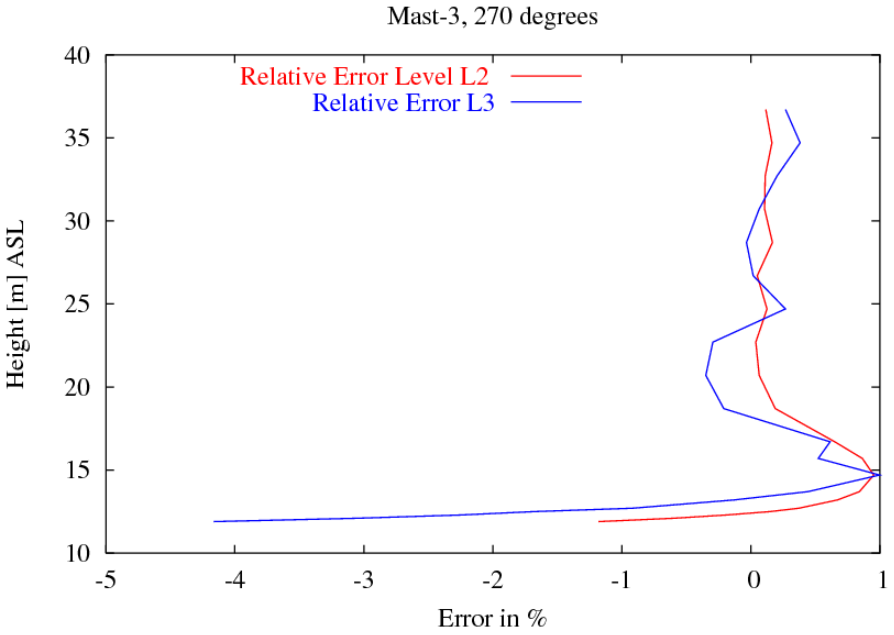
Level-3 = 0.21 Mill

Level-2 = 1.7 Mill.

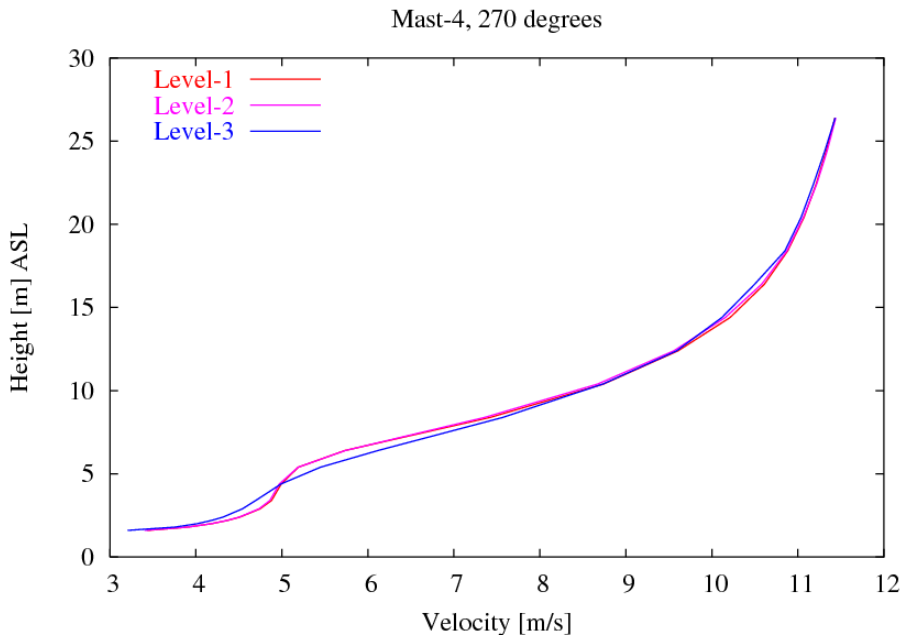
Level-1 = 13.6 Mill.

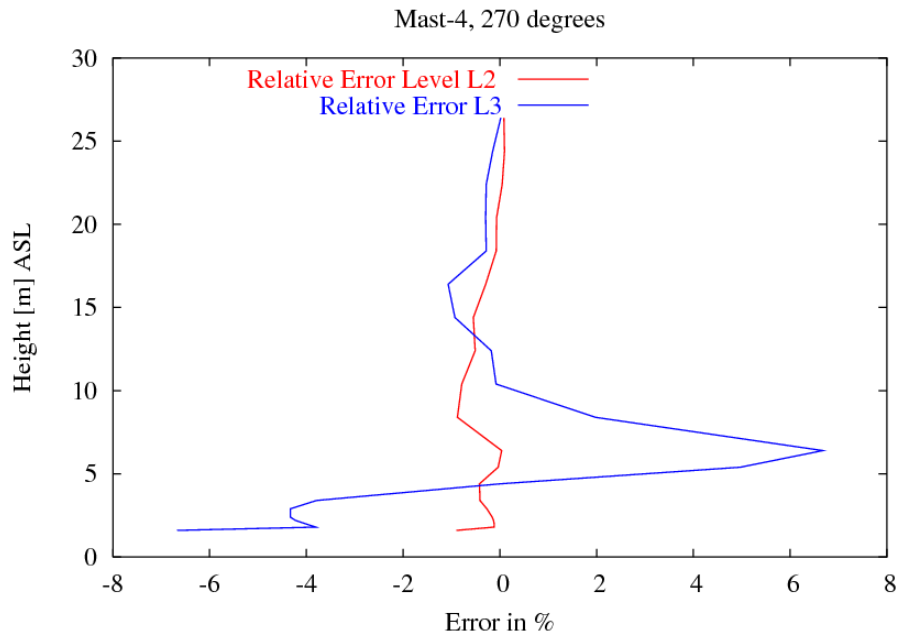
Grid Convergence (2)



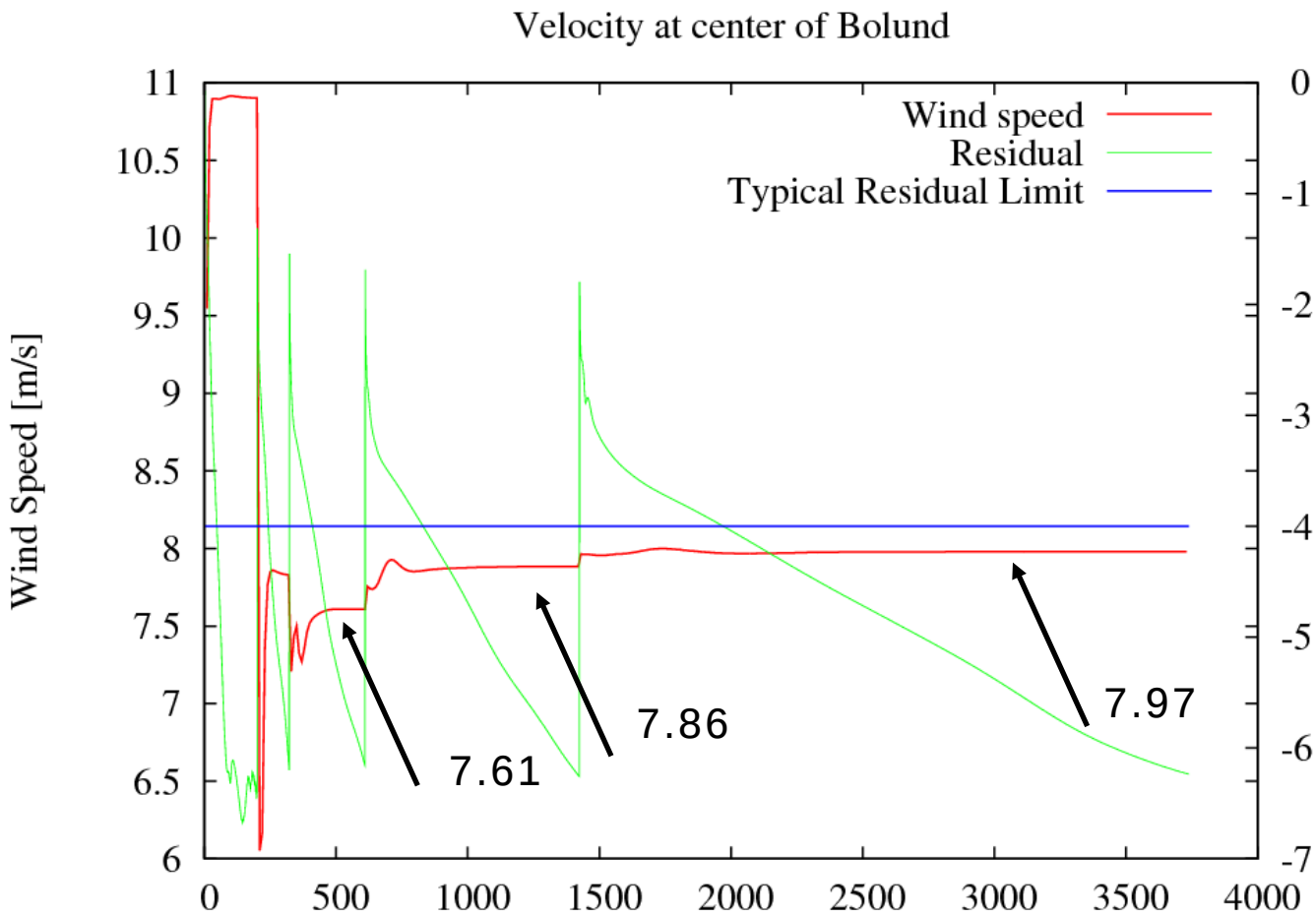


Grid Convergence (3)

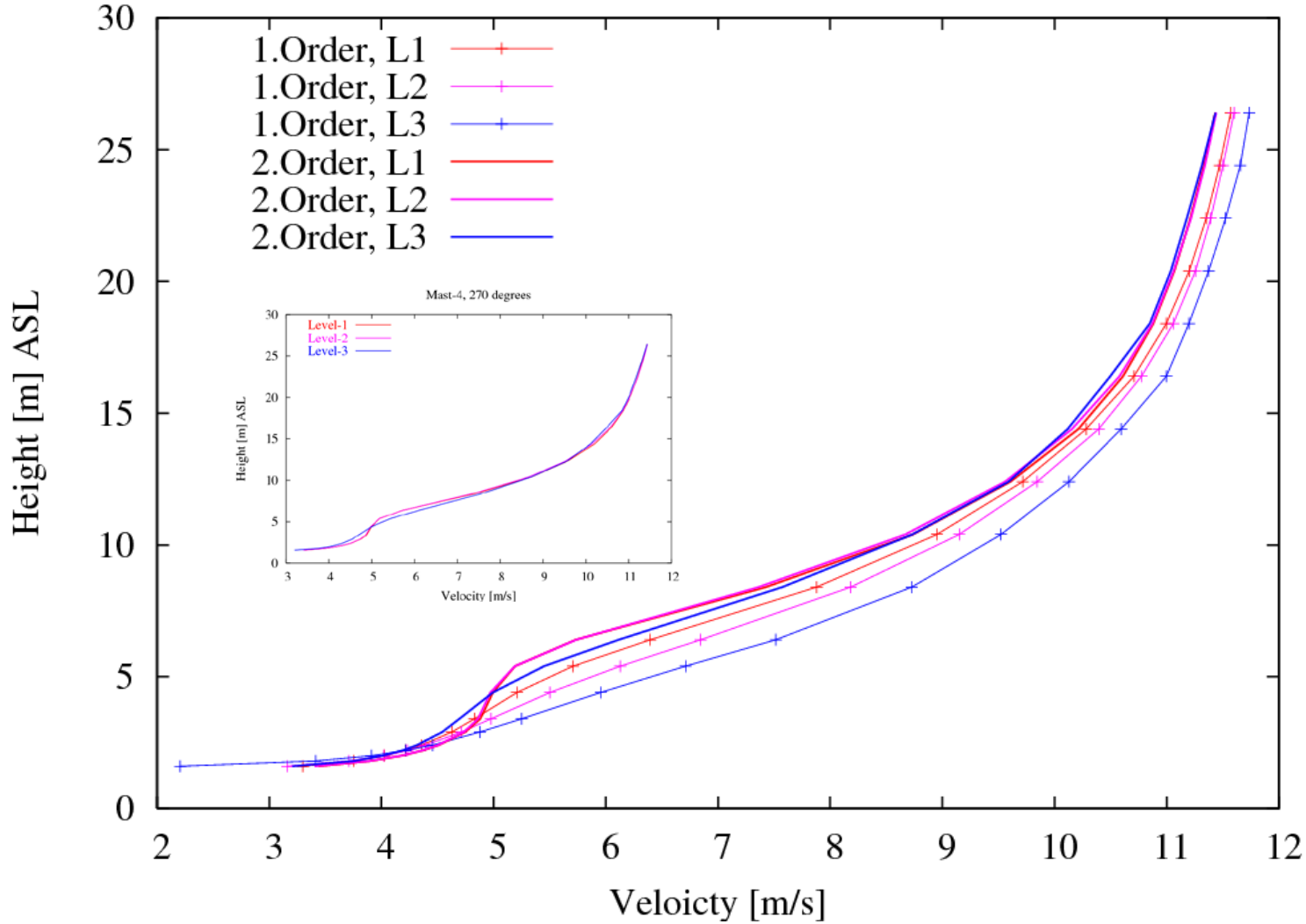




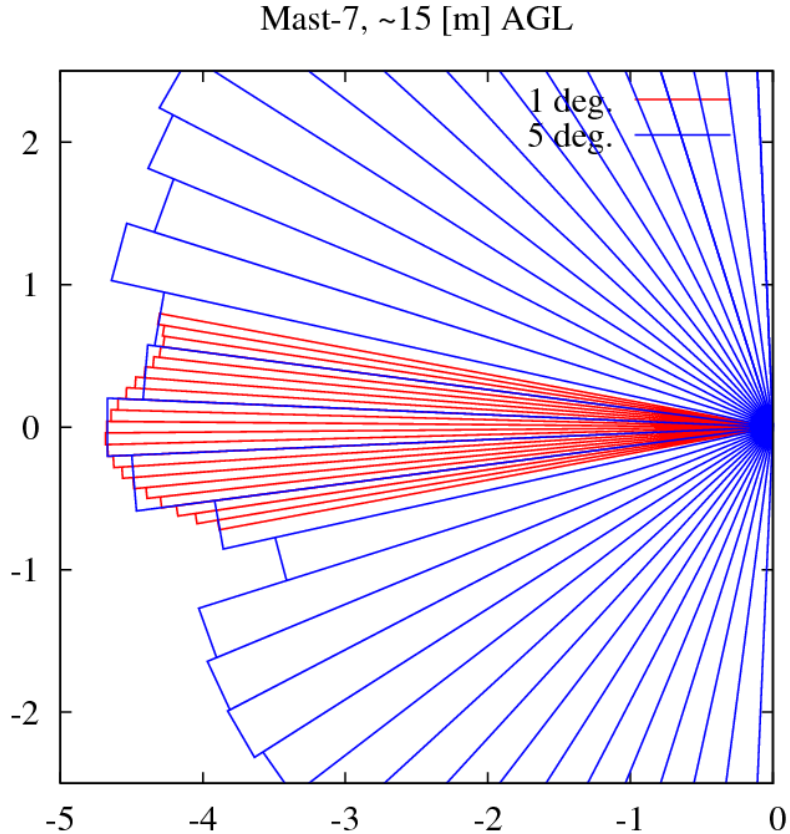
Convergence of the equations



Order of accuracy



Non-linearities due to the terrain and wind direction



Computing a single flow direction using CFD we need to consider:

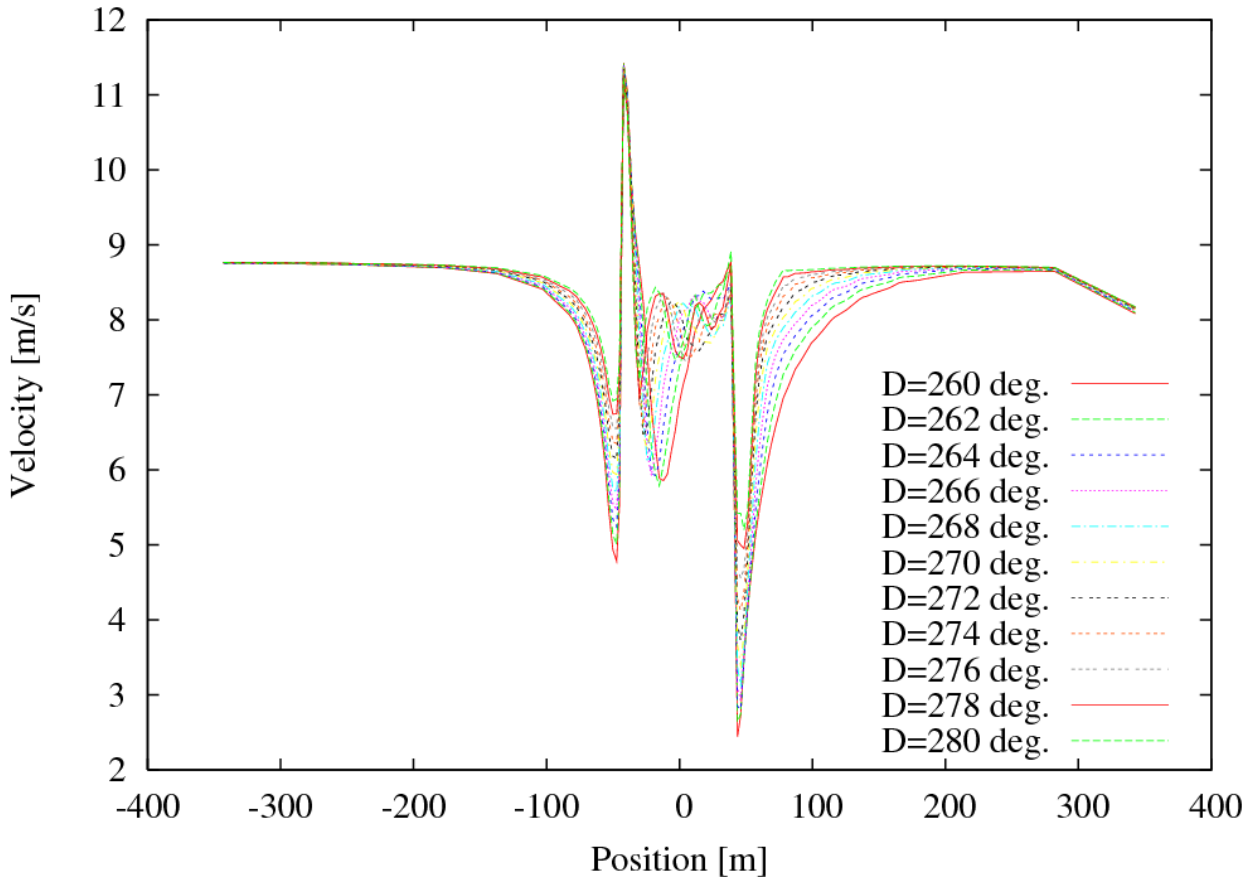
- Non-linear directional effects
- The frequency of the different directions

In the present location and for an inflow variation of +/- 13 degrees the variation is up to ~18%

In other places the variation can be even larger

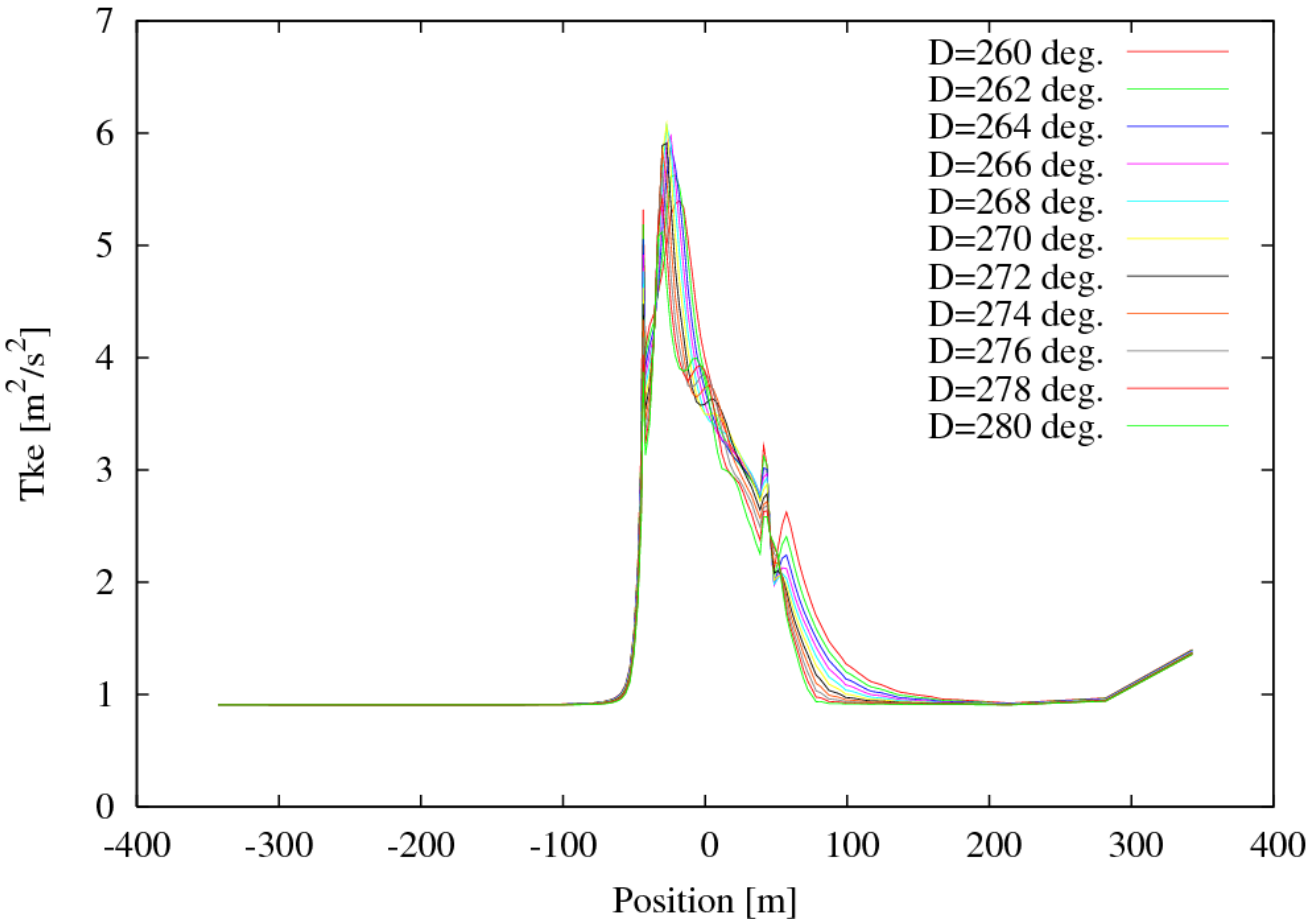
Variation of the velocity with flow angle

- Dir = 270 [deg], Height AGL = 2 [m]

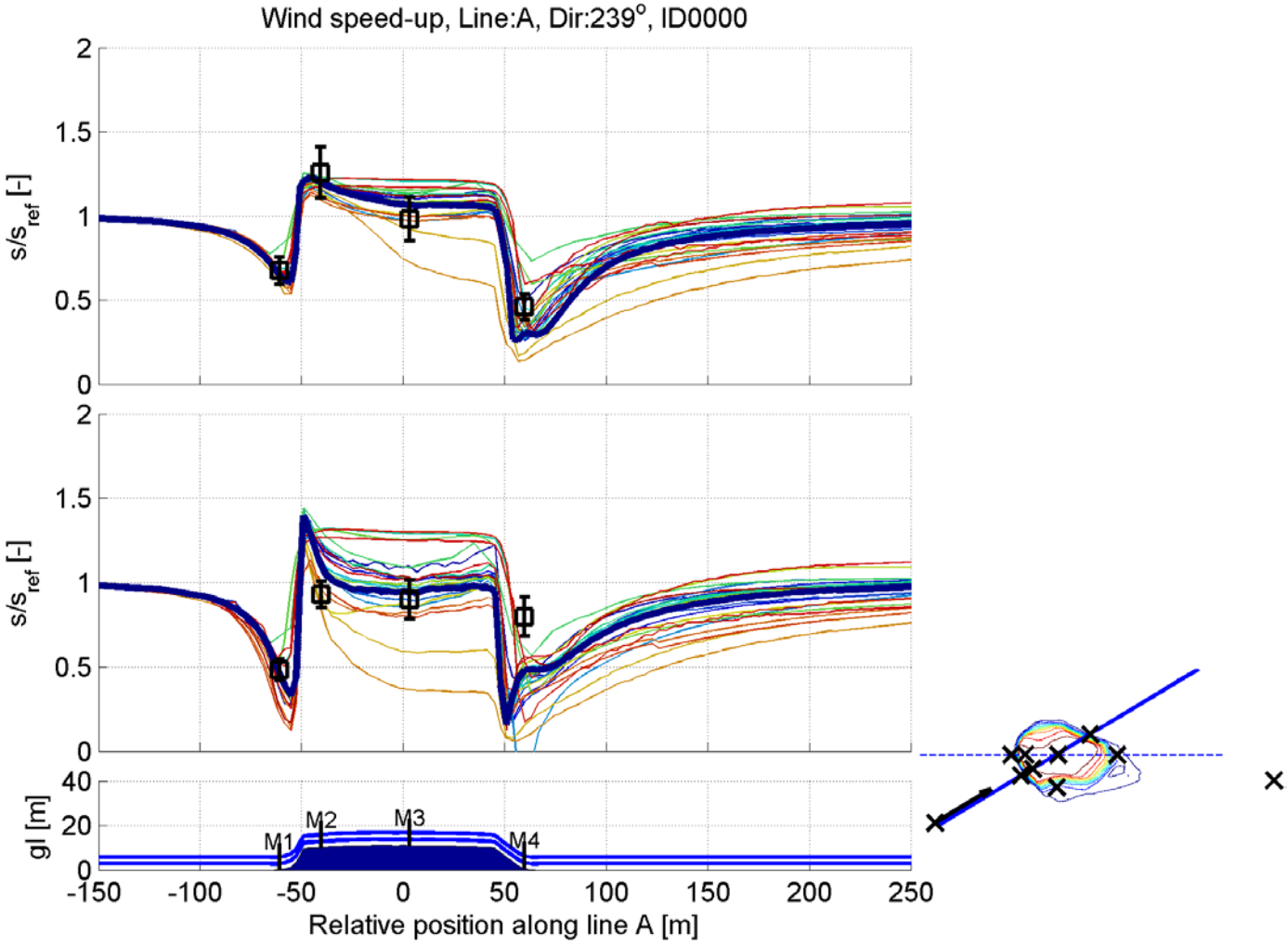


Variation of turbulence with flow angle

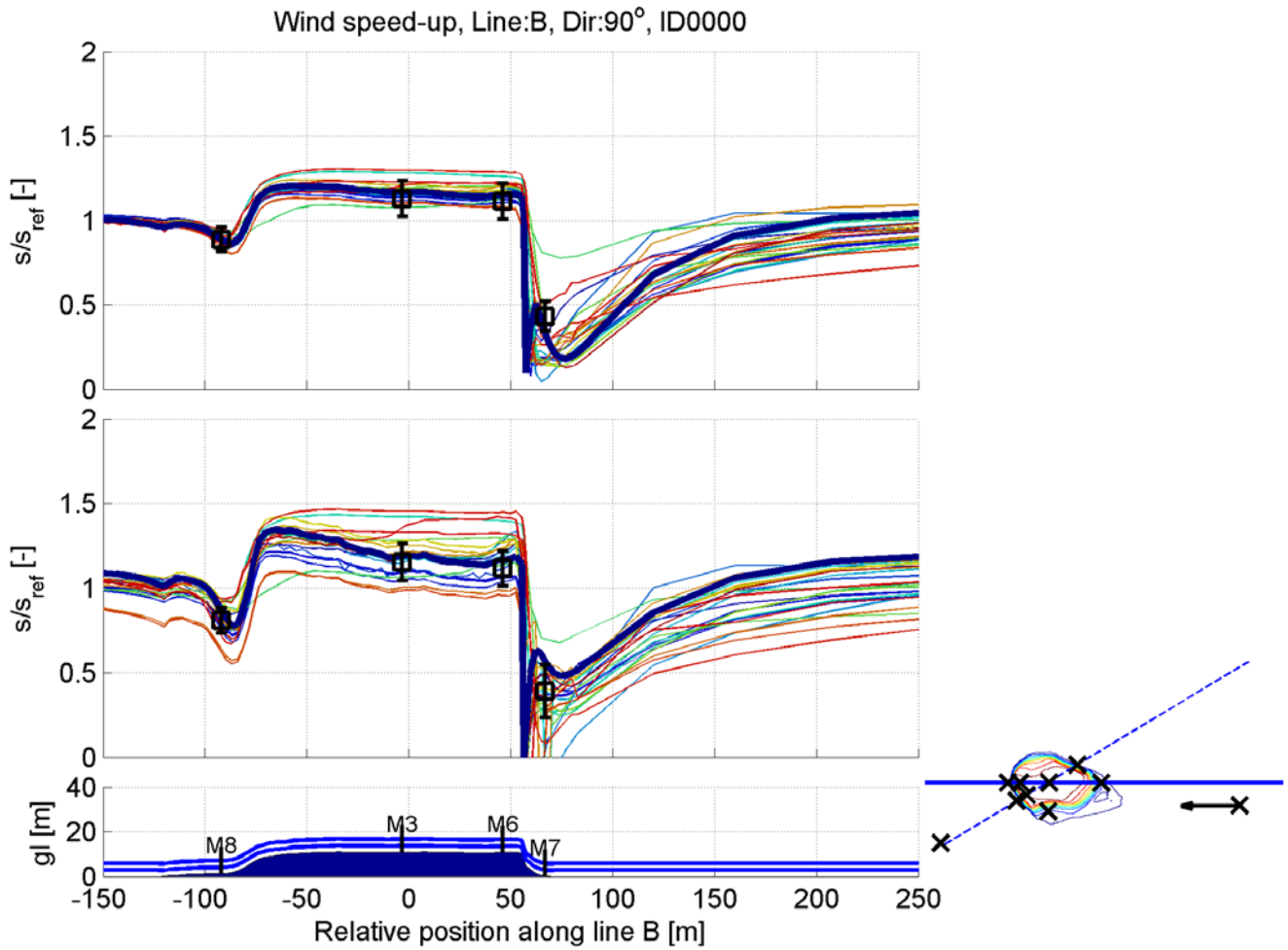
- Dir = 270 [deg], Height AGL = 2 [m]



Comparison with measurements



More detailed measurements would be interesting



Conclusion

- The Bolund Blind Comparison shows that good agreement between the majority of involved RANS type models.
- Yesterday we saw that they were also able to predict the measurements with ~ 15% error.
- The typical number of points ranges from 0.5 to 4 million.
- Typical compute times between 0.01 to 0.1 sec/point.
- Grid refinement studies indicates that already with 0.21 million points a good solution can be obtained (compute time ~< 10min on one CPU)
- With these low computing times the full wind rose with 5 to 10 degrees resolution can easily be computed.

Bolund may not be typical for the majority of sites, due to the well defined inflow boundary conditions. The lack of well defined inflow BC's may severely change the conclusion of good agreement.

Hopefully further large scale experiments aimed directly at code validation will take place in the future.