15th EAWE PhD Seminar on Wind Energy
29–31 October 2019
Centrale Nantes, France

Implementation of a pressure-based incompressible flow solver in SU2 for wind turbine applications

A. Koodly Ravishankara[a,b], H. Özdemir[a], and E.T.A. van der Weide[b]

[a]ECN part of TNO
[b]University of Twente

E-mail: akshay.koodlyravishankara@tno.nl

Abstract

Wind turbine aerodynamics can be broadly classified in the high Reynolds number and low Mach number regime. A great number of tools have been developed with incompressible and inviscid flow assumptions and have been successfully used in the wind energy industry. However, as wind turbine designs become more complicated and more efficient, higher fidelity tools like CFD become necessary. In this paper, a new open source incompressible RANS solver for wind turbine applications is introduced. The new solver is implemented within the open source multi-physics CFD suite SU2. A second order finite volume method is used for the space discretization and Euler implicit and explicit schemes for the time integration. Two turbulence models - the $k-\Omega$ mean shear stress model (SST) and the Spalart-Allamaras (SA), are available. The Bas-Cakmakcioglu (BC) transition model to capture natural transition is also available along with the SA turbulence model. A verification and validation study is carried out on the solver based on a number of standard problems together with some applications that are important in wind energy industry namely, modeling surface roughness and vortex generators.

Keywords: Computational Fluid Dynamics, Incompressible flows, SU2

1 Introduction

Flows around wind turbines generally fall under the high Reynolds number and low Mach number regime. The high Reynolds number means large regions of the flow will be inviscid except for the boundary layers and wakes. The low Mach numbers imply that the flow remains incompressible. This combination of conditions have been exploited to develop a wide variety of numerical tools based on simplified Navier-Stokes equations like blade element momentum theory, lifting line methods, panel methods, viscous inviscid interaction methods, among others[1, 2]. However, for more state-of-the-art concepts higher fidelity tools become necessary. A new open-source CFD tool for the wind energy community is presented in this paper. We hope to leverage the excellent multi-physics capabilities of SU2[3, 4] and make it available as an open source tool for the wider wind energy community.

In this paper we present a pressure based incompressible flow solver implemented within SU2 where equations are discretized on collocated unstructured grids using a second order finite volume method. The integration in time is carried out using Euler implicit and explicit methods. Two turbulence models, Spalart-Allamaras (SA)[5] and the Menter Shear Stress Transport (SST)[6], are available for turbulence modeling. Currently, the SA turbulence model has been extended to treat natural transition by the Bas-Cakmakcioglu (BC) tranisition model[7] and the Langtry-Menter[8] transition model will be incorporated as the next step.

One of the main challenges of solving the incompressible flow equations is the pressure-velocity coupling. There is no explicit equation to compute the evolution of the pressure field. For compressible flow problems, the continuity equation acts as an evolution equation for density which can be used in conjunction with the energy equation and gas law to obtain the pressure field. However, the continuity

equation reduces to a divergence condition on the mass flux for incompressible flows and the energy equation is decoupled. A new equation for the pressure is derived by combining the continuity and momentum equations (pressure projection method). The SIMPLE-like algorithms are very popular for this type and is also implemented in the current study.

Momentum interpolation methods to compute the mass flux is used to overcome the checkerboard pressure fields introduced due to the collocated grids.

The method is validated against standard test cases and then applied to various test cases to analyze different wind energy applications like vortex generators, surface roughness, flow over turbine blades among others. The resulting tool will be made available on Github under the SU2 repository (LGPL 2.1 license).

2 Model equations and numerical discretization

The general structure of the governing equations solved in SU2 is of the form[3]

$$\partial_t U + \nabla \cdot \vec{F}^c - \nabla \cdot \vec{F}^v = Q \quad \text{in} \quad \Omega, \quad t > 0, \tag{1}$$

where U is the vector of state variables, $\vec{F}^c$ are the convective flux, $\vec{F}^v$ are the viscous flux and Q is a source term. In a pressure based approach, the momentum equations and the pressure correction equation are solved sequentially.

2.1 Momentum equation

For the momentum equations, the terms in Eq. 1 are

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \vec{F}_i^c = \begin{bmatrix} \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \end{bmatrix}, \quad \vec{F}_i^v = \begin{bmatrix} \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \end{bmatrix}, \quad Q = -\vec{F}_i^p = \begin{bmatrix} \partial_1 P \\ \partial_2 P \\ \partial_3 P \end{bmatrix} \tag{2}$$

where $\vec{v} = (u_1, u_2, u_3)$ is the velocity vector, ρ is the density, P is the static pressure and the viscous stresses are $\tau_{ij} = \mu_{tot}\left(\partial_j v_i + \partial_i v_j - \frac{2}{3}\delta_{ij}\nabla \cdot \vec{v}\right)$. The total viscosity coefficient, μ_{tot} is the sum of the dynamic viscosity μ_{dyn} and turbulent viscosity μ_{tur}, which is computed via a turbulence model. The SA and the SST turbulence models are available.

2.1.1 Spatial discretization

The spatial discretization is performed on an edge based dual grid using a finite volume approach[9, 10, 11]. The control volumes are constructed using a median-dual (vertex-based) scheme[12, 3]. Integrating the Eq. 1 on the domain Ω,

$$\int_\Omega \frac{\partial U}{\partial t} d\Omega + R(U) = -F_i^p, \tag{3}$$

where $F^p = |\Omega|\nabla P$ and $R(U)$ is the residual vector consisting of the discretized convective and viscous fluxes, $\tilde{F}^c$ and $\tilde{F}^v$.

The convective fluxes are discretized using a standard upwind scheme and second order accuracy is achieved via reconstruction of variables on the cell faces by a MUSCL scheme. The viscous discretization requires the evaluation of the gradients at the faces of control volumes. The gradients at cell centers i and j can be computed using either the Green-Gauss or the least squares theorem.

2.1.2 Time integration

Following the approach outlined in [3] the solution update $\Delta U_i^n = U_i^{n+1} - U_i^n$ of an element i for implicit time stepping is

$$\left(\frac{|\Omega|}{\Delta t_i^n}\delta_{ij} + \frac{\partial R_i(U^n)}{\partial U_j}\right)\Delta U_j = -R(U^n) - F_i^p, \tag{4}$$

where n indicates the current time level. A local time stepping scheme is used to accelerate the convergence as each cell advances at a suitable local time step.

2.2 Continuity equation

The continuity equation in discrete form is simply the mass conservation equation i.e.

$$\sum_f \dot{m}_f = 0, \tag{5}$$

where f is a face between any two nodes i and j. On collocated grids the velocity at the face f is not available directly and must be interpolated using the node velocities. Using a linear interpolation to find this face velocity leads to the checkerboard problem in pressure and momentum interpolation techniques are used[13]. This can also be interpreted as adding a third order derivative of pressure to stabilize the oscillations in the pressure field.

2.2.1 Momentum interpolation of velocities

Starting from the discretized form of Eq. 4 and denoting the matrix $\left(\frac{|\Omega|}{\Delta t}\delta_{ij} + \frac{\partial R_i}{\partial U_j}\right)$ by the matrix $\boldsymbol{A}_{ij}$,

$$\boldsymbol{A}_{ij}\Delta U_j = -R(U_i^n) - F_i^p. \tag{6}$$

The velocity at any node i at time level $n+1$ can be written as

$$U_i = U_i^n + \Delta U_i = U_i^n - \frac{1}{diag(\boldsymbol{A}_{ij})}\Big(R(U^n) + \boldsymbol{N}_{ij}\Delta U_j + F_i^p\Big). \tag{7}$$

where $\boldsymbol{N}_{ij}$ consists of the off-diagonal terms of the jacobian matrix $\boldsymbol{A}_{ij}$. Hypothetically we can write the new face velocity, U_f, as

$$U_f = U_f^n + \Delta U_f = U_f^n - \frac{1}{diag(\boldsymbol{A}_{ij})}\Big(R(U^n) + \boldsymbol{N}_{ij}\Delta U_j + F_f^p\Big). \tag{8}$$

Let $B = diag(\boldsymbol{A}_{ij})^{-1}\Big(R(U^n) + \boldsymbol{N}_{ij}\Delta U_j\Big)$ and $B_f = (\lambda_i B_i + \lambda_j B_j)$. λ_i and λ_j are the weighting factors for the interpolation. Since the solution from momentum equations do not yet satisfy the continuity constraint the velocities will be denoted by U^*. Thus the velocity at a face f after the momentum equation is

$$U_f^* = \Big(\lambda_i U_i^* + \lambda_j U_j^*\Big) - \frac{|\Omega|_f}{diag(\boldsymbol{A})_f}\nabla P_f + \left(\lambda_i \frac{|\Omega|_i}{diag(\boldsymbol{A})_i}\nabla P_i + \lambda_j \frac{|\Omega|_j}{diag(\boldsymbol{A})_j}\nabla P_j\right). \tag{9}$$

The estimated face velocity can now be written as

$$U_f^* = \overline{U_f^*} - \overline{\frac{|\Omega|}{diag(\boldsymbol{A})}}\Big(\nabla P_f - \overline{\nabla P_f}\Big), \tag{10}$$

where $\overline{\frac{|\Omega|}{diag(\boldsymbol{A})}}$ and $\overline{U^*}_f$ are linearly interpolated using λ_i and λ_j.

2.2.2 Pressure Correction equation

Let the velocity corrections be defined as U', pressure corrections as P'. Similar to the velocity estimate at a face, the velocity correction relation can be written as,

$$U_f' = \overline{U_f'} - \overline{\frac{|\Omega|}{diag(\boldsymbol{A})}}\Big(\nabla P_f' - \overline{\nabla P_f'}\Big). \tag{11}$$

Expressing the continuity equation in terms of velocity estimate and corrections,

$$\sum_f \dot{m}_f = \sum_f (\dot{m}_f^* + \dot{m}_f') = 0, \tag{12}$$

where $\dot{m}_f^*$ and $\dot{m}_f'$ are the estimate and correction of the mass flux respectively.

From Eqs. 10, 11 and discrete continuity equation,

$$\sum_f \rho\Big(\overline{U_f'} - \overline{\frac{|\Omega|}{diag(\boldsymbol{A})}}\Big(\nabla P_f' - \overline{\nabla P_f'}\Big)\Big) \cdot \vec{n}_f = -\sum_f \dot{m}_f^*, \tag{13}$$

The terms under the overbar depend directly on the neighbors and are neglected (this is the SIMPLE assumption). Thus, we have an equation for the pressure correction as,

$$-\sum_f \rho\overline{\frac{|\Omega|}{diag(\boldsymbol{A})}}\Big(\nabla P_f'\Big) \cdot \vec{n}_f = -\sum_f \dot{m}_f^*. \tag{14}$$

The term $\sum_f \dot{m}_f^*$ is calculated using the estimated velocities U_f^* in Eq. 10 and is treated as a source term. Eq 14 is a Poisson type equation for pressure correction which has to be solved sequentially with the momentum equations. Finally, the pressure and velocities at node i are corrected as

$$P_i = P_i^n + \alpha_p P', \tag{15}$$
$$U_i = U_i^n + D_i \nabla P_i' \tag{16}$$

2.3 SIMPLE family of algorithms

The solution process described above is commonly known as the SIMPLE[14] algorithm. Many derivatives like SIMPLEC, PISO[9, 15, 16, 17] are also available. The basic SIMPLE algorithm for collocated grids is described below

1. Set the solution at n as the initial guess.
2. Solve the momentum equations, Eq. 4, to find the estimated velocity U^*.
3. Find the mass flux, m_f^*, at the faces using the velocities from Eq. 10.
4. Assemble pressure correction equation based on the mass fluxes and the momentum equation.
5. Solve the pressure correction equation, Eq. 14, to find the pressure and velocity corrections based on Eqs. 15 and 16.
6. Set the updated solutions as the solutions at time $n+1$ and solve other scalar equations (e.g. turbulence).

2.4 Turbulence modeling

SU2 currently supports two RANS models, namely, the Spalart-Allamaras (SA)[5] and the Mean Shear Stress Transport (SST)[6] as mentioned above. The turbulence equations are solved in a segregated manner. Currently, the algebraic Bas-Cakmakcioglu (BC) transition model[7] is available with the SA turbulence model. The governing equations are omitted for the sake of brevity and can be found in [3, 4].

3 Results

3.1 Laminar flow: Channel flow with analytical solution

To verify the order of accuracy of the solver, a fully developed laminar channel flow with $Re = 400$ is chosen. Under the fully developed flow assumptions, the velocity profile can be computed as

$$u(y) = -\frac{dP}{dx}\frac{1}{2\mu}(h^2 - y^2). \tag{17}$$

Three different mesh resolutions are chosen and the numerical results are compared to the analytical solution (Fig. 1, left). The error in the numerical solution is computed for the three meshes and the order of accuracy can be seen to be second order (Fig. 1, right), as expected.

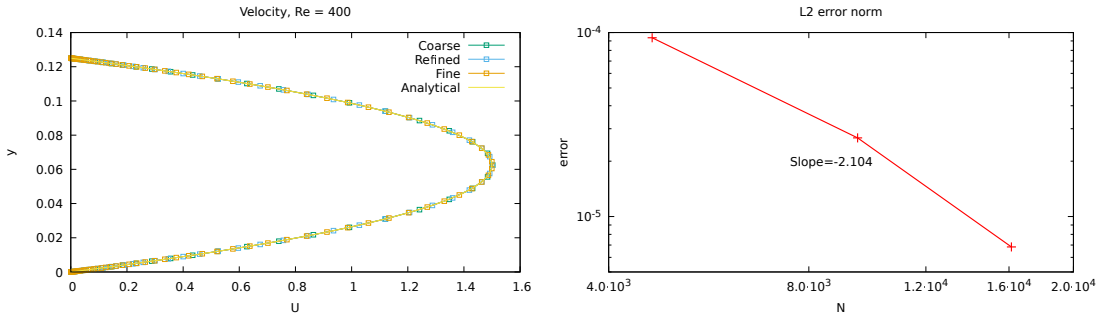
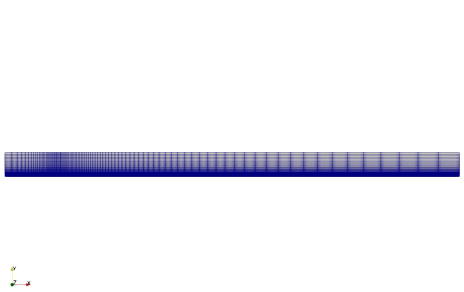


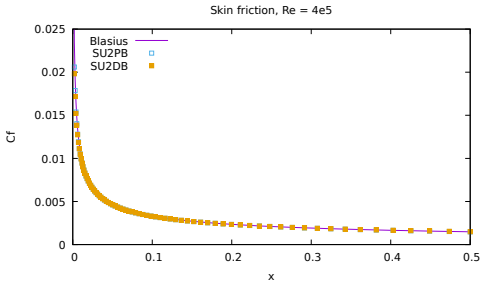
Figure 1: Channel flow. Comparison of numerical solutions and analytical result (left), L2 norm of the error (right)

3.2 Laminar flow over a flat plate compared with Blasius solution

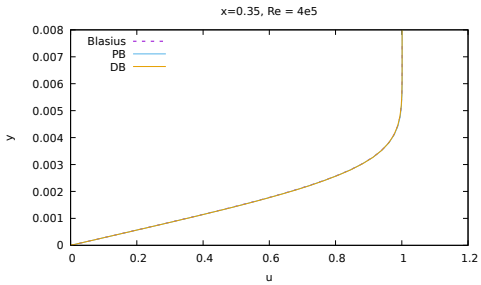
The results from the laminar flow over a flat plate ($Re = 4.0e5$) is compared to the Blasius solution [18] at different locations (Figs. 2c, 2d). A uniform inflow is prescribed and a small inflow region with a symmetry boundary is used before the flat plate begins (Fig. 2a). Skin friction shows excellent



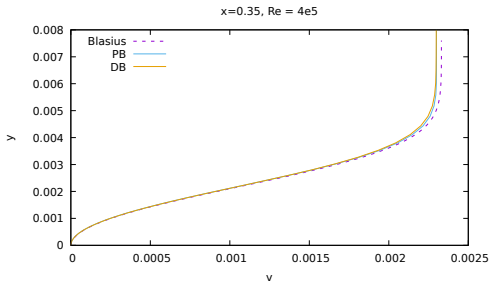
(a) Domain of flat plate



(b) Skin friction on flat plate.



(c) u at $x = 0.35$



(d) v at $x = 0.35$.

Figure 2: Comparison of numerical velocity to Blasius solution.

agreement for both cases (Fig. 2b). The results for the stream-wise component (Fig. 2c) of the velocity shows excellent agreement between the numerical and Blasius solutions for both the density-based and pressure-based methods. The comparison for the normal velocity component (Fig. 2d) is not as good but it can be seen that the pressure-based method does better than the density based method.

3.3 Turbulent flow over NACA0012 airfoil

Fully turbulent flow over NACA0012 airfoil employing the SA turbulent model is compared with the experimental data[19] at a Re number of $6.0e6$ on a grid with approximately 14000 elements. The pressure-based method matches the experimental data very closely at all angles of attack and also captures the maximum lift angle.

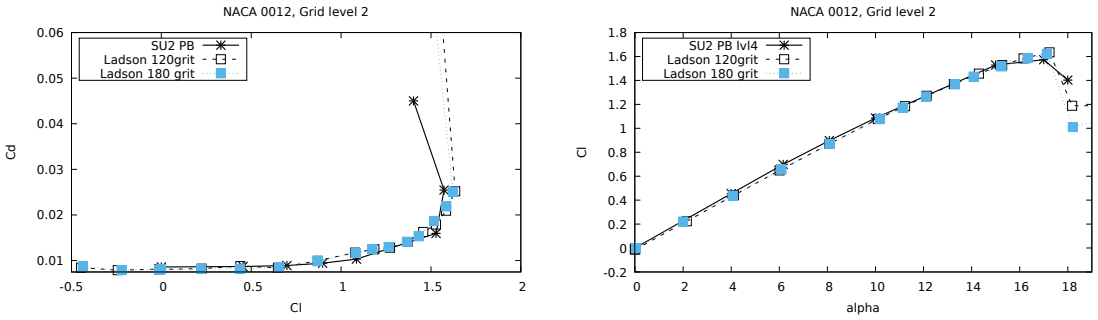


Figure 3: Turbulent flow over NACA0012 airfoil. A comparison of the numerical solution with the experimental data for lift to drag ratio (left) and lift coefficient for various angle of attacks (right).

3.4 Vortex generators

The aim of this work is to use CFD simulations to develop an empirical model to account for the effect of vortex generators (VG) in an integral boundary layer (IBL) method like RFOIL[2]. To this end, a number of CFD simulations were run on flat plates and airfoils with and without VGs. The vortices created by the VGs enhances the mixing in the boundary layer which help keep the boundary layer remain attached for longer. To model this effect in an IBL method, the VG is assumed to create two distinct regions of different velocities which mix downstream of the VG. This mixing layer can be clearly seen from the CFD results (Fig. 4). To define the mixing layer, two characteristic velocities U_l and U_h are defined as follows

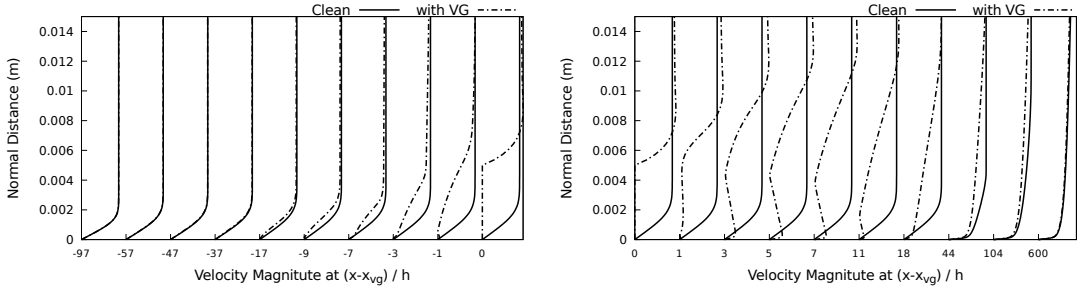


Figure 4: Comparison of velocity profiles with and without VG at various locations (around VG, left, and downstream of VG, right).

$$U_l = \frac{\int_0^{h_{VG}} (\rho u) u dy}{\int_0^{h_{VG}} (\rho u) dy}, \quad U_h = \frac{\int_{h_{VG}}^{\delta} (\rho u) u dy}{\int_{h_{VG}}^{\delta} (\rho u) dy}, \tag{18}$$

where h_{VG} is the height of the VG and δ, the boundary layer thickness. In a plane mixing layer, the scaled velocity profile is typically self similar[20]. Using the scaling relations defined in [20] and the characteristic velocities defined in Eqs. 18 the scaled velocities can be seen to be self similar (Fig. 5). However, unlike plane mixing layers where the flow can extend unhindered in the transverse direction, downstream of a VG the wall stops the development of the mixing layer after a certain distance (Fig. 5, right). Traditionally, the mean velocity profile over the whole boundary layer is represented by a sum of two functions[20, 21]

$$\frac{\overline{U}}{u_\tau} = f_w\left(\frac{y}{\delta_v}\right) + \frac{\Pi}{\kappa} w\left(\frac{y}{\delta}\right). \tag{19}$$

The term $f_w\left(\frac{y}{\delta_v}\right)$ represents the law of the wall which is defined based on wall units. $\overline{U}$ is the mean velocity, $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ is the friction velocity. The second function is typically called the law of the wake

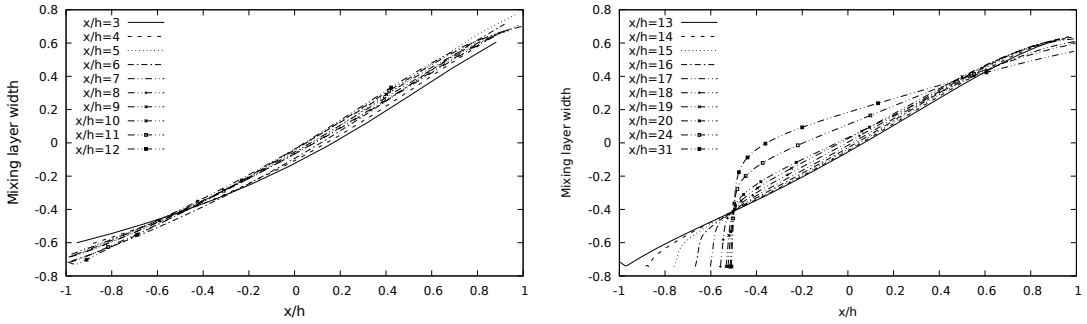


Figure 5: Scaled velocity and the cross stream distance (around VG, left, and downstream of VG, right).

which depends on y/δ which is assumed to be universal. The parameter Π is called the wake strength parameter and is flow dependent. This function can now be modified based on results from the mixing layer to account for the presence of the VG.

3.5 Surface roughness model

Erosion of wind turbine blades can lead performance losses. Quantifying the effect of surface roughness is very important and in this study, we use roughness correction on the SA turbulence model[22] and compare numerical results with other numerical[23] and experimental data[24]. The roughness is assumed to be distributed over a specified region and is characterized using an "equivalent sand grain roughness height (k_S)". The viscous sublayer ([20], also described in Section 3.4) behaves differently on a rough wall compared to a smooth one[18]. This difference is manifested in the law of the wall (represented by the term $f_w\left(\frac{y}{\delta_v}\right)$ in Eq. 19) which can be accounted for in CFD simulations by using a modified boundary condition described in [22]. It should be noted that this modification is only for the turbulent flow region and the transition model is not modified and thus the laminar to turbulent transition process is unaffected. To more accurately capture the effect of roughness, the transition prediction model should also be updated[23].

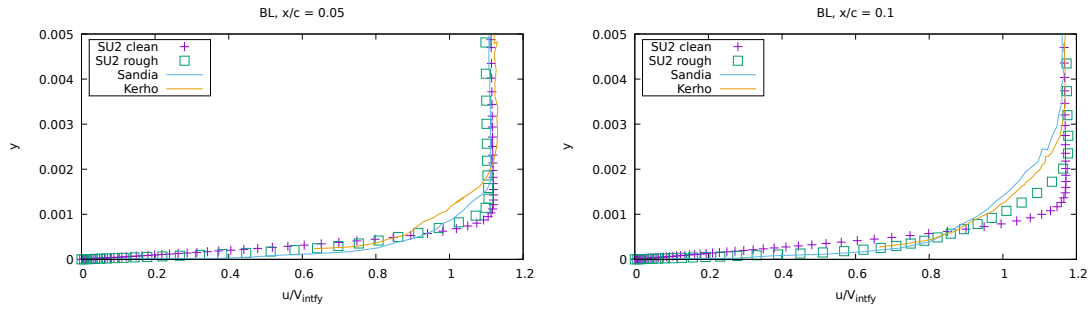


Figure 6: Comparison of clean and rough velocity profiles between numerical solutions and experimental data at $x/c = 0.005$ (left) and at $x/c = 0.01$ (right).

References

[1] Manwell J, Mcgowan J and L Rogers A 2006 *Wind Energy Explained: Theory, Design and Application, Second Edition* vol 30

[2] Van Rooij R 1996 Modification of the boundary layer calculation in rfoil for improved airfoil stall prediction

[3] Palacios F, Economon T D, Aranake A, Copeland S R, Lonkar A K, Lukaczyk T W, Manosalvas D E, Naik K R, Padron S, Tracey B, Variyar A and Alonso J J 2014 *52nd Aerospace Sciences Meeting* (American Institute of Aeronautics and Astronautics) URL https://doi.org/10.2514/6.2014-0243

[4] Duraisamy K, Copeland S, Economon T, Colonno M, Aranake A, Taylor T, Lukaczyk T, Palacios F, Lonkar A, Hicken J, Alonso J and Campos A 2013 1–60

[5] SPALART P and ALLMARAS S *A one-equation turbulence model for aerodynamic flows*

[6] Menter F R 1994 *AIAA Journal* **32** 1598–1605

[7] Cakmakcioglu S C, Bas O and Kaynak U 2018 *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* **232** 3915–3929

[8] Menter F R, Langtry R and Völker S 2006 *Flow, Turbulence and Combustion* **77** 277–303

[9] Moukalled F, Mangani L and Darwish M 2015 *Fluid Mechanics and Its Applications The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM® and Matlab®* (Springer)

[10] Ferziger J H, Perić M and Peric M 2002 *Computational Methods for Fluid Dynamics* 3rd ed (Springer) ISBN 3540420746

[11] Blazek J 2005 *Computational Fluid Dynamics: Principles and Applications (Second Edition)* ed Blazek J (Oxford: Elsevier Science) pp 131 – 182 second edition ed

[12] Barth T J 1992

[13] Rhie C M and Chowt W L 1983 *AIAA Journal* **21**

[14] Patankar S 1980 *Numerical heat transfer and fluid flow* (CRC press)

[15] Murthy J Y and Mathur S 2002 *International Journal for Numerical Methods in Fluids* **25** 659–677 ISSN 02712091

[16] Xiao H, Wang J, Liu Z and Liu W 2018 *International Journal of Heat and Mass Transfer* **120** 1255–1265 ISSN 00179310

[17] Yen R H and Liu C H 1993 *Numerical Heat Transfer, Part B: Fundamentals* **24** 127–141

[18] Lyon H 2014 *Boundary Layer Theory* vol 7 ISBN 9783662529171

[19] Ladson C L 1988

[20] Pope S B 2000 *Turbulent Flows* (Cambridge University Press)

[21] Coles D 1956 *Journal of Fluid Mechanics* **1** 191–226

[22] Aupoix B and Spalart P 2003 *International Journal of Heat and Fluid Flow* **24** 454–462

[23] Langel C M, Chow R and van Dam C Rans based methodology for predicting the influence of leading edge erosion on airfoil performance Tech. rep.

[24] Kerho M F and Bragg M B 1997 *AIAA Journal* **35** 75–84