CFD for Rotor Aerodynamics

Sørensen, Niels N.

Publication date:
2010

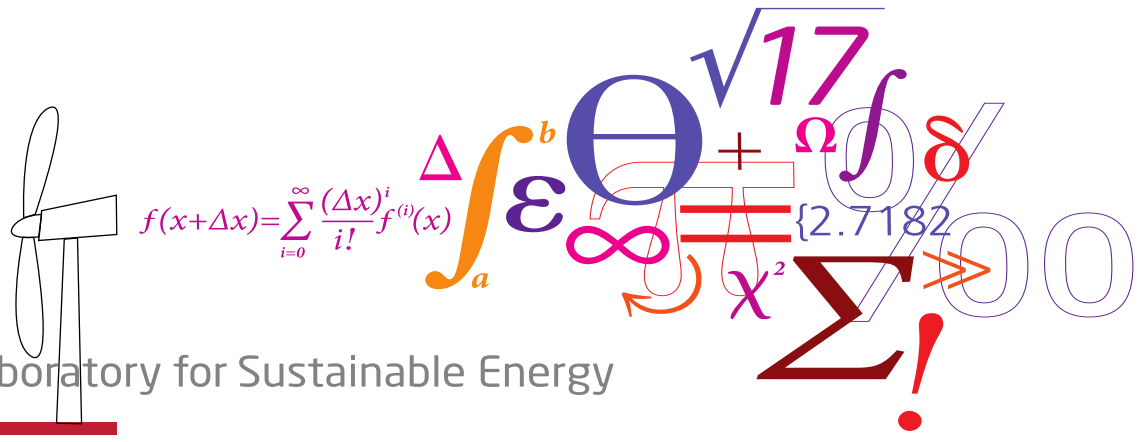
CFD for Rotor Aerodynamics

Glasgow - 2010

Niels N. Sørensen

Wind Energy Division · Risø DTU

RISØ-DTU, 09-09-2010



Risø DTU
National Laboratory for Sustainable Energy

Outline

1. Introduction
2. Mathematical Model
3. Discretization
4. Boundary Conditions
5. Computational domain
6. Solution Evaluation and Test Cases

CFD for Wind Energy

Development and origin of CFD for Wind Energy

- The application of numerical methods to fixed wing and rotor aerodynamics dates back to the late seventies in the aerospace community, solving steady Potential and Euler Equations.
- The first numerical solutions of the unsteady Euler equation were seen through the eighties.
- With the continuous increase in computer power in the late eighties and early nineties the first Reynolds Averaged Navier-Stokes codes for helicopter applications appeared.
- With the possibility of handling viscous flow, the first association of Navier-Stokes CFD solvers appeared in the wind turbine community in the late nineties.
- In Europe a series of EU-financed projects were providing the basis for many of these activities.

Wind Turbine Aerodynamics

- High Reynolds Number $Re > 1.0 \times 10^6$
- Nearly incompressible flows $M < 0.3$
- Operating in the atmospheric shear layer
  - High inflow turbulence
  - Velocity varies with height
  - Inflow depends on terrain
  - Large range of scales
- Thick airfoils
- Flow separation at high angles of attack
- Rotational effects
- Rotor tower interaction
- Aerodynamic and structural coupling, Aeroelasticity

Niels N. Sørensen,
Risø DTU

Basic CFD

- Mathematical Model
  - Compressible/Incompressible.
  - Navier-Stokes, Euler, Potential.
  - Coordinate and basis vector systems
- Discretization
  - Finite Difference, Finite Volume, Finite Element, Other Methods
  - Order of the discretization scheme
  - Computational domain, using structured/unstructured etc.
- Boundary Conditions
  - Inflow conditions
  - Wall boundary conditions
  - Outflow conditions
- Components of a CFD Solver
  - Pre-processor
  - Solver
  - Post-processor

Governing Equations

We would like to investigate airfoils and rotors

- We would like to be able to compute viscous effects:
  - Compute the viscous drag of airfoils and rotors.
  - Compute the effect of changing the Reynolds Number.
  - Compute airfoil at high angles of attack.

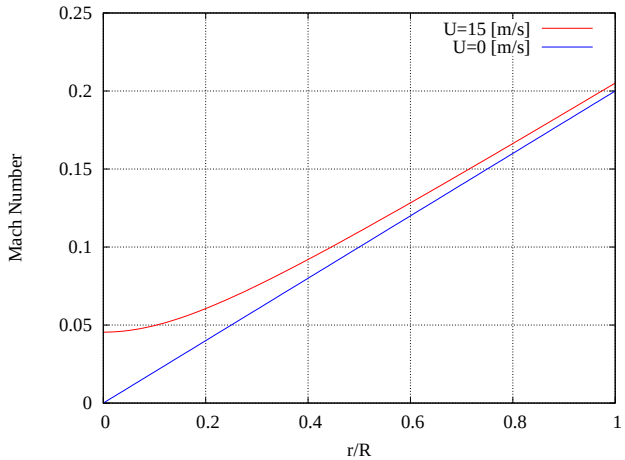
- The Potential Equation is inviscid and irrotational. Can be used as an approximation at low angles of attack and for thin airfoils.
- The Euler Equations are inviscid.
- Only the Navier-Stokes will allow us to compute the viscous effects.

Mach number variation on a typical Rotor

$$M = \frac{U_{eff}}{c} \; , c \sim 330[m/s] \qquad (1)$$

$$U_{eff} = \sqrt{U_\infty^2 + (r\omega)^2} \; , R\omega = 0.2c \qquad (2)$$

$$M = \sqrt{\left(\frac{U_\infty}{c}\right)^2 + \left(\frac{r}{R}0.2c\right)^2} \qquad (3)$$

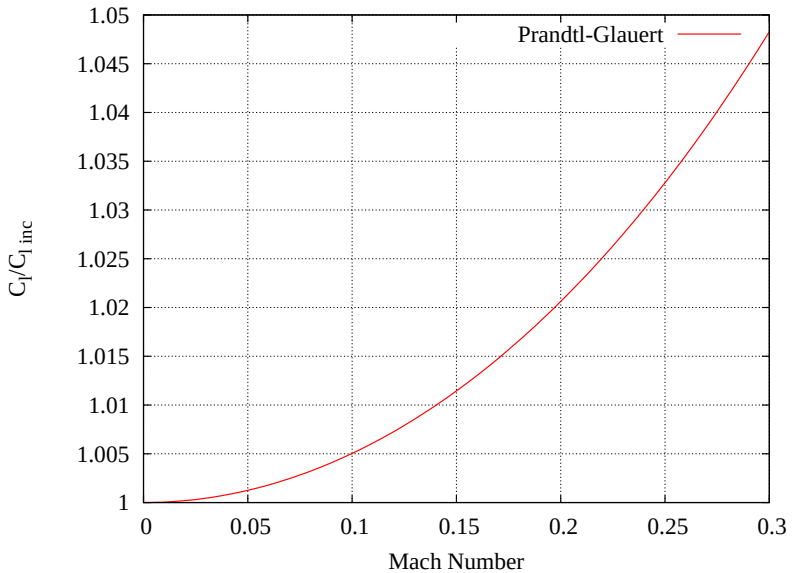


At the stagnation and in separated regions the Mach number may be even lower!

Mach number correction

Several low Mach number corrections exists, eg. Prandtl-Glauert:

$$C_l = \frac{C_{l,0}}{\sqrt{1 - M^2}} \tag{4}$$



The work of Turkel, Radespiel and Kroll shows that the error at M=0.01 may be more than 10 percent, and divergence may occur using a standard compressible solver.

Reynolds Averaged Navier-Stokes

Reynolds averaging of the Navier-Stokes equation, splitting the velocities in the mean and the fluctuating component

$$u_i(\vec{r},t) = U_i(\vec{r}) + u'(\vec{r},t) \text{ , where } U_i(\vec{r}) = \lim_{T\to\infty} \frac{1}{T} \int_t^{t+T} u_i(\vec{r},t)dt$$

Inserting the Reynolds decomposed velocity in the Navier-Stokes and continuity equations

Perform time averaging of the equations. The equations are in principle time independent, or steady state.

The Reynolds Averaged Navier-Stokes

The Reynolds Averaged incompressible Navier-Stokes equations and additional equations have the following form:

Continuity equation:

$$\frac{\partial}{\partial x_j}(\rho U_j) = 0$$

Momentum equations:

$$\frac{\partial}{\partial t}(\rho U_i) + \frac{\partial}{\partial x_j}(\rho U_i U_j) - \frac{\partial}{\partial x_j}\left[(\mu + \mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] + \frac{\partial \hat{P}}{\partial x_i} = S_v \ ,$$

Auxiliary equations:

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x_j}(\rho U_j \phi) - \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\phi}\right)\frac{\partial \phi}{\partial x_i}\right] = S_\phi$$

Closing the Equations

To close the flow equations we need an expression for the μ_t, this is typically handled by the turbulence model:

Typically a two equation model will be used for more complex cases, e.g.. the $k-\epsilon$ or the $k-\omega$ model

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \ .$$

The two additional transport equations has a form similar to the previous stated general transport equation, and mainly the deviation between the models are in the source terms on the RHS.
This is discussed more in the accompanying discussion of turbulence.

Coordinate and basis vector system

Coordinate System

- Cartesian Coordinates (x,y,z)
- Polar Coordinates (r,θ,z)
- Spherical Coordinates (r,θ,Φ)
- General Curvilinear Coordinates (ξ,η,ζ)
- Fixed Frame
- Moving Frame

Velocity basis

- Fixed Cartesian Velocity basis
- Fixed Polar Velocity basis
- Fixed Spherical Velocity basis
- Co- or Contra- Variant Velocity basis

Depending of the chosen coordinate basis and chosen velocity basis, the resulting version of the Navier-Stokes equations will have different complexity.

Rotor Flows

Non-inertial frame of reference (Moving Frame)

- Additional acceleration terms, changed boundary conditions
- Can be used both steady/unsteady
- Do not need re-computations of geometrical quantities
- Need careful treatment of source term implementation
- Do not allow deformation of the geometry

Inertial frame of reference (Moving Mesh)

- Mesh fluxes, changed boundary conditions
- Is by nature unsteady
- Allows deformation of the geometry
- Need re-computations of metrics (geometrical quantities)

Transformation to curvilinear Coordinates

If we need to handle curvilinear Coordinates, the equations must be transformed from the Cartesian formulation to the curvilinear formulation.

Typically, we derive an expression for the divergence using the chain rule of differentiation:

$$\frac{\partial}{\partial x_i} = \frac{1}{J}\left(\frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x_i} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x_i} + \frac{\partial}{\partial \zeta}\frac{\partial \zeta}{\partial x_i}\right) \tag{5}$$

Using fixed Cartesian base vectors, this approach will directly lead to a strong conservation form:

The volume sources and the time terms are invariant during transformation.

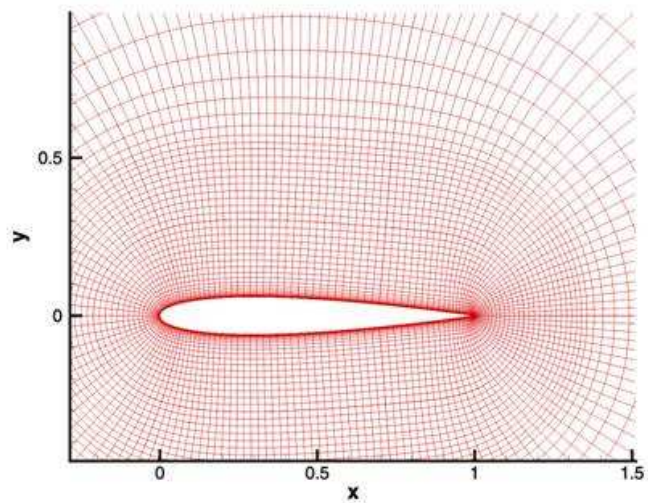
Discretization Methods

We need to transform the partial differential equations, into a set of algebraic equation that can be solved numerically.

- Finite Difference:
  - Differential from, using Taylor Series or Polynomial fitting.
  - Structured grids.
- Finite Volume:
  - Integral form, using Gauss or Divergence Theorem
  - Structured and unstructured grids
- Finite Element:
  - Integral form, shape or weight functions.
  - Unstructured grids.
- Other Methods:
  - Spectral, Smoothed Particle Dynamics, Boundary Element Methods.

Computational Grids

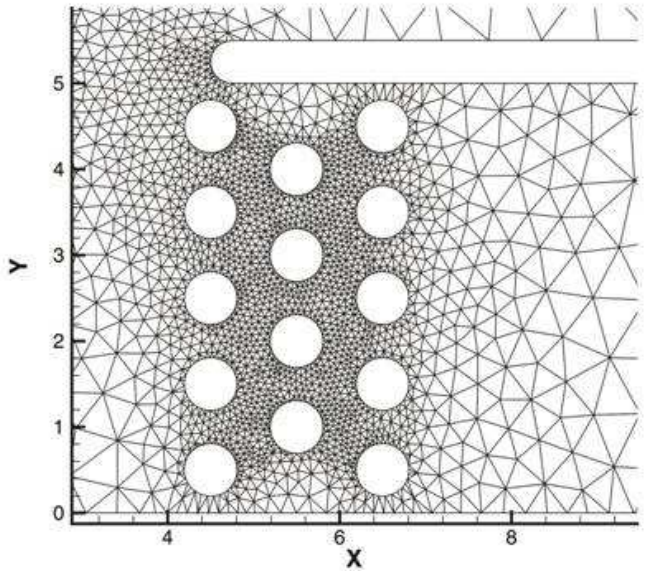
- Structured Grid



Niels N. Sørensen,
Risø DTU

Computational Grids

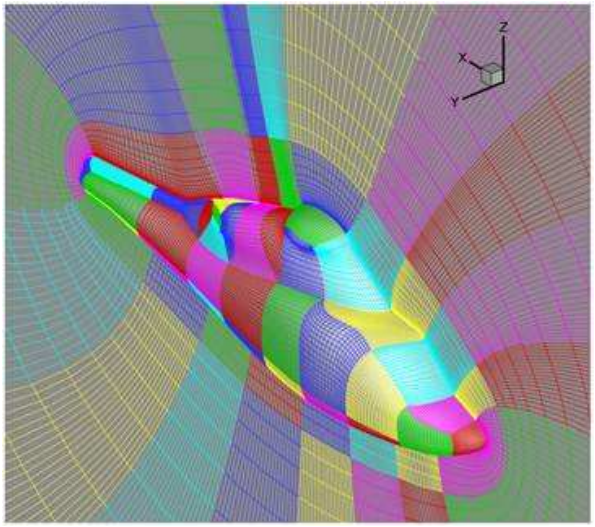
- Structured Grid
- Unstructured Grid



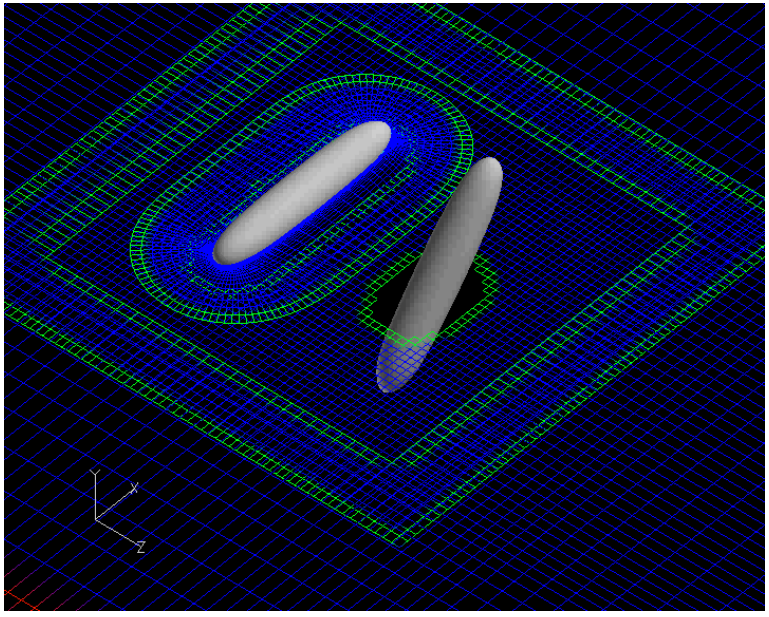
Niels N. Sørensen,
Risø DTU

Discretization

Computational Grids



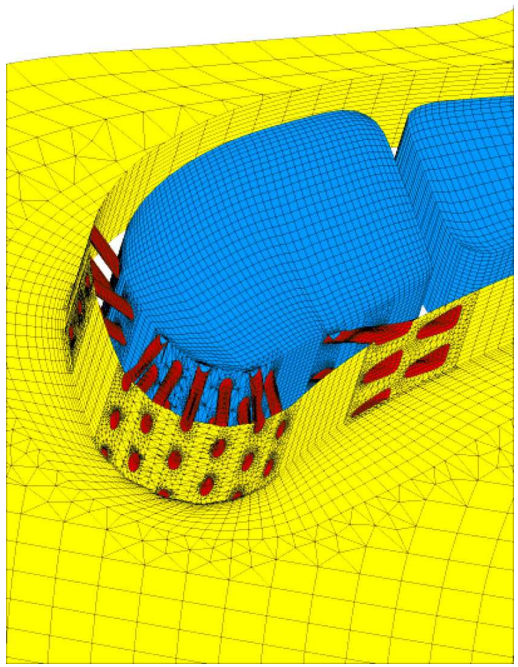
- Structured Grid
- Unstructured Grid
- Multi-Block
  - Conforming Grids
  - Overlapping or Chimera



Niels N. Sørensen,
Risø DTU

Computational Grids

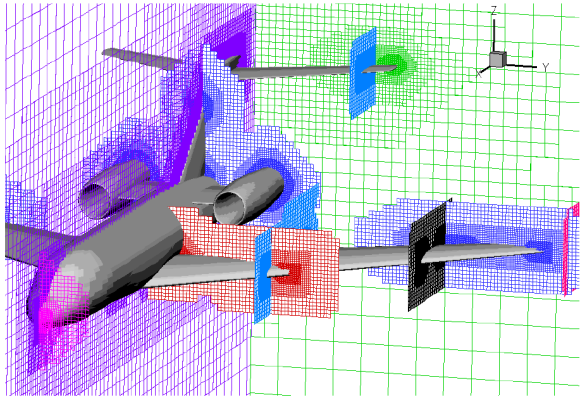
- Structured Grid
- Unstructured Grid
- Multi-Block
  - Conforming Grids
  - Overlapping or Chimera
- Hybrid Meshes



Niels N. Sørensen,
Risø DTU

Computational Grids

- Structured Grid
- Unstructured Grid
- Multi-Block
  - Conforming Grids
  - Overlapping or Chimera
- Hybrid Meshes
- Cartesian Cut Cells

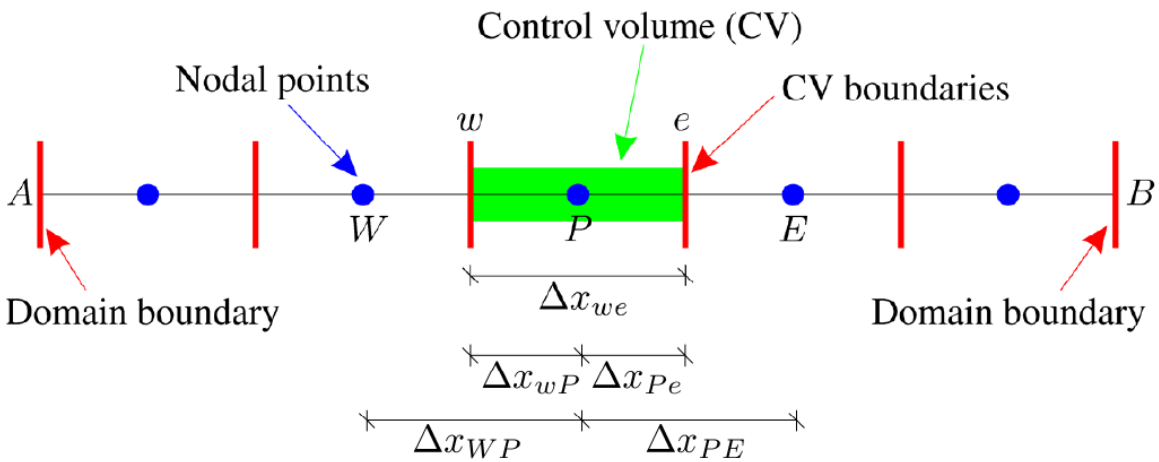


Niels N. Sørensen,
Risø DTU

Finite Volume Discretization

In the Finite Volume method, the algebraic equations are obtained by partitioning the solution domain into a set of finite control volumes and integrating the equation using the Gauss' or divergence theorem.
The Gauss theorem states that the volume integral of the divergence is equal to the outward flux of the vector field through the boundary enclosing the volume:

$$\int_{CV} \frac{\partial \phi u_i}{\partial x_i} dV = \int_A n_i (\phi u_i) dA \qquad (6)$$



Niels N. Sørensen,
Risø DTU

Discretization of the equations

The Reynolds Averaged Navier-Stokes equations are given by:

$$\frac{\partial}{\partial t}(\rho U_i) + \frac{\partial}{\partial x_j}(\rho U_i U_j) - \frac{\partial}{\partial x_j}\left[(\mu + \mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] + \frac{\partial \hat{P}}{\partial x_i} = S_v \ , \qquad (7)$$

Even for the curvilinear we can arrive at a algebraic set of equations of the form below:

$$A_p U_p + \sum A_{nb} U_{nb} = S \qquad (8)$$

Niels N. Sørensen,
Risø DTU

Source Terms

Generally, the source terms are treated in a non-conservative way:

$$\int_{CV} S_v dVol = S_v Vol \tag{9}$$

The non-conservative treatment may result in problems eg. in connection with moving-frame formulations.

Unsteady Terms

The time derivative can be approximated by a implicit three point backward scheme, which is formally of second order:

$$\frac{\partial \phi}{\partial t} = \frac{3\phi^t - 4\phi^{t-1} + \phi^{t-2}}{2\Delta t} .$$

Typically, the implicit discritization will be coupled with a dual time-stepping or sub-iteration approach.

$$\int_{CV} \frac{\partial \rho U_j}{\partial t} dV = \int_{CV} \rho \left(\frac{3\phi^t - 4\phi^{t-1} + \phi^{t-2}}{2\Delta t} \right) dVol = \rho \frac{3U^t - 4U^{t-1} + 2U^{t-2}}{2\Delta t} Vol$$

Pressure term

$$\frac{\partial P}{\partial x_i}$$

The pressure term for the x-momentum equation can be approximated in the following way

$$\int_{CV} \frac{\partial P}{\partial x} dVol = P_e A_e - P_w A_w \tag{10}$$

In a cell centered formulation, the P_e and P_w are not directly available, using linear interpolation we get

$$P_e = \frac{P_E + P_P}{2} \; , P_w = \frac{P_P + P_W}{2}$$

For a Cartesian mesh where $A_e = A_w$ the P_P will exit the equation:

$$\int_{CV} \frac{\partial P}{\partial x} dVol = A_e \frac{P_E - P_W}{2}$$

Diffusive terms

$$\int_{CV} \frac{\partial}{\partial x_j}\left[(\mu+\mu_t)\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)\right] dVol \tag{11}$$

Looking at the x-momentum equation:

$$\int_{CV} \frac{\partial}{\partial x}\left[2(\mu+\mu_t)\frac{\partial U}{\partial x}\right] + \frac{\partial}{\partial y}\left[(\mu+\mu_t)\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[(\mu+\mu_t)\left(\frac{\partial U}{\partial z}+\frac{\partial W}{\partial x}\right)\right] dol$$

Again identifying a divergence of a vector field, using the divergence theorem:

$$\int_{CV} \nabla \vec{F} dVol = \int_A \vec{n}\vec{F} dA \tag{12}$$

Niels N. Sørensen,
Risø DTU

Diffusive terms 2

Assuming a Cartesian grid we get, using eg. $\vec{n_e} = (1,0,0)^T$ and $\vec{n_s} = (0,-1,0)^T$

$$
\begin{aligned}
&2\mu_{eff}\left.\frac{\partial U}{\partial x}\right|_{east} A_e - 2\mu_{eff}\left.\frac{\partial U}{\partial x}\right|_{west} A_w \\
&+\mu_{eff}\left.\left(\frac{\partial V}{\partial x}+\frac{\partial U}{\partial y}\right)\right|_{north} A_n - \mu_{eff}\left.\left(\frac{\partial V}{\partial x}+\frac{\partial U}{\partial y}\right)\right|_{south} A_s \\
&+\mu_{eff}\left.\left(\frac{\partial W}{\partial x}+\frac{\partial U}{\partial z}\right)\right|_{top} A_t - \mu_{eff}\left.\left(\frac{\partial W}{\partial x}+\frac{\partial U}{\partial z}\right)\right|_{bottom} A_b
\end{aligned}
$$

The effective viscosity $\mu + \mu_t$ can be linearly interpolated between the two opposing cell centres

The gradients can be approximated by central differences, which will have 2. order accuracy:

$$\left.\frac{\partial U}{\partial x}\right|_e = \frac{U_E - U_P}{\Delta x_{PE}}$$

Convective terms

$$\frac{\partial}{\partial x_j}(\rho U_i U_j)$$

Looking at the X-momentum direction:

$$\int_{CV} \frac{\partial \rho U U_j}{\partial x_j} dVol = \int_{CV} \nabla \cdot (\rho U \vec{V}) dVol = \int_A \vec{n} \rho U \vec{V} dA$$

We can now derive the discrete version:

$$\rho UU|_{east} A_e - \rho UU|_{west} A_w + \rho UV|_{north} A_n - \rho UV|_{south} A_s + \rho UW|_{top} A_t - \rho UW|_{bottom} A_b$$

For a cell centered scheme we will need approximations for the cell face velocities. One simple option with second order accuracy for the convective terms could be a linear interpolation:

$$U_e = \frac{U_E + U_E}{2}$$

This scheme is unfortunately only stable for very low Peclet numbers $\frac{LV}{\mu}$

Convective scheme

A very accurate and often used scheme for approximating the cell face variables is the QUICK scheme:

$$\phi_e = \begin{cases} \frac{1}{2}\left(\phi_P + \phi_E\right) - \frac{1}{8}\left(\phi_W + \phi_E - 2\phi_P\right) & \text{if } \rho U_e A_e \geq 0 \\ \frac{1}{2}\left(\phi_E + \phi_P\right) - \frac{1}{8}\left(\phi_P + \phi_{EE} - 2\phi_E\right) & \text{if } \rho U_e A_e \leq 0 \end{cases} .$$

Often the higher order schemes are written as deffered corrections in the following way:

$$\phi_e = \begin{cases} \phi_P + \frac{1}{8}\left(3\phi_E + 2\phi_P - \phi_W\right) & \text{if } \rho U_e A_e \geq 0 \\ \phi_E + \frac{1}{8}\left(3\phi_P - 2\phi_E - \phi_{EE}\right) & \text{if } \rho U_e A_e \leq 0 \end{cases} .$$

Only the first part of the scheme is treated fully implicit while the remaining part is evaluated at last iteration or sub-iteration.

Niels N. Sørensen,
Risø DTU

Solution Methods for Incompressible Flow

For the Finite Volume and Finite Difference methods, the typical solution methods are listed below:

- Artificial Compressibility Methods
  - Explicit Methods
  - Implicit Methods
- Fractional Step Methods
  - Explicit Methods
  - Implicit Methods
- Pressure Correction Methods
  - SIMPLE
  - PISO
  - SIMPLEC

Requirements of a Numerical Solution Method

For the discrete equations to be solved the following properties should be fulfilled

- Consistency and convergence
  - The difference between the discretized and the exact equations should become zero in the limit of infinitely small cells.
- Stability
  - A numerical procedure is said to be stable if it does not magnify the errors that appear in the course of the numerical solution process.
- Conservation
  - The numerical method should reflect the conservation property of the governing equation.
- Boundedness and Realizability
  - Physically non-negative quantities (density, concentration etc) must always be positive. Some convective schemes may produce nonphysical negative values on coarse and skewed computational meshes.

Boundary Conditions

The two fundamental things determining the quality of the results from a numerical model, assuming that every thing is performed correctly are:

- The model equations
  - Are the model equations adequate for the present purpose etc.
- The boundary conditions
  - Boundary conditions are needed for all variables at all external boundaries of the computational domain.
  - Boundary conditions needs to represent the problem in question

In the following slides we will look at some typical boundary conditions.

Inflow Boundary Conditions

Two approaches can be used:

- Use a large domain, placing the farfield boundary at large distance from the airfoil/rotor.
  - Simple Dirichlet Condition
- Use a dynamic boundary conditions that adapts to the flow disturbance from the airfoil/rotor.
  - Advanced Dirichlet Condition that adapts to the loading.

Outflow conditions

The outflow conditions can be crucial for the computations:

- Simple fully developed flow assumptions are often used.
  - The outlet should be placed far from the area of interest
  - There should not be recirculation through the outlet

$$\frac{\partial \phi}{\partial n} = 0 \, .$$

The pressure will typically be extrapolated using either linearly or quadratic extrapolation.

- Convective boundary conditions, will allow reversed flow through the outlet.

$$\frac{\partial \phi}{\partial t} - U \frac{\partial \phi}{\partial n} = 0 \, .$$

Wall Boundary Conditions

For most airfoil and rotor flows, simple no-slip boundary conditions are applied.

This will typically require a very fine grid, with off cell spacing of $y^+ \sim 1$.

In contrast to many typically engineering flows, where the $y^+ \sim 100$ to 200 are use along with log-law conditions.
In connection with the discussion of the computational domain, we will see the implication of the y^+ requirement on the cell size.

Computational Domain

Airfoil and rotor flows involves a large range of scales, from the viscous scales in the boundary layer of the blades, to the wake of the rotor which can be several rotor diameters long.

- For most wind turbine related flows, we are looking at external flows where the outer boundary must be placed far from the area of interest.
- We need a sufficient fine resolution at the wall boundaries, and at all regions with large gradients of the flow variables (separation points, where layers, wake, etc)
- We only have a limited number of grid points available. A typical 2D airfoil computation would have around $384 \times 128 \sim 50.000$ cells and a typical rotor computation would have around number of blades $\times 128 \times 256 \times 128 \sim 13$ million cells.
- In order to avoid excessive number of cells we will need stretching functions, to expand the cell size when moving away from the wall surface.

Estimation of Viscous Cell Size

Definition of non-dimensional wall distance:

$$y^+ = \frac{\rho y U_\tau}{\mu} \tag{13}$$

Skin friction coefficient for turbulent boundary layer over flat plate:

$$C_f = \frac{0.0576}{Re^{\frac{1}{5}}} \tag{14}$$

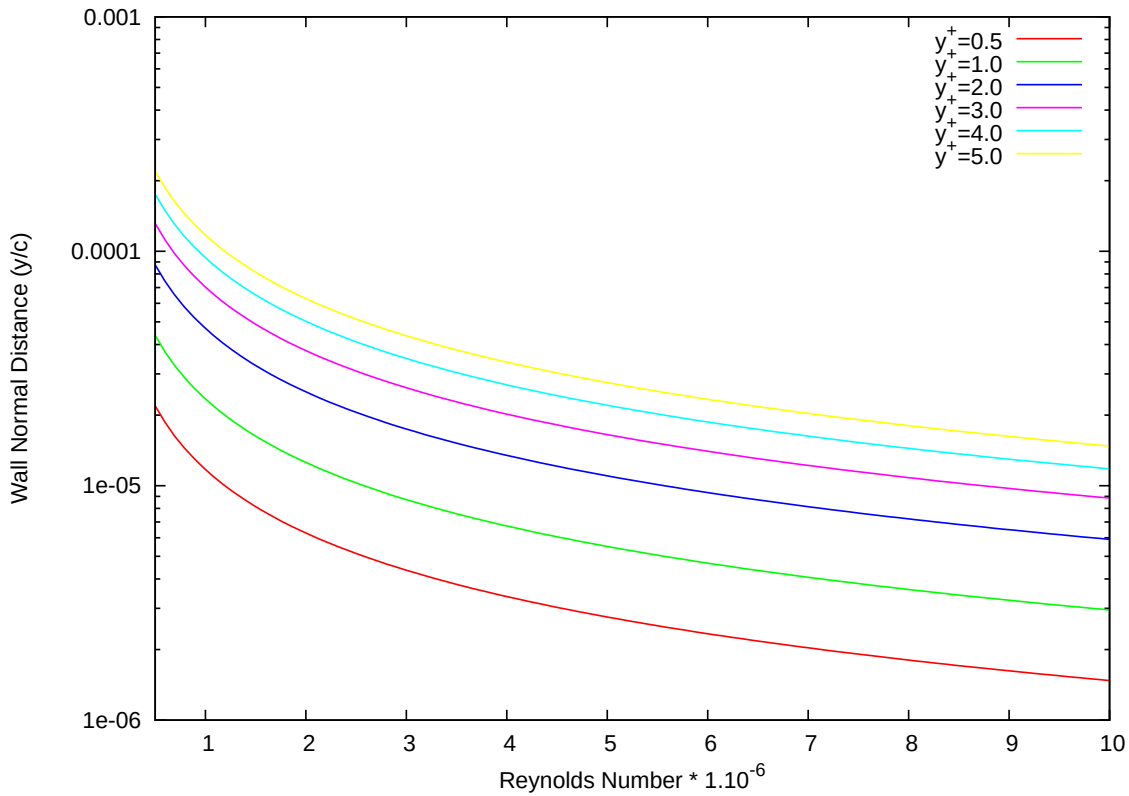
Definition of skin friction coefficient:

$$C_f = \frac{\rho U_\tau^2}{\frac{1}{2}\rho U_\infty^2} \tag{15}$$

Estimation of Viscous Cell Size

An approximate formula for the necessary wall normal distance can now be derived:

$$\frac{y}{c} = 5.89 Re^{-\frac{9}{10}} y^{+} \qquad (16)$$



Niels N. Sørensen,
Risø DTU

Domain size for 2D airfoil computations

The influence from a lifting airfoil on its surroundings can be determined from simple potential flow using the Kutta-Jukowski theorem:

$$L = \rho_\infty \vec{U_\infty} \vec{\Gamma} \text{ , where } L \text{ is the Lift per unit span,} \tag{17}$$

the definition of the lift:

$$L = C_l \frac{1}{2} \rho_\infty U_\infty^2 Chord \ , \tag{18}$$

and the Biot-Savart law:

$$V_\theta = \frac{\Gamma}{2\pi r} \tag{19}$$

Combining these we can arrive at the following expression for the:

$$\frac{U_\theta}{U_\infty} = -\frac{-C_l}{4\pi \frac{r}{Chord}} \tag{20}$$

Domain size for 2D airfoil computations

The effect of the induced velocity on the effective Reynolds number can thus be approximated by:

$$U_{eff} = U_{\infty}\sqrt{1+\left(\frac{C_l}{4\pi\frac{r}{Chord}}\right)^2} \tag{21}$$

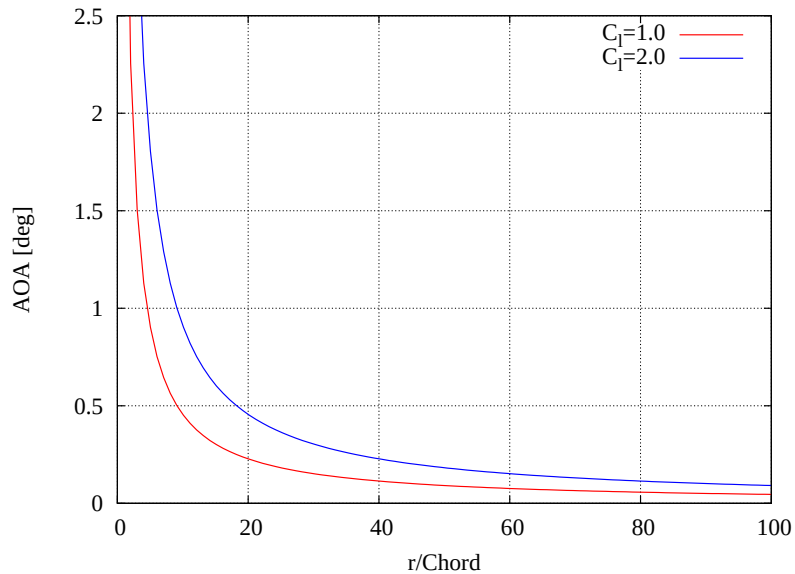
The effect of the induced velocity on the effective angle of attack can thus be approximated by:

$$AOA = \frac{180}{\pi} tan^{-1}\left(\frac{-C_l}{4\pi\frac{r}{Chord}}\right) \tag{22}$$

Niels N. Sørensen,
Risø DTU

Domain size for 2D airfoil computations

The effect on the Reynolds number is negligible for all practical purposes, but as can be seen from the present figure, the effect on the AOA is quite large for small domains:



Neglecting the induced tangential velocity, may lead to an error of ~ 5 percent for a distance from the airfoil of 10 chords, while the error is reduced to below 1 percent for a 50 chords distance.
Alternatively, the empirical expression for U_θ can be used as a iterative boundary condition, the so called vortex correction.

Domain size for rotor computations

In rotor computations we need to consider the influence of the rotor induced velocity

The induced velocity can be estimated based on the rotor thrust, using the Froude actuator disc model.

$$C_T = \frac{T}{\frac{1}{2}\rho U_\infty^2 \pi R_{Rotor}^2}$$

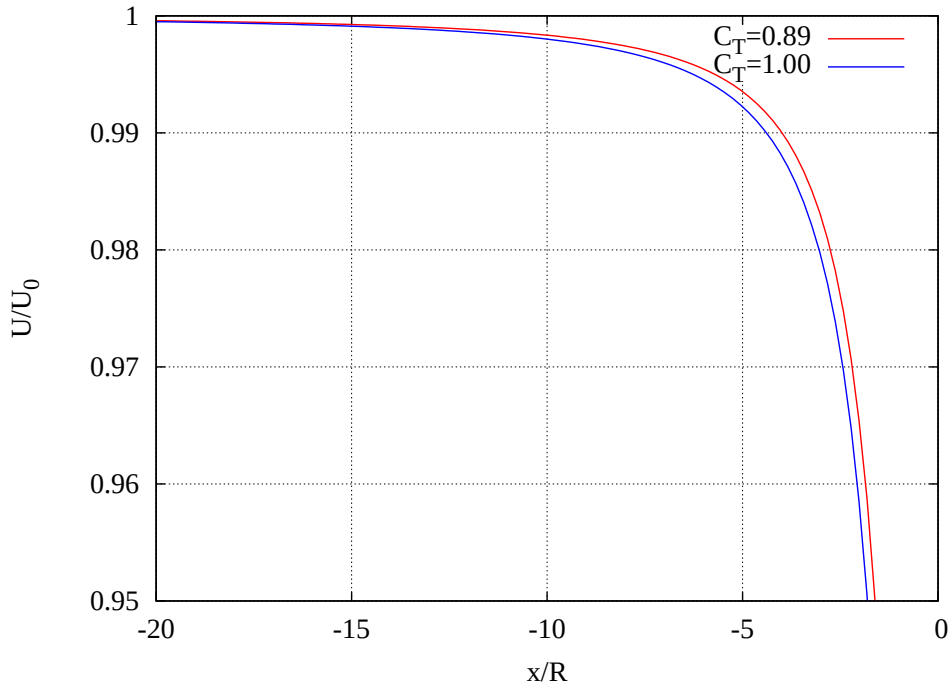
From axial momentum theory we can compute the induction:

$$C_T = 4a(1-fa)\ , f = \begin{cases} 1 & \text{for } a < 1/3 \\ (5-3a)/4 & \text{for } a \geq 1/3 \end{cases} \qquad (23)$$

$$U = U_\infty \left(1 - a\left(1 + \frac{z}{\sqrt{R_{rotor}^2 + z^2}}\right)\right)$$

Niels N. Sørensen,
Risø DTU

Domain size for rotor computations



Using that $P_{available} \sim U_\infty^3$

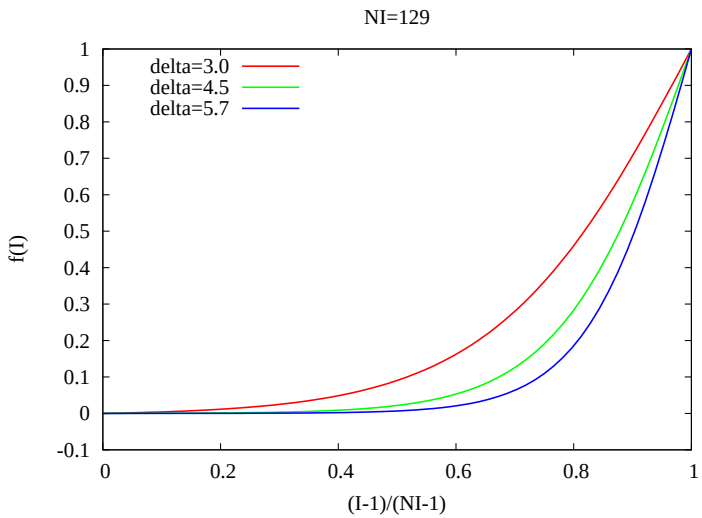
r/R	5	10	20
$\frac{P_{available} - P_\infty}{P_\infty} \times 100$	1.95	0.50	0.14

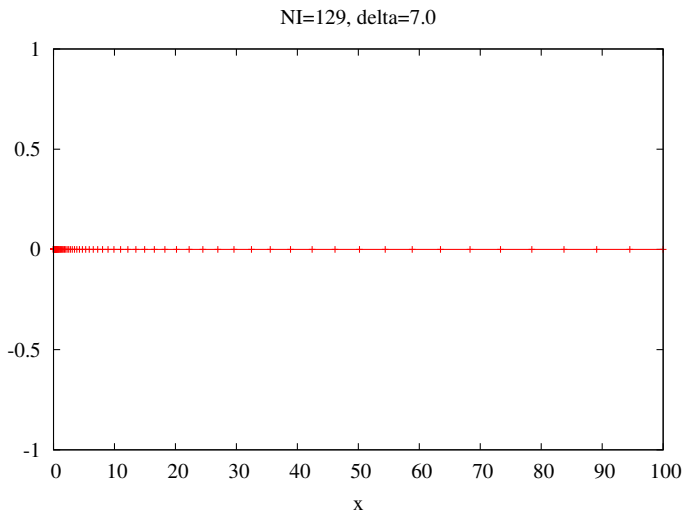
Niels N. Sørensen,
Risø DTU

Distribution function 1

Distribution function based on the hyperbolic tangent function:

$$f(I) = 1 + \frac{\tanh\left[\delta\left(\frac{I-1}{NI-1} - 1\right)\right]}{\tanh(\delta)} \text{ ,where } I = [1, NI]$$





Niels N. Sørensen,
Risø DTU

Distribution function 2

Double sided stretching function by Vinokur, based on the tanh function:

$$
\begin{aligned}
f(l) &= \frac{g(l)}{A + (1-A)g(l)} \text{ ,where } l = [1, Nl] \\
g(l) &= \frac{1}{2}\left(1 + \frac{\tanh\left(\delta\left[\frac{l-1}{Nl-1} - \frac{1}{2}\right]\right)}{\tanh\left(\frac{\delta}{2}\right)}\right) \\
B &= \frac{\sinh(\delta)}{\delta} \text{ , when } B = \frac{s}{(Nl-1)\sqrt{\Delta_1 \Delta_2}} > 1 \\
A &= \frac{\sqrt{\Delta_2}}{\sqrt{\Delta 1}}
\end{aligned}
$$

Niels N. Sørensen,
Risø DTU

Verification of the Simulation

Having performed a simulation, it is necessary to have some idea of the quality of the solution :

- Iterative Convergence
  - Are the governing equations solved on the present grid
- Grid Convergence
  - Are the solution on the present grid level independent of the grid resolution
- Comparing with Measurements
  - Do the model agree with reality

Convergence of the iterative method

Are the equations solved:

$$A_p \phi_p - \sum A_{nb} \phi_{nb} = F$$

Typically we compute the residual of the equation in each cell, using:

$$Res = \left| F - \left(A_p \phi_p - \sum A_{nb} \phi_{nb} \right) \right|$$

The sum of the residual over all cells in the computational grid is computed and compared to the starting residual.

$$Reduction = \frac{\sum_{AllCells} Res}{\sum_{AllCells} Res_0}$$

Typically a reduction of 1×10^{-4} to 1×10^{-5} is used. The fact that the residual is only changing slightly from prior iteration is not a good measure for convergence.

Grid Convergence

Is the present solution a sufficient approximation of the specified computational case?

- Often we don't know the desired solution, and the only check is to see if the numerical model is consistent and converged.
- A typical way to do this is to do consecutive grid refinements, and verify that the solution converges towards a value with the correct decrease in error e.g. 2. order.
- This procedure will only assure that we have a solution to the numerically specified problem, given by the numerical model and the boundary conditions, not that the present problem approximate the physical problem in question.

Richardson Extrapolation

Error Estimation:
Assuming that the discrete equation has order P we can write

$$\Phi = \phi_h + \alpha h^p + H \text{ , } \epsilon_h = \alpha h^p + H$$

Using this on two grid levels h and $2h$ we can estimate the error on the fine level

$$\epsilon_h \sim \frac{\phi_h - \phi_{2h}}{2^p - 1} \text{ , here assuming a doubling of the grid size}$$

The order of the scheme can be estimated using three grid levels:

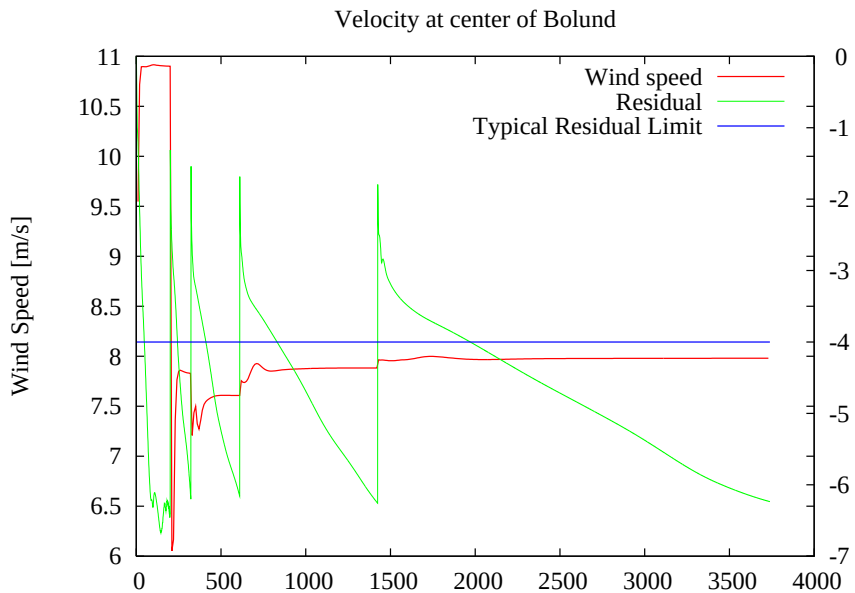
$$p = \log\left(\frac{\phi_{4h} - \phi_{2h}}{\phi_{2h} - \phi_h}\right) \frac{1}{\log(2)}$$

Here we again have assumed a doubling of the grid size. The above procedure assumes that we are in the asymptotic range, where the error is dominated by the discretization error.

Iterative Convergence

Here is an example of iterative convergence of our EllipSys code for five grid levels, from a series of computations on the Bolund blind comparison cases

- The typical residual limit of 1×10^{-4} is indicated
- For verification the convergence is taken further
- The velocity is shown at a position at the hill center

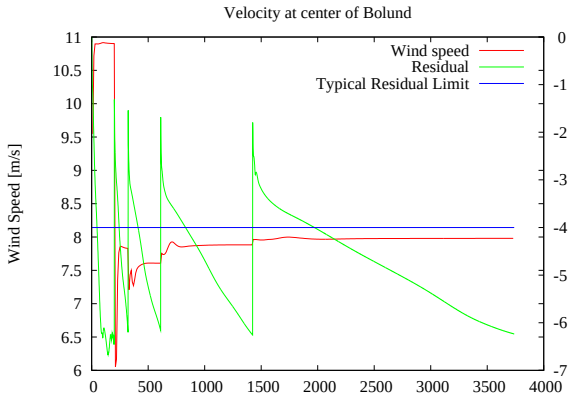


Niels N. Sørensen,
Risø DTU

Grid Convergence

Grid Level	h	2h	4h
Velocity [m/s]	7.97	7.88	7.6

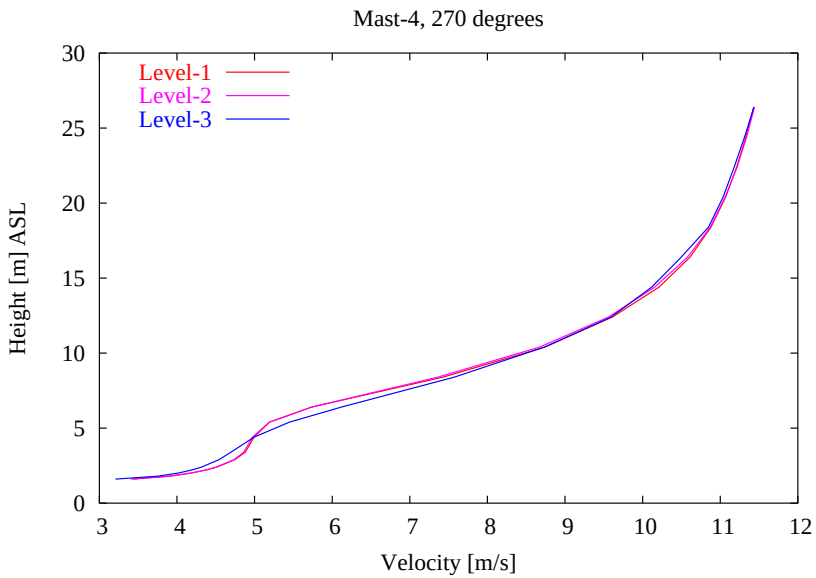
- Estimated order 1.64
- Estimated error on level one $\sim$ 0.5%

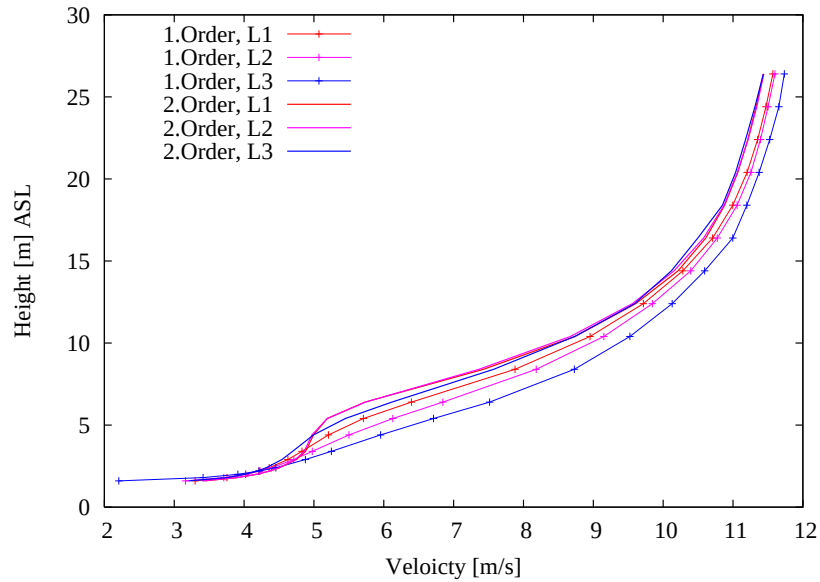


Even though the Bolund case has very complex terrain features, these are limited to a very small area $\sim 200 \times 200$ meter.

The effect of the order of the method

Comparing the solution on three grid refinements, using either a second and a first order scheme, reveals the importance of using at least a second order scheme:
The figure is taken from the bolund comparison





Niels N. Sørensen,
Risø DTU

Comparison With Measurements

Having proved that the solution is iteratively converged and grid converged we will need to confirm that the model actually agrees with the physical case in question:

- We need good experimental data
- We need well defined inflow conditions
- We need a high density of the measuring points

Test Cases

A series of test cases should be computed before doing the actual simulations:

- Airfoil Flows
  - NACA data from Abbott and Doenhoff
  - Onera-A airfoil
  - Wind turbine airfoils, Stuttgard, Delft and Risø
- Advanced Airfoil Flows
  - Pitching airfoils
- Axial Rotor Flows
  - NREL Phase VI, Axial Flow Case
  - MEXICO experiment
  - UPWIND test rotor
- Advanced Rotor Cases
  - NREL Phase VI, Yaw Cases
  - NREL Phase VI, Pitch Step