



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

Akshay Koodly Ravishankara^{*†}
ECN part of TNO, Petten, 1755 LE The Netherlands
University of Twente, 7500 AE, Enschede, The Netherlands

Hüseyin Özdemir[‡]
ECN part of TNO, Petten, 1755 LE, The Netherlands

Edwin van der Weide[§]
University of Twente, 7500 AE, Enschede, The Netherlands

Wind turbine aerodynamics can be broadly classified in the high Reynolds number and low Mach number regime. Flows in this regime are generally incompressible and have large regions where they can be considered as inviscid. Thus, a great number of tools have been developed with incompressible and inviscid flow assumptions. However, as wind turbines designs become more complicated and more efficient, higher fidelity and more accurate tools like CFD are necessary. In this paper, a new open source pressure based incompressible RANS solver for wind turbine applications is introduced. The new solver is implemented within the open source multiphysics 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-Allmaras model, are available. A verification and validation study is carried out on the solver based on a number of standard problems and finally an investigation into the effect of a vortex generator on turbulent boundary layer is presented.

I. Nomenclature

$\dot{m}_f$	Mass flux across a face f	μ_{tot}	Total viscosity
Ω	Domain of the problem	μ_{tur}	Turbulent viscosity
$\vec{n}$	Normal vector	ρ	Density
C_d	Drag coefficient	<i>Superscripts</i>	
C_f	Skin friction coefficient	*	Estimate
P	Pressure	$\prime$	Correction
U	Solution vector	c	Convective fluxes
<i>Symbols</i>		n	Current time level
μ_{dyn}	Dynamic viscosity	p	Pressure contribution
		v	Viscous fluxes

^{*} Junior Scientist Innovator, Wind Energy, Westerdunweg 3, akshay.koodlyravishankara@tno.nl

[†] PhD Candidate, Engineering Fluid Dynamics Group, a.koodlyravishankara@utwente.nl

[‡] Researcher, Wind Energy, Westerdunweg 3, huseyin.ozdemir@tno.nl, Member AIAA

[§] Associate Professor, e.t.a.vanderweide@utwente.nl, Senior Member AIAA

II. Introduction

Flows around wind turbines generally fall under the high Reynolds number and low Mach number regime that is somewhat unique and greatly advantageous in terms of numerical analysis. The high Reynolds number means large regions of the flow can be considered inviscid except for a small region around the body known as the boundary layer. 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 simplifications of the Navier-Stokes equations. Blade element momentum theory, based on global momentum balance across the wind turbine[1], has been extremely useful in determining wind turbine behavior and has been extended using empirical corrections[2] to account for more complicated effects. Other numerical methods based on inviscid theory like panel methods and lifting line theory[3] have been successfully used to compute flow around a wind turbine blade including the wake. Viscous inviscid interaction tools like RFOIL[4, 5] can be used in combination with the lifting line methods to account for the presence of a boundary layer on turbine blades. However as the size of the turbine blades has increased, issues like thick airfoils, transition modeling are becoming more important. Additionally, new concepts to improve the efficiency of the turbines (e.g. vortex generators) are becoming more common. While it is possible to extend the existing tools like RFOIL to account for some of the new problems[6–8] arising out of modern wind turbine blades, they are still limited in their scope of applicability. Thus, a higher fidelity general purpose tool like CFD becomes necessary. This, among other factors, motivated the development of a new open-source CFD method for the wind energy community. While other tools exist (e.g. OpenFOAM [9] (also open source), EllipSys2D/3D[10, 11]), we hope to leverage the excellent multi-physics capabilities of SU2[12, 13] and make it available for the wider wind energy community. Apart from wind turbine blade analysis, CFD methods has been extensively used for wind farm analysis[14] as well.

In this paper we present a pressure based incompressible RANS solver implemented within the open source multiphysics suite SU2[12, 13]. The 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-Allmaras (SA)[15] and the Menter Shear Stress Transport (SST)[16], are available for turbulence modeling. Currently, the SA turbulence model has been extended to treat natural transition by the Bas-Cakmakcioglu (BC) transition model[17] and the Langtry-Menter[18] transition model will be incorporated as the next step.

The main challenge in solving the incompressible flow equations lies in resolving the pressure-velocity coupling[19–23]. Physically this challenge arises due to the pressure disturbances travelling at the speed of sound which, under the incompressible flow assumption, is infinitely fast and leads to numerical problems. For compressible flow problems (with a finite speed of sound), the continuity equation acts as an evolution equation for density which can be used in conjunction with the energy equation and a gas law to obtain the pressure field. However as the continuity equation reduces to a divergence condition on the mass flux for incompressible flows and the energy equation is decoupled, there is no direct way to compute the pressure field. To alleviate this, the pressure can be eliminated from the equations using derived quantities like stream function and vorticity[24] which can then be solved to obtain the flow field. Such methods, however, do not generalize well into 3-D and the use of primitive variables (pressure and velocity) is preferred. The pressure-velocity coupling can be overcome using primitive variables in a variety of ways which are broadly classified as either "pressure-based" or "density-based" methods. An example of the density based method is the pseudo compressibility approach[19, 25, 26] where an artificial speed of sound is introduced in the continuity equation to mimic the compressible flow formulation. This method belongs to the more general approach of pre-conditioned compressible flow solution methods[27]. The existing incompressible solver in SU2 follows this approach and has been extended to variable-density flows and heat transfer applications[27]. Alternatively in a pressure based method the continuity equation is satisfied directly[20]. The combination of the continuity equations and the momentum equations can be used to obtain a condition for pressure which can then be used to solve the continuity and momentum equations. The earliest such approach is the marker-and-cell (MAC) method[28] where pressure is used as a mapping parameter to satisfy the continuity equation and a Poisson equation is solved for pressure. However, this method was expensive due to its iterative nature and requirement of a Poisson solution in every iteration. This was improved upon by solving for pressure and velocity corrections instead of pressure leading to faster convergence. The semi-implicit pressure linked equations (SIMPLE) and SIMPLE-like algorithms[29, 30] are very popular for this type and is also implemented in the current paper.

Originally, these methods were developed for staggered grids, where pressure is stored at cell centers and velocities on the cell faces. This conserves the mass and also avoids the odd-even decoupling of pressure. Extension to collocated grids needs special attention. Momentum interpolation methods to compute the mass flux based on the formulation introduced by Rhie and Chow in [31] is adapted in the current paper to avoid the odd-even decoupling of pressure. Numerous authors[32–36] have pointed out the dependence of the original interpolation scheme on factors such as

under-relaxation and time step size. Consequently, the original momentum interpolation scheme is adapted to account for such factors.

In the following sections, first the governing equations and the discretization schemes are described, followed by a section on verification and validation based on standard problems. Finally, the effect of the presence of a vortex generator on a turbulent boundary layer is examined.

III. Model equations and numerical discretization

The general structure of the PDE solved in SU2 is of the form[13]

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

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. The pressure correction equation is derived based on a combination of momentum and continuity equations. The following sections briefly outline the derivation.

A. Momentum equation

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

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

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 $\tau_{ij} = \mu_{tot}(\partial_j v_i + \partial_i v_j - \frac{2}{3} \delta_{ij} \nabla \cdot \vec{v})$. 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 Spalart-Allmaras(S-A) and the Mean Shear Stress Transport(SST) turbulence models can be used to compute μ_{tur} .

1. Spatial discretization

The spatial discretization is performed on an edge based dual grid using a finite volume approach[21, 22, 37]. The control volumes are constructed using a median-dual (vertex-based) scheme[13, 38]. Integrating the Eq. 1 on the domain Ω ,

$$\int_{\Omega} \frac{\partial U}{\partial t} d\Omega + \int_{\Omega} \nabla \cdot (\vec{F}^c - \vec{F}^v) d\Omega = - \int_{\Omega} \nabla P d\Omega, \quad (3)$$

$$\int_{\Omega} \frac{\partial U}{\partial t} d\Omega + \int_{\partial\Omega} (\vec{F}^c - \vec{F}^v) \cdot \vec{dS} = - \int_{\partial\Omega} P \vec{I} \cdot \vec{dS},$$

$$\int_{\Omega} \frac{\partial U}{\partial t} d\Omega + R(U) = -F^p, \quad (4)$$

where $F^p = |\Omega| \nabla P$ is the pressure contribution treated as a source term and $R(U)$ is the residual vector consisting of the discretized convective and viscous fluxes, $\vec{F}^c$ and $\vec{F}^v$.

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. For a general variable ϕ this gradient is evaluated as

$$\nabla \phi \cdot \vec{n} = \frac{\phi_j - \phi_i}{|x_j - x_i|} \alpha_f + \frac{1}{2} (\nabla \phi|_i + \nabla \phi|_j) \cdot (\vec{n} - \alpha_f \vec{s}), \quad (5)$$

where $\vec{n}$ is the face normal, $\vec{s}$ is the normalised vector connecting the cell centers i and j across the face, $|x_j - x_i|$ is the distance between the nodes i and j and $\alpha_f = \vec{s} \cdot \vec{n}$. The gradients at cell centers i and j can be computed using either the Green-Gauss or the least squares theorem.

2. Time integration

Following the approach outlined in [13] the solution update $\Delta U_i^n = U_i^{n+1} - U_i^n$ of an element i for implicit time stepping is

$$\left(\frac{|\Omega|}{\Delta t^n} \delta_{ij} + \frac{\partial R_i(U^n)}{\partial U_j} \right) \Delta U_j = -R(U^n) - F_i^P, \quad (6)$$

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.

B. Continuity equation

The continuity equation in the discrete form is

$$\sum_f \dot{m}_f = \sum_f \rho \vec{U}_f \cdot \vec{n}_f = 0, \quad (7)$$

where $\vec{U}_f$ is the velocity at a face f , ρ is the fluid density and $\vec{n}_f$ is the face normal. Using a linear interpolation to find this face velocity leads to the checkerboard problem in pressure and thus momentum interpolation techniques are used. This can also be interpreted as adding a third order derivative of pressure to stabilize the oscillations in the pressure field.

1. Momentum interpolation of velocities

Consider a face f between two nodes P and E . Denoting the matrix $\left(\frac{|\Omega|}{\Delta t} \delta_{ij} + \frac{\partial R_i}{\partial U_j} \right)$ in Eq. 6 by the matrix A_{ij} , we have

$$A_{ij} \Delta U_j = -R(U_i^n) - F_i^P,$$

which can also be written as

$$\text{diag}(A_{ij}) \Delta U_j + N_{ij} \Delta U_j = -R(U_i^n) - F_i^P,$$

where N_{ij} consists of the off-diagonal terms of the jacobian matrix A_{ij} . Thus, the velocity at any two points P and E at time level $n + 1$ can be written as

$$U_P = U_P^n + \Delta U_P = U_P^n - \frac{1}{\text{diag}(\mathbf{A})_P} \left(R(U^n) + N_P \Delta U_{N(P)} + F_P^P \right), \quad (8)$$

$$U_E = U_E^n + \Delta U_E = U_E^n - \frac{1}{\text{diag}(\mathbf{A})_E} \left(R(U^n) + N_E \Delta U_{N(E)} + F_E^P \right). \quad (9)$$

Here $N(P)$ represents the neighbors of any node P . We can hypothetically write the new velocity at a face f between P and E , U_f as,

$$U_f = U_f^n + \Delta U_f = U_f^n - \frac{1}{\text{diag}(\mathbf{A})_f} \left(R(U^n) + N_f \Delta U_{N(f)} + F_f^P \right). \quad (10)$$

Denoting $B_P = \text{diag}(\mathbf{A})_P^{-1} \left(R(U_P^n) + N \Delta U_{N(P)} \right)$, we can write

$$B_f = (\lambda_P B_P + \lambda_E B_E), \quad (11)$$

where λ_P and λ_E are the weighting factors for the interpolation. Using Eqs. 8, 9, 10 and 11 and expanding the pressure source term F^P from Eq. 4, the velocity at a face f after the momentum equation is

$$U_f^* = \left(\lambda_P U_P^* + \lambda_E U_E^* \right) - \frac{|\Omega|_f}{\text{diag}(\mathbf{A})_f} \nabla P_f + \left(\lambda_P \frac{|\Omega|_P}{\text{diag}(\mathbf{A})_P} \nabla P_P + \lambda_E \frac{|\Omega|_E}{\text{diag}(\mathbf{A})_E} \nabla P_E \right). \quad (12)$$

First, we assume

$$\left(\lambda_P \frac{|\Omega|_P}{\text{diag}(\mathbf{A})_P} \nabla P_P + \lambda_E \frac{|\Omega|_E}{\text{diag}(\mathbf{A})_E} \nabla P_E \right) = \frac{|\Omega|}{\text{diag}(\mathbf{A})} \nabla P_f \quad (13)$$

and let

$$\frac{|\Omega|_f}{diag(\mathbf{A})_f} = \frac{\overline{|\Omega|}}{diag(\mathbf{A})},$$

$$\overline{U}_f = (\lambda_P U_P + \lambda_E U_E).$$

The estimated face velocity can now be written as

$$U_f^* = \overline{U}_f - \frac{\overline{|\Omega|}}{diag(\mathbf{A})} (\nabla P_f - \overline{\nabla P_f}). \quad (14)$$

Since the solution from momentum equations do not yet satisfy the continuity constraint the velocities are denoted by U^* . The assumption made in Eq. 13 can be shown to be second order accurate[22]. The derivations so far follow the procedure outlined by Rhie-Chow[31] and no corrections[32–36, 39] have been applied to account for under-relaxation and time stepping.

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 based on pressure corrections can be written as,

$$U'_f = \overline{U}'_f - \frac{\overline{|\Omega|}}{diag(\mathbf{A})} (\nabla P'_f - \overline{\nabla P'_f}). \quad (15)$$

Rewriting Eq. 7 in terms of estimated velocity and velocity corrections,

$$\sum_f \dot{m}_f = \sum_f (\dot{m}_f^* + \dot{m}'_f) = 0, \quad (16)$$

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

From Eqs. 14, 15 and 7,

$$\sum_f \rho \left(\overline{U}_f - \frac{\overline{|\Omega|}}{diag(\mathbf{A})} (\nabla P_f - \overline{\nabla P_f}) \right) \cdot \vec{n}_f = - \sum_f \dot{m}_f^*, \quad (17)$$

$$- \sum_f \rho \frac{\overline{|\Omega|}}{diag(\mathbf{A})} (\nabla P'_f) \cdot \vec{n}_f = - \sum_f \dot{m}_f^* - \sum_f \rho \overline{U}'_f \cdot \vec{n}_f - \sum_f \rho \frac{\overline{|\Omega|}}{diag(\mathbf{A})} \overline{\nabla P'_f} \cdot \vec{n}_f. \quad (18)$$

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

$$- \sum_f \rho \frac{\overline{|\Omega|}}{diag(\mathbf{A})} (\nabla P'_f) \cdot \vec{n}_f = - \sum_f \dot{m}_f^*. \quad (19)$$

The term $\sum_f \dot{m}_f^*$ is calculated using the estimated velocities U_f^* in Eq. 14 and is treated as a source term. Eq 19 is a Poisson equation for the pressure correction which has to be solved sequentially with the momentum equations. No under-relaxation is used for the Poisson solver. A multigrid method can be applied specifically for the Poisson problem to speed up the convergence, especially for unsteady problems.

3. Pressure and velocity corrections

Finally, based on the solution to the pressure correction and the pressure correction field, the pressure and velocities at a node P can be corrected as

$$P_P^{n+1} = P_P^* + \alpha_P P', \quad (20)$$

$$U_P^{n+1} = U_P^* + D_P \nabla P'_P. \quad (21)$$

Here, $D_P = \frac{|\Omega|}{diag(\mathbf{A})}$ is the ratio of the volume of the element to the momentum equation coefficients at the node P and α_P is the under-relaxation factor which is explained in more detail in the following section. Typically, the velocity correction is also under-relaxed but instead here the pseudo time-stepping scheme is used.

C. SIMPLE family of algorithms

There are many popular schemes for the pressure-based methods and one of the more widely used algorithm is the SIMPLE[30] and its derivatives[22, 40–42]. In this paper, the SIMPLE, SIMPLEC and PISO algorithms are used.

1. SIMPLE/SIMPLEC

In summary, to find the solution at time $n + 1$ the following algorithm is employed:

- 1) Set the solution at n as the initial guess.
- 2) Solve the momentum equations Eq. 6 to find the estimated velocity U^* .
- 3) Find the mass flux at the faces m_f^* using the velocities from Eq. 14.
- 4) Assemble the pressure correction equation based on the mass fluxes and the momentum equation.
- 5) Solve the pressure correction equation (Eq. 19) to find the pressure and velocity corrections based on Eqs. 21 and 20.
- 6) The updated solutions are set as the solution at time $n + 1$ and solve other scalar equations, like turbulence.

One of the key assumptions involved in deriving the pressure correction equation was neglecting the terms involving the pressure and velocity corrections of neighbors. While this does not affect the final solution as all corrections go to zero at convergence, it does slow down the rate of convergence. Many improvements are employed to improve the speed of convergence.

One such improvement is to modify the under-relaxation used during the pressure correction in Eq. 20. The underrelaxation factor for pressure, α_P is set as $1 - \alpha_U$, where α_U is the velocity under-relaxation. While there is no explicit under-relaxation applied to the velocity equation, the pseudo time step acts like an under-relaxation. Thus, the pressure under-relaxation is

$$\alpha_P = \frac{(|\Omega|/\Delta t)_{P,U}}{1 + (|\Omega|/\Delta t)_{P,U}}. \quad (22)$$

Another variation that is implemented in the current work is the SIMPLEC algorithm. The velocity correction at any node P is assumed to be the weighted average of the corrections at the neighboring points. This leads to a smaller term being neglected from the pressure correction, thus improving the speed of convergence. There is only a modification of the coefficients of the pressure correction equation compared to the SIMPLE algorithm and the sequence of operations remains the same.

2. PISO

In this variation, an additional correction step is employed. The same sequence of operations outlined for SIMPLE is followed, and the corrected velocity and pressure field is used to explicitly solve the momentum equations to find a new estimate of the velocity field. Based on the explicit velocity solution, the mass imbalance is computed once again and the pressure correction is solved to find a newer estimate of velocity. The explicit solution recovers a portion of the neglected terms and aids in increasing the rate of convergence.

D. Boundary conditions

The boundary conditions available are Euler (flow tangency), no slip wall, velocity inlet, pressure outlet and symmetry boundaries. Eq. 6 is discretized using the median-dual or vertex-centered approach which is different from the conventional approaches[22, 37, 43, 44]. The node for the boundary control volume lies on the boundary face itself and thus special attention is required during application of boundary conditions[37]. Additionally, there is no need for momentum interpolation of velocities to compute mass flux across the face as the velocity on the boundary face is available from the solution.

The application of each of these boundaries for the momentum equations, mass flux computation and pressure correction equation is described below.

1. Momentum equations

- 1) Euler or flow tangency condition: This boundary condition specifies a zero normal flux across the boundary (e.g. inviscid wall). For the momentum equations, this is applied as a Neumann boundary with zero flux across the face. Mass flux across the face is also set to zero.

- 2) Wall (no-slip): This is a strong boundary condition and is generally used to impose a no slip condition on walls. Since the discretization is vertex-based the boundary node lies on boundary face and thus the velocity boundary condition can be enforced as a Dirichlet boundary condition. Mass flux across the face is also set to zero.
- 3) Inlet: For a prescribed velocity at the inlet, the velocity is imposed as a Dirichlet boundary condition similar to the wall boundary. However, the mass flux is not zero but can be easily computed based on the specified velocity.
- 4) Outlet: For a specified pressure outlet, a Neumann boundary condition is applied at the outlet. Similarly, the mass flux across the face is also computed using the latest estimate of the velocity.
- 5) Symmetry: A symmetry boundary does not only imply a zero flux across the face but also a reflection of the solution state across the boundary face. Consequently, a reflected state of the current state is computed and a Neumann boundary condition is applied. Mass flux across the face is set to zero.
- 6) Farfield: Farfield boundaries are generally used in external flow simulations to denote the freestream conditions. This is treated as an inlet-outlet type boundary where a Dirichlet condition is used for incoming flow and a Neumann boundary for outgoing flow based on the nature of the flux at the boundary face.

2. Pressure correction equations

If the pressure at a particular boundary is unknown (Euler wall, Wall, Inlet, Symmetry) it is treated as a Neumann boundary and the value of the pressure is updated based on the pressure correction. However, if the value of pressure is specified (Outlet with a specified pressure), the value of the pressure is fixed and the pressure correction is set to zero as a Dirichlet boundary condition.

IV. Results

A. Verification

1. Laminar flow: Channel flow with analytical solution

To verify the order of accuracy of the solver, a fully developed laminar channel flow (schematic in Fig. 1a) with $Re = 400$ is chosen. Under the fully developed flow assumptions, the velocity profile at any location y can be computed as

$$u(y) = -\frac{dP}{dx} \frac{1}{2\mu} (h^2 - y^2), \quad (23)$$

where $\frac{dP}{dx}$ is the pressure gradient across the channel, μ is the laminar viscosity, h is the half channel width.

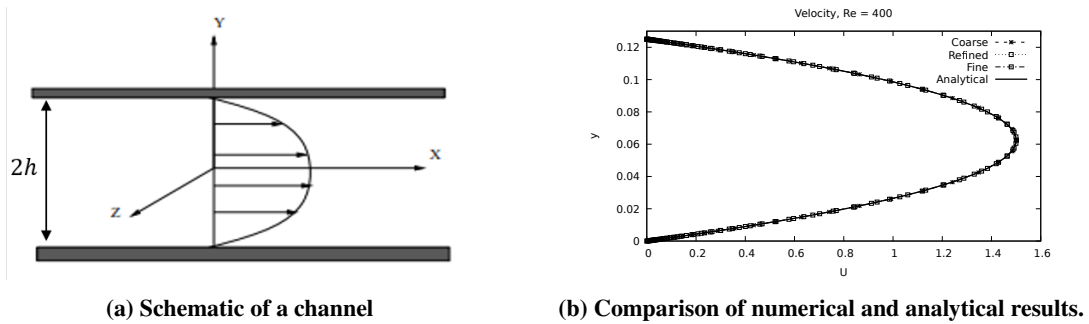
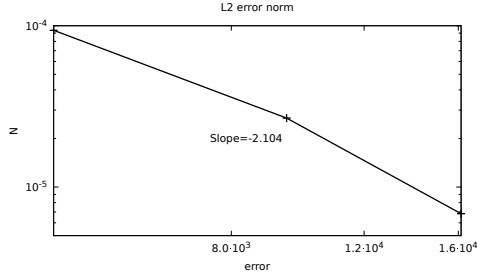
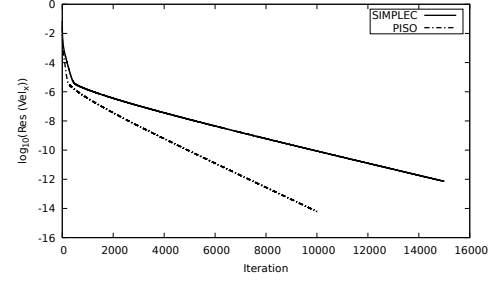


Fig. 1 Channel flow schematic (a) and velocity profile comparison between numerical and analytical solution at different grid resolutions.

Three different mesh resolutions are chosen and the numerical results are compared to the analytical solution (Fig. 1b). 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. 2a), as expected. Convergence history is also shown for the coarse grid with SIMPLEC and PISO algorithms (Fig. 2b). As expected, the PISO algorithm converges to the final solution faster than the SIMPLEC algorithm.



(a) L2 norm of the error



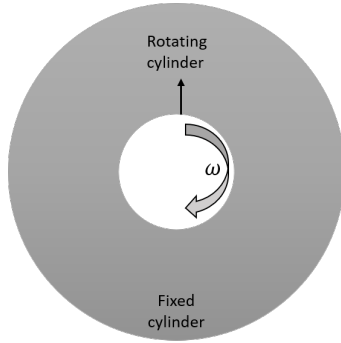
(b) Convergence history.

Fig. 2 Channel flow error norm (a) and convergence history (b).

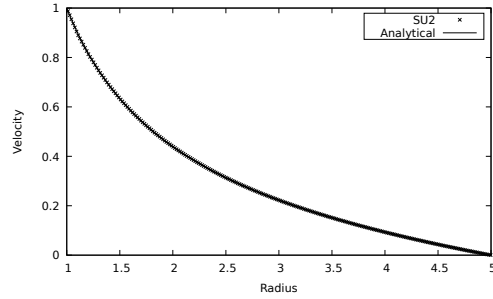
2. Couette flow

Couette flow can be described as the laminar flow between two concentric rotating cylinders. The schematic for Couette flow is shown in Fig. 3a with r_0 as the inner radius and r_1 as the outer radius. ω_1 and ω_2 refer to the angular velocities of the two cylinders. The analytical solution for the velocity as a function of radius r can be described as

$$u_\theta(r) = r_0\omega_0 \frac{r_1/r - r/r_1}{r_1/r_0 - r_0/r_1} + r_1\omega_1 \frac{r/r_0 - r_0/r}{r_1/r_0 - r_0/r_1}, \quad (24)$$



(a) Schematic of Couette flow.



(b) Comparison of numerical and analytical results.

Fig. 3 Couette flow.

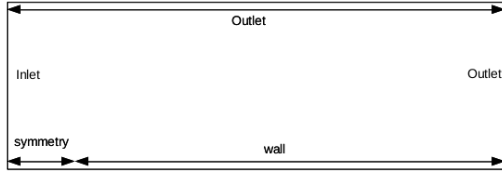
The simulation was carried out on a domain with $r_0 = 1$ and $r_1 = 5$ units. The outer wall is held fixed and the inner wall is rotating at an angular velocity $\omega_0 = 1$ rad/s. Fig. 3b shows the comparison of velocity along $y = 0$. We observe a very good agreement between the numerical and analytical solution

B. Validation

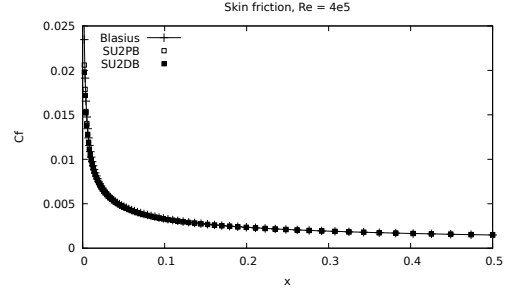
1. Laminar flow over a flat plate

The results from the laminar flow over a flat plate ($Re = 4e5$) is compared to the Blasius solution [45] at different locations (Figs. 5a 5b, 6a, 6b). The domain and the mesh used for the simulation is shown Fig. 4a. A uniform inflow is prescribed and a small inflow region with a symmetry boundary is used before the flat plate begins. Nodes are clustered at near the wall and stretched away from it in the normal direction and clustered around the interface between the symmetry and the wall region and stretched towards the outlet in the streamwise direction.

The results for the streamwise component (Figs. 5a and 5b) of the velocity shows excellent agreement between the numerical and Blasius solution for both the density-based and pressure-based methods. The comparison for the normal velocity component (Figs. 6a and 6b) is not as good but it can be seen that the pressure-based method does better than the density based method. Skin friction shows excellent agreement for both cases (Fig. 4b).

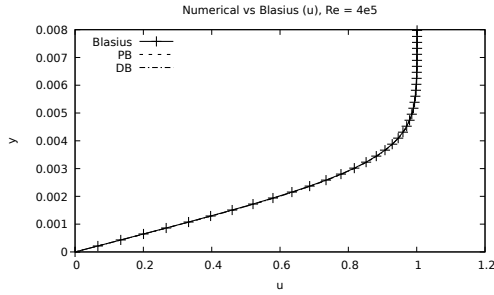


(a) Flat plate domain.

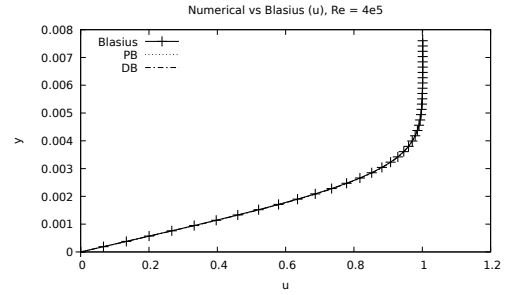


(b) Skin friction comparison.

Fig. 4 Flat plate domain (a) and skin friction comparison between numerical and Blasius solution (b).

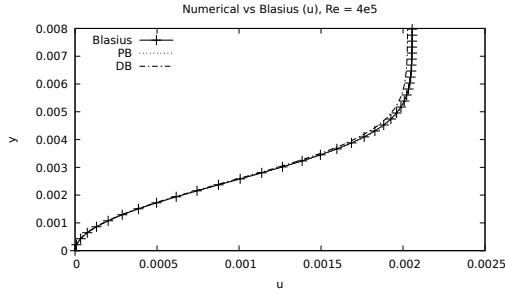


(a) $x = 0.45$

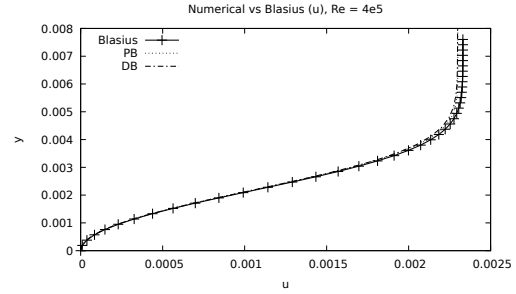


(b) $x = 0.35$.

Fig. 5 Comparison of x - component of velocity to the Blasius solution for laminar flow over a flatplate.



(a) $x = 0.45$



(b) $x = 0.35$.

Fig. 6 Comparison of y - component of velocity to the Blasius solution for laminar flow over a flatplate.

2. Laminar flow over a cylinder

A laminar flow over a cylinder is simulated for $Re = 40$ and the resulting drag coefficient is plotted as a function of grid refinement. The simulations converge to a fixed value of C_d as the grid is refined (Fig. 7a). Fig. 7b shows the pressure coefficient along the cylinder for different grid resolutions which also converge upon grid refinement. The low Reynolds number is chosen to ensure the flow remains laminar and steady. The results in Fig. 7a obtained match closely with those obtained from other reference solutions[46].

3. Inviscid flow over a NACA 0012 airfoil

Inviscid flow over a NACA 0012 hydrofoil at an angle of attack of 5° is simulated. This case is chosen to demonstrate the inviscid flow solver. The airfoil is placed in a rectangular channel and inclined at an angle of 5 degrees to the incoming flow. A flow tangency condition (i.e. free slip) is enforced on the surface of the airfoil. The boundary

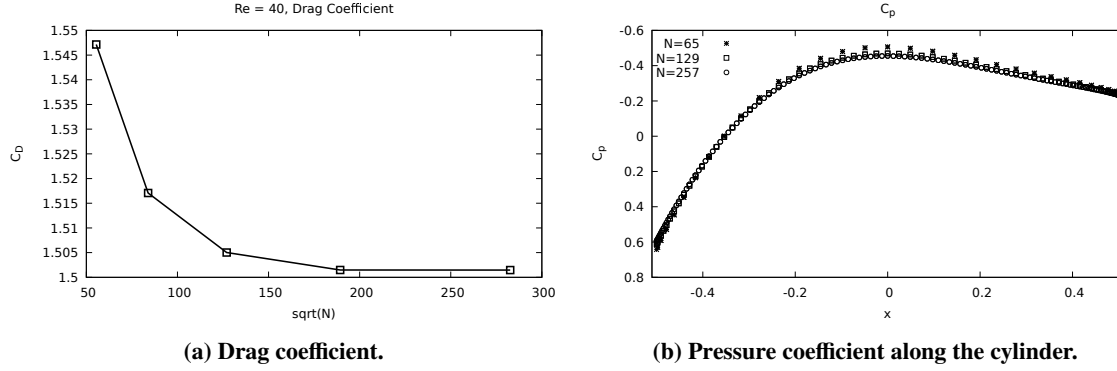


Fig. 7 Laminar flow over a cylinder at $Re = 40$.

conditions used on the channel are inlet, outlet and free slip walls. This configuration of boundary conditions is necessary to ensure a unique solution from the solver. The streamlines and the pressure contours are shown in Fig. 8a and the distribution of the pressure coefficient over the hydrofoil is shown in Fig. 8b and compared to the results from RFOIL[6], a wind energy industry standard airfoil design tool. We observe good agreement between the inviscid version of the new pressure based method in SU2 and RFOIL.

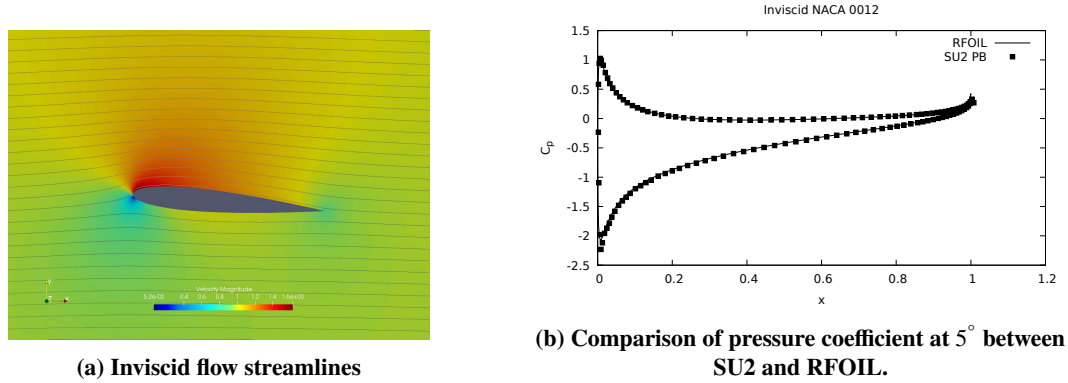


Fig. 8 Results from inviscid flow over NACA 0012.

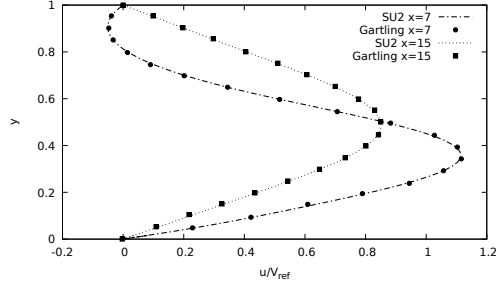
4. Laminar flow over a backward facing step

Flow over a backward facing step is a very popular validation problem and is commonly used to study the behavior of a numerical solver in separated flows. The domain consists of an inlet channel which expands into a larger channel across a step. The flow separates at the step and re-attaches downstream along the lower wall. Depending on the Reynolds number, a secondary separated region can also occur along the top wall. We simulate a laminar flow over a backward facing step at $Re = 800$ where flow is expected to separate along both the bottom and top walls. The numerical results compared to results from Gartling[47]. Step height is 0.5 units and the channel height is 1 unit.

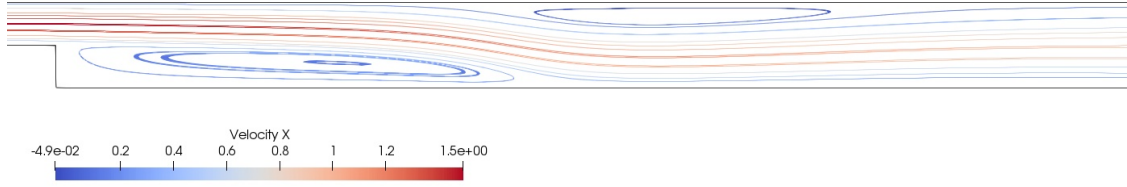
Fig. 9a compares the numerical and experimental results for the streamwise velocity component at two different locations. The velocity profile shows good agreement at both locations. The streamlines for a part of the domain are shown in 9b which clearly shows the two recirculating regions that are formed. The length of the recirculating zone along the lower wall is 5.81 units and along the upper wall is 5.69 units which match the results from [47].

5. Lid driven cavity

Flow within a lid-driven cavity is another classic validation problem in the CFD community. This case is chosen here to test the moving wall boundary condition. In this section, the results from a lid driven cavity is compared against



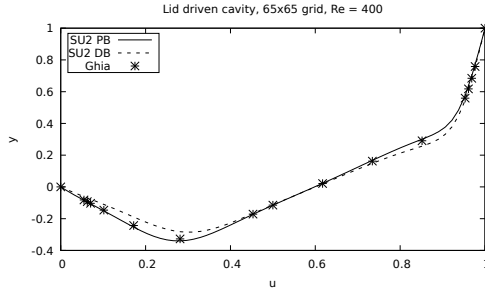
(a) Velocity at $x = 7$ and $x = 15$



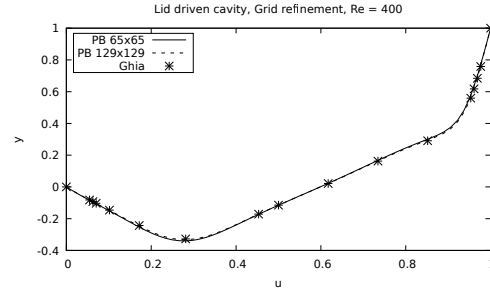
(b) Streamlines at $Re = 800$.

Fig. 9 Laminar flow over a backward facing step. Comparison of velocity profiles obtained from the numerical results with the literature at two different x -locations (a) and streamlines showing the two recirculating zones along the bottom and top walls (b).

benchmark results from Ghia[48]. The flow is fully laminar and at a $Re = 400$. The domain consists of a unit square, where all four sides are treated as solid walls. The top wall is moving at a constant unit velocity in the x -direction. Results from the density based (DB) version and the newly implemented pressure based (PB) version are shown in Fig. 10a and results from two successively fine grids are shown in Fig. 10b. On the coarser (65×65) grid, the PB method matches the reference solution more closely than the DB method. The difference between the reference solution and the PB solution improves upon grid refinement as expected.



(a) Comparison between density based and pressure based methods on a 65×65 grid



(b) Results from 65×65 and 129×129 grid with the pressure based method.

Fig. 10 Comparison of the velocity profile obtained from pressure based (PB) and density based (DB) methods in SU2 and reference solution[48] along the centerline($x = 0.5$) for lid driven cavity, $Re = 400$.

6. Turbulent flow over a flat plate

A turbulent flow over a flat plate is simulated at $Re = 5e6$ and the results compared to the standard 2D zero pressure gradient flat plate validation case from the NASA Turbulence Modeling resource[49]. The domain used is similar to the one shown in Fig. 4a, with more refinement in the wall normal direction to ensure adequate resolution of the turbulent boundary layer near the wall. The grids used are from the NASA turbulence modeling database and the coarsest grid has

an average $y^+ \approx 1.7$. Subsequent grids are refined ensuring proper resolution of the boundary layer near the wall. Fig. 11a shows the comparison of skin friction obtained from the current pressure based method with the one obtained from FUN3D[49]. Fig. 11b shows the drag coefficient of the flat plate obtained for different grid resolutions. We observe that the drag coefficient converges to a single value upon grid refinement.

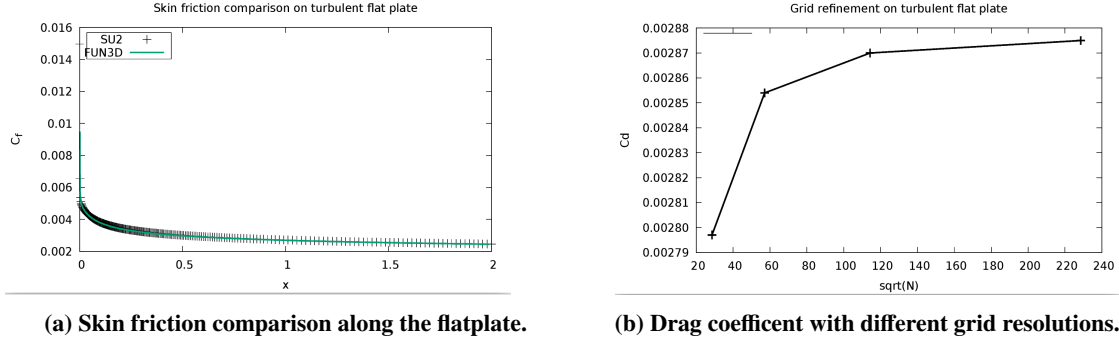


Fig. 11 Turbulent flow over a flat plate at $Re = 5e6$.

7. Turbulent flow over NACA0012 airfoil

Flow over airfoils are a very typical problem in most aerodynamic applications. Lift and drag polars of airfoils used in the different sections of a wind turbine blade are commonly used as an input to the lower fidelity tools based on Blade Element Momentum theory, lifting line theory, etc. which can then be used to analyze the performance of turbine blades. In this section, a fully turbulent flow over NACA0012 airfoil employing the SA and the SST turbulent models are compared with the experimental data[50] at a Re number of $6.0e6$ on a grid with approximately 14000 elements. The pressure-based method matches the experimental data very closely (Fig. 12) at all angles of attack and also captures the maximum lift angle.

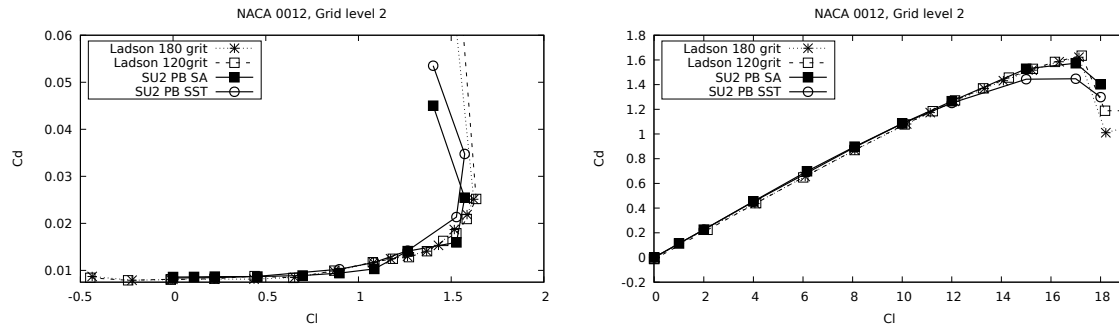


Fig. 12 Turbulent flow over NACA0012 airfoil. A comparison of the numerical solution with the experimental data for lift to drag ratio (left) and lift coefficient for various angle of attacks (right).

V. Applications

A. Boundary layer analysis for flow past vortex generators

In this section, some results from the flow past a vortex generator (VG) on a flat plate are presented. VGs are commonly used on wind turbines as a means to delay stall. To better understand the effect of a VG on turbulent boundary layers, a VG is placed on a flat plate under zero pressure gradient. The height of the VG is 10 mm and is placed such that the boundary layer thickness at the start of the VG is 45mm. The flat plate is 5m long and the leading edge of the VG is at 2.71m from the start of the flat plate. The VG is inclined at an angle of 19° to the incoming flow.

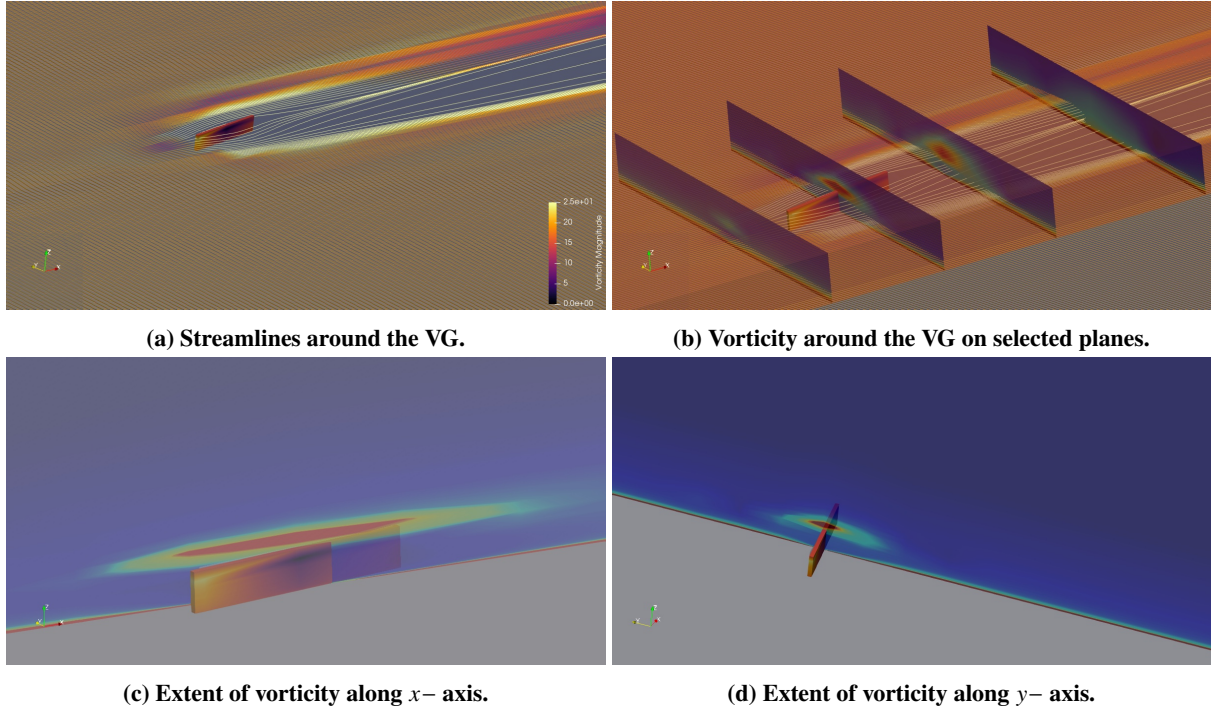


Fig. 13 Vorticity profiles in a turbulent boundary layer around the Vortex Generator.

Fig. 13a shows the streamlines around the VG and is colored by the Q criterion. Fig. 13b shows the strength of the vorticity in the boundary layer due to the presence of the VG at different sections ($x = 2.65$, $x = 2.71$, $x = 2.77$ and $x = 2.90$) and how it decays further away from the VG. Figs. 13c and 13d show the extent of the vortex induced by the VG in x - and y - directions respectively. It is clear that the effect of VG starts upstream from the leading edge of the VG and extends far downstream. A vortex is embedded in the turbulent boundary layer which decays gradually as the presence of the wall starts to dominate further away from the VG.

A more quantitative investigation into the boundary layer can be carried out by extracting the velocity, vorticity and other profiles along the boundary layer on different planes like in Fig. 14. From these profiles, it becomes clear that the effect of the VG on velocity begins from $x = 2.65$ on the $y = 0$ plane and extends up to $x = 3.10$. From Figs. 14a and 14b, we can see how a new shear layer is introduced into the boundary layer.

VI. Conclusions

The new pressure-based incompressible Navier-Stokes solver has been verified and validated for a variety of test cases. Based on the verification study, the order of convergence is shown to be second order. Comparisons to Blasius solution for flow over a flat plate and comparison with experimental data for flow over a backward facing step show good agreements. The numerical results obtained from the new pressure based solver are also compared with an wind energy industry standard tool, RFOIL and very good agreement is observed. SIMPLE and PISO algorithms are implemented for the pressure-velocity coupling. Two turbulence models, namely, SA and SST are coupled and also tested with the standard validation test cases. A typical application problem is also presented to demonstrate the use of the new open source solver for wind energy applications.

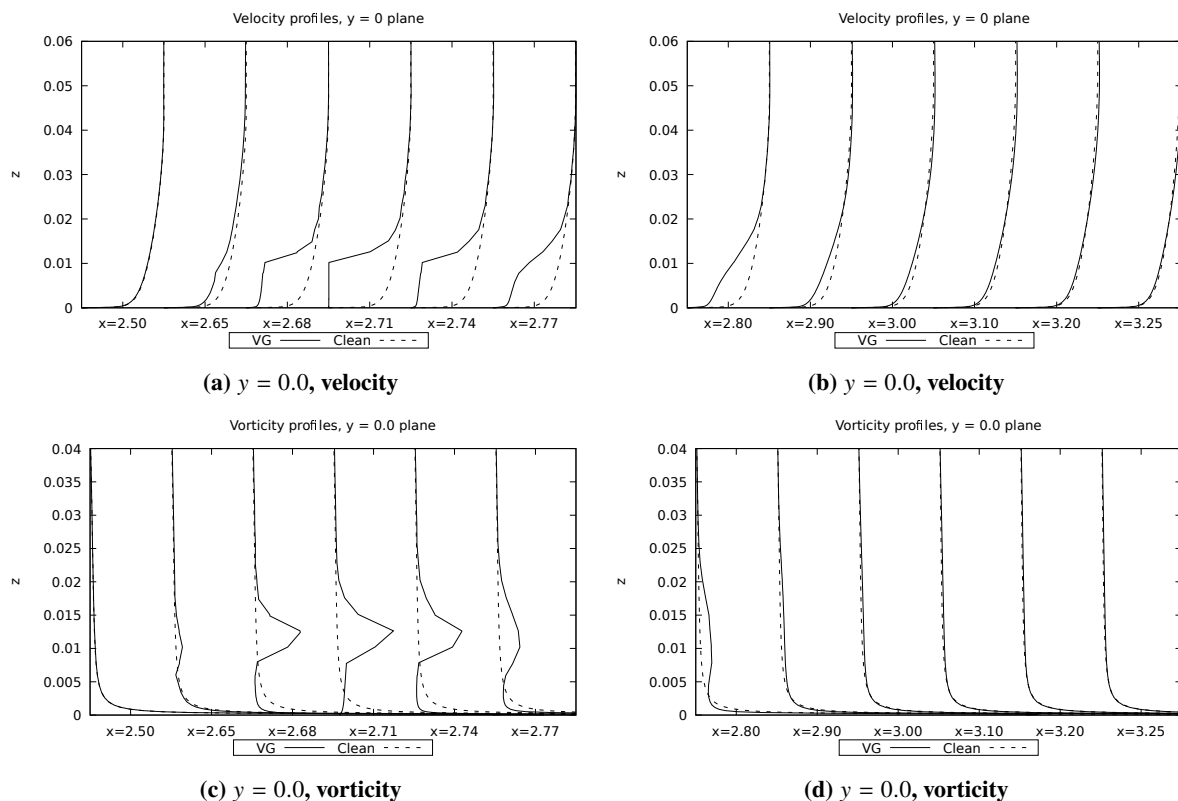


Fig. 14 Boundary layer profile comparison between clean and VG cases at different planes.

References

- [1] Manwell, J., McGowan, J., and L Rogers, A., *Wind Energy Explained: Theory, Design and Application, Second Edition*, 2006, Vol. 30. doi:10.1260/030952406778055054.
- [2] Schepers, J., "Engineering models in wind energy aerodynamics: Development, implementation and analysis using dedicated aerodynamic measurements," Ph.D. thesis, Delft University of Technology, 2012.
- [3] Katz, J., and Plotkin, A., *Low-Speed Aerodynamics*, 2nd ed., Cambridge Aerospace Series, Cambridge University Press, 2001. doi:10.1017/CBO9780511810329.
- [4] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.
- [5] Van Rooij, R., "Modification of the boundary layer calculation in RFOIL for improved airfoil stall prediction," , Sep 1996.
- [6] Ramanujam, G., Özdemir, H., and Hoeijmakers, H. W. M., "Improving airfoil drag prediction," *Journal of Aircraft*, Vol. 53, No. 6, 2016, pp. 1844–1852.
- [7] Ramanujam, G., and Özdemir, H., "Improving airfoil lift prediction," *35th Wind Energy Symposium*, 2017, p. 1999.
- [8] Vaithiyathanasamy, R., Özdemir, H., Bedon, G., and van Garrel, A., "A double wake model for interacting boundary layer methods," *2018 Wind Energy Symposium*, 2018, p. 0516.
- [9] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, Vol. 12, No. 6, 1998, pp. 620–631. doi:10.1063/1.168744.
- [10] Michelsen, J., "Basis3D - a Platform for Development of Multiblock PDE Solvers," Tech. Rep. AFM 92-05, Technical University of Denmark, 1992.
- [11] Michelsen, J., "Block structured Multigrid solution of 2D and 3D elliptic PDE's," Tech. Rep. AFM 94-06, Technical University of Denmark, 1994.

- [12] 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., "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," , No. January, 2013, pp. 1–60. doi:10.2514/6.2013-287.
- [13] Palacios, F., Padron, S., Tracey, B., Manosalvas, D. E., Aranake, A., Copeland, S. R., Variyar, A., Alonso, J. J., Lukaczyk, T. W., Lonkar, A. K., Naik, K. R., and Economon, T. D., "Stanford University Unstructured (SU2): Analysis and Design Technology for Turbulent Flows," , No. January, 2014, pp. 1–33. doi:10.2514/6.2014-0243.
- [14] Porté-Agel, F., Bastankhah, M., and Shamsoddin, S., "Wind-Turbine and Wind-Farm Flows: A Review," *Boundary-Layer Meteorology*, 2019. doi:10.1007/s10546-019-00473-0.
- [15] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th aerospace sciences meeting and exhibit*, 1992, p. 439.
- [16] Menter, F. R., "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.
- [17] Cakmakcioglu, S. C., Bas, O., and Kaynak, U., "A correlation-based algebraic transition model," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 232, No. 21, 2018, pp. 3915–3929.
- [18] Menter, F. R., Langtry, R., and Völker, S., "Transition Modelling for General Purpose CFD Codes," *Flow, Turbulence and Combustion*, Vol. 77, No. 1, 2006, pp. 277–303.
- [19] Kwak, D., and Kiris, C., "Success and Challenges in Incompressible Flow Simulation." AIAA, 2003.
- [20] Kwak, D., Kiris, C., and Kim, C. S., "Computational challenges of viscous incompressible flows," *Computers and Fluids*, Vol. 34, No. 3, 2005, pp. 283 – 299.
- [21] Ferziger, J. H., Perić, M., and Peric, M., *Computational Methods for Fluid Dynamics*, 3rd ed., Springer, 2002. doi: 10.1016/S0898-1221(03)90046-0.
- [22] Moukalled, F., Mangani, L., and Darwish, M., *Fluid Mechanics and Its Applications The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM® and Matlab®*, Springer, 2015.
- [23] Shyy, W., "Elements of Pressure-Based Computational Algorithms," *Advances in Heat Transfer*, Vol. 24, No. C, 1994, pp. 191–275.
- [24] Hirsch, C., *Numerical computation of internal and external flows*, Wiley, 1997.
- [25] Park, C. A., and Kwak, D., "On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows." AIAA, 1984.
- [26] Chorin, A. J., "A numerical method for solving incompressible viscous flow problems," *Journal of computational physics*, Vol. 2, No. 1, 1967, pp. 12–26.
- [27] Economon, T. D., "Simulation and adjoint-based design for variable density incompressible flows with heat transfer," *AIAA Journal*, 2019, pp. 1–13.
- [28] F.H.Harlow, and J.E.Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface," *Physics of FLuids*, Vol. 8, No. 12, 1965.
- [29] Patankar, S. V., and Spalding, D. B., "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, 1972.
- [30] Patankar, S., *Numerical heat transfer and fluid flow*, CRC press, 1980.
- [31] Rhie, C. M., and Chowt, W. L., "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation," *AIAA Journal*, Vol. 21, No. 11, 1983. doi:10.2514/3.8284.
- [32] Majumdar, S., "Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids," *Numerical Heat Transfer*, Vol. 13, No. 1, 1988, pp. 125–132. doi:10.1080/10407788808913607.
- [33] Choi, S. K., "Note on the use of momentum interpolation method for unsteady flows," *Numerical Heat Transfer Part a-Applications*, Vol. 36, No. August 1998, 1999, pp. 545–550.

- [34] Shen, W. Z., Michelsen, J. A., and Sorensen, J. N., "Improved Rhie-Chow interpolation for unsteady flow computations," *AIAA Journal*, Vol. 39, No. 12, 2012, pp. 2406–2409. doi:10.2514/3.15042.
- [35] Cubero, A., and Fueyo, N., "A compact momentum interpolation procedure for unsteady flows and relaxation," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 52, No. 6, 2007, pp. 471–493. doi:10.1080/10407790701563334.
- [36] Bartholomew, P., Denner, F., Abdol-Azis, M. H., Marquis, A., and van Wachem, B. G., "Unified formulation of the momentum-weighted interpolation for collocated variable arrangements," *Journal of Computational Physics*, Vol. 375, 2018, pp. 177–208.
- [37] Blazek, J., "Chapter 5 - Unstructured Finite Volume Schemes," *Computational Fluid Dynamics: Principles and Applications (Second Edition)*, edited by J. Blazek, Elsevier Science, Oxford, 2005, second edition ed., pp. 131 – 182.
- [38] Barth, T. J., "Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations," 1992.
- [39] Yu, B., Kawaguchi, Y., Tao, W. Q., and Ozoe, H., "Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 41, No. 1, 2002, pp. 85–94. doi:10.1080/104077902753385027.
- [40] Murthy, J. Y., and Mathur, S., "Periodic flow and heat transfer using unstructured meshes," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 6, 2002, pp. 659–677.
- [41] Xiao, H., Wang, J., Liu, Z., and Liu, W., "A consistent SIMPLE algorithm with extra explicit prediction — SIMPLEPC," *International Journal of Heat and Mass Transfer*, Vol. 120, 2018, pp. 1255–1265. doi:10.1016/j.ijheatmasstransfer.2017.12.119.
- [42] Yen, R.-H., and Liu, C.-H., "Enhancement of the SIMPLE algorithm by an additional explicit corrector step," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 24, No. 1, 1993, pp. 127–141. doi:10.1080/10407799308955885.
- [43] Sezai, I., "Implementation of boundary conditions in pressure-based finite volume methods on unstructured grids," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 72, No. 1, 2017, pp. 82–107. doi:10.1080/10407790.2017.1338077.
- [44] Darwish, M., Mangani, L., and Moukalled, F., "Implicit boundary conditions for coupled solvers," *Computers and Fluids*, Vol. 168, No. 2018, 2018, pp. 54–66. doi:10.1016/j.compfluid.2018.03.046.
- [45] Lyon, H., *Boundary Layer Theory*, Vol. 7, 2014. doi:10.1108/eb029898.
- [46] Gautier, R., Biau, D., and Lamballais, E., "A reference solution of the flow over a circular cylinder at $Re=40$," *Computers & Fluids*, Vol. 75, 2013, pp. 103 – 111. doi:https://doi.org/10.1016/j.compfluid.2012.12.017.
- [47] Gartling, D. K., "A test problem for outflow boundary conditions—flow over a backward-facing step," *International Journal for Numerical Methods in Fluids*, Vol. 11, No. 7, 1990, pp. 953–967. doi:10.1002/flid.1650110704.
- [48] Ghia, G., and Shin, "High-Re solutions for incompressible flow using Navier-Stokes equations and a multigrid method," *Journal of Computational Physics*, Vol. 48, 1982.
- [49] Rumsey, C., "NASA Turbulence Modeling Resource," <https://turbmodels.larc.nasa.gov/index.html>, 2019.
- [50] Ladson, C. L., "Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section," 1988.