OpenFOAM®
Journal

Volume 3 [Full Papers], Pages 177–199
ISSN: 2753-8168

Sensitivity Analysis of an Unsteady Flow Around a Pitching Airfoil

Javier Añez[1,2], Hakim Hamdani[1], Julien Reveillon[1,*], Benjamin Duret[1], and François-Xavier Demoulin[1]
Email address: `Julien.Reveillon@coria.fr`

[1]CORIA, University of Rouen Normandy

[2]R&D Division, Électricité de France (EDF)

DOI: https://doi.org/10.51560/ofj.v3.81
Results with version(s): OpenFOAM® v2112
Repository: https://github.com/JulienReveillon/OFJ-pitching-airfoil

Abstract. Various ways to control the loads and thus the output of the wind turbine by pitching the blades and by controlling the rotational speed already exist [1]. However, adverse pressure gradient leading to flow separation at the trailing edge, with relatively high angles of attack (AOA) due to blade-pitching motion, largely affects the airfoil aerodynamic performance. This work presents the results of a numerical study of the unsteady flow around a pitching `FFA-W3-301` airfoil at a Reynolds 1.6×10^6 using OpenFOAM®, as obtained by first performing $2D$ Unsteady Reynolds-Averaged Navier-Stokes (URANS) simulations whereby the flow characteristics are simulated by the shear stress transport (SST) $k - \omega$ model [2, 3] and Spalart-Allmaras (SA) [4]. The influence of various parameters on the numerical results is investigated, namely $y+$, computational grid resolution, dynamic mesh technique, time step and turbulence model. Integral aerodynamic forces and detailed flow patterns are compared with experimental measurements presented in the literature [5]. A range of angles of attack, including early stalls, are examined. For best-performing parameters, an adequate refinement close to the wall was imperative in order to match experiments, especially during the upstroke motion of the airfoil, while for the downstroke phase, some differences still appeared. The agreement was greatly improved by using a $2.5D$ hybrid RANS-LES approach with enhanced delayed-Detached Eddy Simulation (DES) capabilities [6].

1. Introduction

Modern large wind turbines may have a blade diameter greater than 100 m, and the cut-in and cut-out wind speeds are generally 5 m/s and 25 m/s, respectively. With these conditions, the reduced frequencies of a modern large wind turbine are mostly within the range in which dynamic stall appears, as reported in the literature [7]. Therefore, dynamic stall does occur on wind turbines. An airfoil with a pitching motion is a frequent geometry for studying this phenomenon. A pitching airfoil can stall at a higher angle of incidence than a static airfoil, according to numerous experimental investigations such as [5, 8], resulting in higher maximum values of lift and drag coefficients. However, at a lower angle of incidence than in the static airfoil configuration, the reattachment of the boundary layer is delayed, resulting in a hysteresis loop in the evolution of the aerodynamic coefficients [9].

The stall delay is greater when the pitch oscillating airfoil is combined with an oscillating freestream velocity, as reported in [10]. For instance, an accelerating freestream velocity during dynamic stall speeds up the vortex sheet expansion in the wake. When the static angle of incidence is exceeded, two main mechanisms are responsible for stall delay: (1) A delay in the boundary layer separation and (2) The initiation and growth of a Leading Edge Vortex (LEV) [11]. For a pitching airfoil, the magnitudes of the maximum drag and minimum moment decrease with the increase in the free-stream turbulence [7]. Considering the wind turbine aerodynamics, yawed wind (when the wind is not normal to the rotating plane of the turbine), wind shear, tower shadow, wake from the upstream turbine, and large atmospheric turbulence eddies all contribute to unsteady inflow conditions which can also lead to dynamic stall [7].

[*] Corresponding author

Received: 7 February 2022, Accepted: 26 June 2023, Published: 20 November 2023

Studies have reported that virtually all airfoils experience a fully developed dynamic stall. They concluded that the parameters affecting dynamic stall, in order of decreasing importance, are airfoil geometry (profile and chord length), frequency of pitching, pitching amplitude and Reynolds number [12]. It was found that increasing the reduced frequency k increases the incidence delay related to the boundary layer detachment. Therefore, the LEV initiates at a higher angle of incidence as the reduced frequency increases [9].

The integral forces and moments encounter huge fluctuations and large hysteresis during dynamic stall, resulting in a major rise in stresses, making the blades more susceptible to fatigue damage, i.e., quick mechanical failure. [8,13] discusses the main physical elements involved with dynamic stall. It is commonly acknowledged that simulating the above characteristics accurately and reliably using computational fluid dynamics (CFD) analysis is difficult, as the CFD findings might be highly reliant on the computational settings utilized. For example, the turbulence modeling approach is critical. The lift coefficient benefited from the Transition SST model, which proved to be more accurate compared to the original $SST\ k{-}omega$ model [10] or the $k-\omega\ DES$ approach [14] in a 2D study performed around a NACA 0012 pitching airfoil with both approaches for the same calculation. Nonetheless, the $SST\ k-\omega\ DES$ presented a better agreement with the experimental data, especially for the down-stroke phase compared with the $Transition\ SST$ model. Another important parameter was highlighted by [15, 16] under static airfoil conditions related to the trailing edge profile shape, which can have a significant impact on the flow field across the airfoil, thereby affecting its overall aerodynamic performance. Generally, there are two ways of geometrically modeling the trailing edge, by either (1) extending or (2) truncating the trailing edge. However, the two approaches were tested in dynamic stall conditions on the NACA 0012 airfoil profile and both gave a good overall agreement with experimental data [17].

Some researchers [7,18] studied the effect of free-stream turbulence on the aerodynamic characteristics of a pitching airfoil. They found that results for the cases with relatively small frequencies were in good agreement with experiments, while greater deviations were observed for higher frequencies. They concluded that the dynamic flow characteristics are dependent on the problem details. With regard to vortex development, it was also found that the leading edge vortex at a higher frequency is much more violent than in cases with lower k, which most likely requires a greater computational resolution. In this article, a relative medium reduced frequency $k = 0.07$ is employed, for which numerical results can be compared with the wind tunnel test data obtained by [5].

Some researchers have previously worked with pitching airfoils in OpenFOAM® before. For example, [19] based on sensitivity analysis comprising the grid, numerical schemes, turbulence models concluded the need of fully resolved wall to accurately predict the hysteresis lift loop in pitching airfoils. Even though turbulence models evaluation was done under static conditions. Mesh and turbulence model sensitivity are equally important when it comes to pitched blade near-stall region as demonstrated for oscillating airfoils [20]. $y^+ \leq 1$ in OpenFOAM® was required to ensure accuracy predictions. Some years later, same authors [21] validated against experimental data using OpenFOAM® the pitching motion of an $2D$ $S809$ airfoil near-stall region to better study fatigue loads on pitched blades. The goal of the present paper is to perform a validation study of an unsteady flow around a pitching $2D$ and $2.5D$ `FFA-W3-301` airfoil beyond stall region at a Reynolds $1.6 * 10^6$ using OpenFOAM®, in which a sensitivity analysis including $y+$, computational grid resolution, time step size, mesh generation and turbulence modeling is carried out.

From the point of view of the OpenFoam community, this work proposes to test in a pragmatic way the influence of different meshing methods `blockMesh`, `SHM`, `oversetMesh`, different mesh refinement `y+`, different turbulence models and different time steps strategies and, finally, the effective difference between 2D and 2.5D calculations. This will allow anyone wishing to do external aerodynamics calculations to have a first idea of the accuracy of each method. Consequently, by knowing the impact of each of these parameters, this study will enable the person to save time on the implementation of their numerical configurations.

The paper is organized as follows. Section 1 illustrates the physical/numerical modeling applied to wind turbine airfoils, including details of the turbulence models used and the well-known aerodynamic dimensionless numbers. Section 2 describes the test case setup. The procedure of a C-type non-symmetrical airfoil mesh generation will be explained, followed by the initial and boundary conditions along with the numerical setup. Section 3 is dedicated to the numerical/experimental comparison both for static and dynamic stall conditions. Firstly, numerical/experimental comparisons are made in $2D$ simulations for validation purposes by varying $y+$, computational grid resolution, time step size and turbulence model. Secondly, $2.5D$ numerical simulations are compared with experiments using the $SST\ k-\omega\ DES$. Final conclusions are given in section 4.

2. Physical and numerical modeling

A very common physical phenomenon appears in dynamic stall studies while drastically varying the angle of attack, e.g. in blade-pitching airfoils is the well-known flow separation under an adverse pressure gradient. A brief description is introduced through the boundary layer separation. The behavior of a boundary layer in a positive pressure gradient (i.e., pressure increasing with distance downstream) may be considered with Fig. 1, which shows a length of surface of an airfoil beyond the point of maximum thickness. Consider the small element of fluid marked in the figure as $ABCD$, in which an increase in the pressure, $\frac{\partial p}{\partial x}\delta x$, is highlighted from faces AC to BD. Thus, the net pressure force on the element tends to delay the flow, therefore slowing the element as it progresses downstream [22]. At the separation point indicated on the airfoil surface, the velocity gradient $\left(\frac{\partial u}{\partial y}\right)_w$ becomes zero. Then it is evident that the boundary layer must thicken rapidly in order to satisfy continuity within the boundary layer. Downstream of separation point, the flow adjacent to the surface follows an upstream direction so that a circulatory motion exists and, at the separation point, the original boundary layer separates from the surface. The presence of a positive pressure gradient causes the flow to separate considerably more quickly than if it were only turbulent [22]. As a result, the separated laminar boundary layer may undergo a turbulence transition with rapid thickening characteristics. As stated in [22], a bubble of fluid is trapped under the separated shear layer between the separation and reattachment sites.

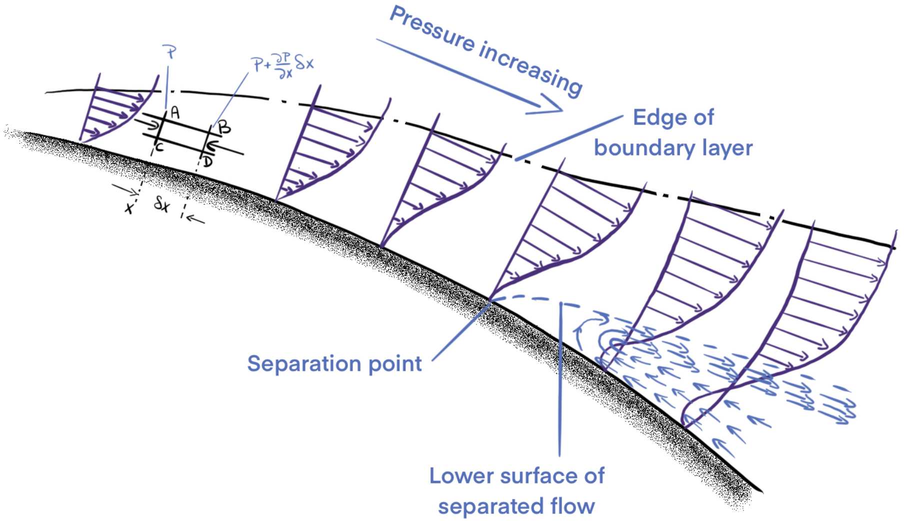

Pressure increasing
P
$P+\frac{\partial P}{\partial x}\delta x$
A
B
C
D
x
δx
Edge of boundary layer
Separation point
Lower surface of separated flow

Figure 1. Boundary layer separation from [22].

2.1. **Governing equations.** The equations of motion for real, viscous Newtonian fluids such as water and air are the Navier-Stokes equations. The governing equations that are solved are those for an incompressible and isothermal (low Mach number flow) single-phase fluid. Thus, the Navier-Stokes continuity and momentum equations are as follows:

$$\begin{gathered}\boldsymbol{\nabla}\cdot\boldsymbol{U}=0 \quad , \\ \frac{\partial \boldsymbol{U}}{\partial t}+\boldsymbol{\nabla}\cdot(\boldsymbol{U}\otimes\boldsymbol{U})=-\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_t+\boldsymbol{f}_b \quad ,\end{gathered} \tag{1}$$

where $\boldsymbol{U}$ is the velocity vector, the kinematic pressure $p = P/\rho$ is defined as the absolute pressure divided by the fluid density ρ and τ_t is the stress tensor decomposed in the following form: $\boldsymbol{\tau}_t = 2(\nu)\boldsymbol{D}$, in which $\boldsymbol{D}$ is proportional to the rate of deformation tensor $\boldsymbol{D} = 0.5[\boldsymbol{\nabla}\boldsymbol{U} + (\boldsymbol{\nabla}\boldsymbol{U})^T]$, and finally $\boldsymbol{f}_b$ is the body forces per unit of mass. The above equations are solved directly in case of laminar flow, however in this study the Reynolds number is about $\approx 10^6$, so in order to include the effect of turbulence on the flow, the velocity field is studied in terms of means and fluctuating values based on the Reynolds formalism as follows:

$$
\begin{aligned}
&\boldsymbol{\nabla}\cdot\bar{\boldsymbol{U}} = 0 \quad , \\
&\frac{\partial\bar{\boldsymbol{U}}}{\partial t} + \boldsymbol{\nabla}\cdot(\bar{\boldsymbol{U}}\otimes\bar{\boldsymbol{U}}) = -\boldsymbol{\nabla}\bar{p} + \boldsymbol{\nabla}\cdot\left[\bar{\nu}(\boldsymbol{\nabla}\bar{\boldsymbol{U}} + \boldsymbol{\nabla}\bar{\boldsymbol{U}}^t)\right] + \bar{\boldsymbol{f}}_b - \boldsymbol{\nabla}\cdot R_U \quad ,
\end{aligned}
\tag{2}
$$

The first term that arrives from the averaging procedure is the so-called Reynolds stress tensor $\boldsymbol{R}_U$. Following Boussinesq proposal it can be arranged into: $\boldsymbol{R}_U = \bar{\nu}_t(\boldsymbol{\nabla}\bar{\boldsymbol{U}} + \boldsymbol{\nabla}\bar{\boldsymbol{U}}^t)$. The primary issue is simple: there are 2 equations and 3 variables, namely $\boldsymbol{U}$, p, and ν_t determined through a turbulence model, which is necessary. Generally, it is based on deriving additional transport equations from the Navier-Stokes equations for quantities such as the Reynolds stress tensor, turbulent kinetic energy, and viscous dissipation rate, explained below. Since in this paper, a turbulence comparison analysis is conducted, a brief description of each turbulence model used is presented below, along with its intrinsic range of applicability.

2.1.1. *Spalart-Allmaras Turbulence model.* The Spalart-Allmaras (S-A) turbulence model [23] has been widely used and has proven to be numerically well behaved in most cases. There are, however, situations of under-resolved grids and unphysical transients where discretization of the model can lead to undesired results. Undershoots at the edge of boundary layers and wakes is one such situation [4]. The baseline of the model is made by solving a differential equation for the working variable, namely $\tilde{\nu}$ defined as follows:

$$
\begin{aligned}
\frac{D\tilde{\nu}}{Dt} &= P - D + T\frac{1}{\sigma}\left\{\boldsymbol{\nabla}\cdot[(\tilde{\nu}+\nu)\boldsymbol{\nabla}\tilde{\nu}] + c_{b2}(\boldsymbol{\nabla}\tilde{\nu})^2\right\} \quad , \\
\nu_t &= \tilde{\nu} f_{v1} \quad , \\
f_{v1} &= \frac{\chi^3}{\chi^3 + (c_{v1})^3} \quad , \\
\chi &\approx \frac{\tilde{\nu}}{\nu} \quad ,
\end{aligned}
\tag{3}
$$

where P, D, and T are the production, wall destruction and trip terms respectively [4]. For more mathematical details, the SA turbulence model is documented with various modifications on the NASA Langley Turbulence Modeling Resource website [24]. The original SA references prescribed a zero Neumann condition for inviscid walls, but it is now recommended to use a zero Dirichlet boundary condition to treat inviscid walls, also referred to as slip walls. The rationale for this change is re-examination of the blocking effect of inviscid walls on the development of Reynolds stresses.

2.1.2. *$k-\omega$ SST Turbulence model.* The standard $k-\omega$ SST is a mix between the $k-\epsilon$ and the $k-\omega$ models. The former is used outside the boundary layer, and the latter inside the boundary layer [2]. The advantage of using different models for different flow regions is that each model has its own region of validated applicability. The transport equation for ϵ is rewritten as an equation for ω. Thus, solving for a pair of equations and switching between the two equations is done by some blending functions. The Menter SST $k-\omega$ has been developed from the Wilcox $k-\omega$ [25] to specify numerical results within the boundary layer. Apart from this, the $k-\omega$ has been recently equipped with additional features over its two base models. Firstly, it has a shear stress limiter (reducing ν_t), which ensures that the turbulent shear stress does not become too large in adverse/positive pressure gradient regions. Secondly, a production limiter is applied on the production term in the k equation in order to prevent the build-up of turbulence in stagnant regions [2]. Finally, the two equations of the model can be written by:

$$
\begin{aligned}
&\frac{\partial k}{\partial t} + \boldsymbol{U}\cdot\boldsymbol{\nabla}k = \boldsymbol{\nabla}\cdot\left[(\nu + \sigma_k\nu_t)\,\boldsymbol{\nabla}k\right] + G - \beta^*\omega k + S_k \quad , \\
&\frac{\partial \omega}{\partial t} + \boldsymbol{U}\cdot\boldsymbol{\nabla}\omega = \boldsymbol{\nabla}\cdot\left[(\nu + \sigma_\omega\nu_t)\,\boldsymbol{\nabla}\omega\right] + \frac{\gamma G}{\nu} - \beta\omega^2 + S_\omega \quad , \\
&\qquad\qquad - 2\,(F_1 - 1)\,\frac{1}{\omega}\frac{1}{\sigma_{\omega,2}}\boldsymbol{\nabla}k\cdot\boldsymbol{\nabla}\omega \quad ,
\end{aligned}
\tag{4}
$$

where S_k and S_ω are the rate of production of k and ω, respectively [26]. F_1 is a blending function that varies from unity at the wall to zero outside wall boundary layers. Each coefficient of the Menter model $(\phi = \sigma_k, \sigma_\omega, \beta, \gamma)$ is evaluated from the following equation:

$$
\phi = \phi_1 F_1 + \phi_2\,(1 - F_1) \tag{5}
$$

where the subscript $()_1$ (in ϕ) is related to the adjusted $k-\omega$ model and the subscript $()_2$ is connected with the standard $k-\varepsilon$. Constant values of $\sigma_{k,1}$, $\sigma_{k,2}$, $\sigma_{\omega,1}$, $\sigma_{\omega,2}$, β_1, β_2, β^* are respectively 1.18, 1.0, 2.0, 1.17, 0.075, 0.0828 and 0.09.

2.1.3. *$k-\omega$ SST-SAS Turbulence model.* The *SAS* approach represents a different class of *URANS* models. Unlike conventional *RANS* formulations, the *SAS* adjusts the turbulence length scale to the local flow inhomogeneities, as reported in [27]. As a measure of the local flow length scale, a classic boundary layer length scale introduced by Von Karman is generalized for 3D flows. The resulting model remains a *RANS* model, as it delivers proper *RANS* solutions for stationary flows and maintains these solutions through grid refinement. On the other hand, for flows with transient instabilities such as those in massive separation zones, the model reduces its eddy viscosity according to the locally resolved vortex size represented by the Von Karman length scale. The governing equations of the $SST-SAS$ model differ from those of the *SST* model (eqs. 4) by the additional *SAS* source term Q_{SAS} in the transport equation for the turbulent eddy frequency ω. This term Q_{SAS} is defined as follows:

$$\begin{aligned} Q_{SAS} &= max\left[\rho\lambda_2 k S^2\left(\frac{L}{L_{\nu K}}\right)^2 - C\frac{2\rho k}{\sigma}\max\left(\frac{|\nabla\omega|^2}{\omega^2},\frac{|\nabla k|^2}{k^2}\right),0\right] \quad , \\ L &= \frac{\sqrt{k}}{\left(C_\nu\right)^{1/4}\omega} \quad , \\ L_{\nu k} &= \frac{\kappa U'(y)}{U''(y)} \quad , \end{aligned} \tag{6}$$

where the model parameters are $\lambda_2 = 3.51$, $\sigma = 2/3$, and $C = 2$. Noticeable is the quadratic length scale ratio, $\left(\frac{L}{L_{\nu K}}\right)^2$ in Eqn. 6 above: the value of L in the *SAS* is the length scale of the modeled turbulence, and $L_{\nu K}$ the Von Karman length scale, which comes from a generalization of the classic boundary layer definition, $\frac{\kappa U'(y)}{U''(y)}$ [27]. The first velocity derivative $U'(y)$ is represented here by $S = \sqrt{2S_{ij}S_{ij}}$, which is a scalar invariant of the strain rate tensor. The second velocity derivative $U''(y)$ is represented by the magnitude of the velocity Laplacian $|\nabla^2 U| = \sqrt{(\nabla^2 U)^2 + (\nabla^2 V)^2 + (\nabla^2 W)^2}$.

2.1.4. *$k-\omega$ SST transition Turbulence model.* Even though significant improvements have been made in the development of reliable turbulence models in the past decades, the important effect of the laminar-turbulent transition has only recently been included in today's engineering CFD simulations. In aerodynamic flows, transition is typically a result of a flow instability (Tollmien–Schlichting waves or cross-flow instability), where the exponential growth finally leads to turbulence due to a nonlinear breakdown [28]. The bypass transition [29, 30] imposed on the boundary layer by high levels of turbulence in the free stream flowing from upstream blade rows is the main transition mechanism in turbomachinery applications. Separation-induced transition [31] is another major transition mechanism, in which a laminar boundary layer separates under the influence of a pressure gradient, and transition occurs within the separated shear layer. Finally, an already turbulent boundary layer can be relaminarized under a strong favorable pressure gradient [32]. The standard *RANS* model typically suffers from the close interaction of the transition capability and the viscous sublayer modeling, which prevents an independent calibration of the two phenomena [33]. In addition, low-Re models can at best be expected to simulate bypass transition, which is dominated by diffusion effects.

A solution proposed by F. Menter [34] was a correlation-based transition model, built on transport equations, using only local variables. The concept is termed Local Correlation-based Transition Model (*LCTM*). The model consists of two components. The first is a generic framework comprised of two transport equations that connect the CFD code to experimental correlations. The transition model solves a transport equation for intermittency, γ, which is used to initiate the transition, and the transitional momentum thickness Reynolds number, Re_θ, which allows to capture the non-local effect of free-stream turbulence intensity and pressure gradient at the boundary layer edge.

$$\begin{aligned}\frac{\partial(\rho\gamma)}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{U}_i\gamma) &= P_\gamma - E_\gamma + \boldsymbol{\nabla}\cdot\left[\left(\mu+\frac{\mu_t}{\sigma_f}\right)\boldsymbol{\nabla}\gamma\right] \quad , \\ \frac{\partial(\rho\overline{Re}_{\theta t})}{\partial t} + \boldsymbol{\nabla}\cdot\left[\rho \boldsymbol{U}_i\overline{Re}_{\theta t}\right] &= P_{\theta t} + \boldsymbol{\nabla}\cdot\left[\sigma_{\theta t}(\mu+\mu_t)\boldsymbol{\nabla}\overline{Re}_{\theta t}\right] \quad .\end{aligned} \tag{7}$$

The production term P_γ, is designed to be equal to zero in the laminar boundary layer upstream of transition and active wherever the local strain-rate Reynolds number ($Re_\nu = \frac{\rho y^2 S}{\mu}$, where μ is the dynamic viscosity, ρ is the density, y is the distance from the nearest wall and S is the absolute value of the strain rate) exceeds the local transition onset criteria. Outside the boundary layer, the source term $P_{\theta t}$ is designed to force the transported scalar $\overline{Re}_{\theta t}$ to match the local value of $Re_{\theta t}$ calculated from an empirical correlation. Finally, the boundary condition for $\overline{Re}_{\theta t}$ at a wall is zero flux. The boundary condition for $\overline{Re}_{\theta t}$ at an inlet is calculated from the empirical correlation based on the inlet turbulence intensity. More details can be found in [28].

2.2. **Dimensionless numbers.** Four non-dimensional numbers allow one to compare two aerodynamic bodies of different size, shape, and orientation, in different conditions of flow, to one another. These numbers are the representations of the lift, drag, pitching moment and pressure around the body. They are detailed below.

For each angle of attack, these three variables are computed:

- The lift coefficient C_l is the ratio of the lift force L, the aerodynamic resultant force which acts perpendicular to the free stream direction, to the force produced by the dynamic pressure $1/2\rho U_\infty^2$ times the airfoil area A:
$$C_l = \frac{L}{1/2\rho U_\infty^2 A} \tag{8}$$
- The drag coefficient C_d is the ratio of the drag force D, the aerodynamic resultant force which acts parallel to the free stream direction, to the force produced by the dynamic pressure times the airfoil area:
$$C_d = \frac{D}{1/2\rho U_\infty^2 A} \tag{9}$$
- The moment coefficient C_m is the ratio between the pitching moment M, which is equal to the lift force multiplied by the moment arm between the quarter chord and the center of pressure, to the force produced by the dynamic pressure times the airfoil area and the chord length c:
$$C_m = \frac{M}{1/2\rho U_\infty^2 Ac} \tag{10}$$

In addition to these three variables, for a few values of angle of attack, the pressure coefficient C_p is computed along the airfoil profile. This dimensionless number is defined as the ratio of the wall pressure relative to the free stream pressure, to the dynamic pressure:

$$C_p = \frac{p - p_\infty}{1/2U_\infty^2} \tag{11}$$

The above-mentioned coefficients, C_l, C_d, and C_m, are functions of α, and Re for wind turbines. α is the angle of attack defined as the angle between the chord-line and the free stream direction. Re is the Reynolds number based on the chord i.e., $\frac{cV_\infty}{\nu}$. Experimental observations showed that C_l increases linearly with α, with an approximated slope of $2\pi/rad$, until a certain value of α, where a maximum value of C_l is reached. Thereafter, the airfoil is said to stall and C_l decreases in a very geometrically dependent manner. On the other hand, the Re dependency is related to the point on the airfoil where the boundary layer transition from laminar to turbulent flow occurs. The way an airfoil stalls is highly dependent on the geometry. Thin airfoils with a sharp nose i.e., high curvature around the leading edge, tend to stall more abruptly than thick airfoils [1]. Normally, there are two ways in which the boundary layer separates from the upper side of the airfoil. If the separation starts at the trailing edge of the airfoil and increases slowly with increasing angle of attack, a **soft stall** is observed, but if the separation starts at the leading edge of the airfoil, the entire boundary layer may separate almost simultaneously with a dramatic loss of lift [1].

3. Test-case set-up

3.1. **Experimental set-up.** The experimental measurements were taken from the literature [5]. Here, the experimental set-up used in the wind tunnel is briefly described. A *VELUX* closed return wind tunnel with an open test section, whose cross section is $7.5 \times 7.5\ m$. A length of 10.5 m was used, see Fig. 2 below. The cross section of the jet blowing into the test section is $3.4 \times 3.4\ m$. The maximum flow velocity is 45 m/s. The airfoil section with a span of 1.9 m and a chord of 0.6 m was mounted 1.7 m from the tunnel floor and 3.2 m from the nozzle outlet. End-plates were fixed to the stand at the ends of the airfoil section to limit $3D$ effects. Three Pitot tubes, as illustrated in the diagram 2 below, monitored static and total pressure at various positions in the test portion. These Pitot tubes were used to capture the wind tunnel reference pressures and to estimate the turbulence level and wind tunnel flow stability, with a measured turbulent intensity of 1 % at the test section intake. Aerodynamic properties were measured at $Re = 1.6 * 10^6$.

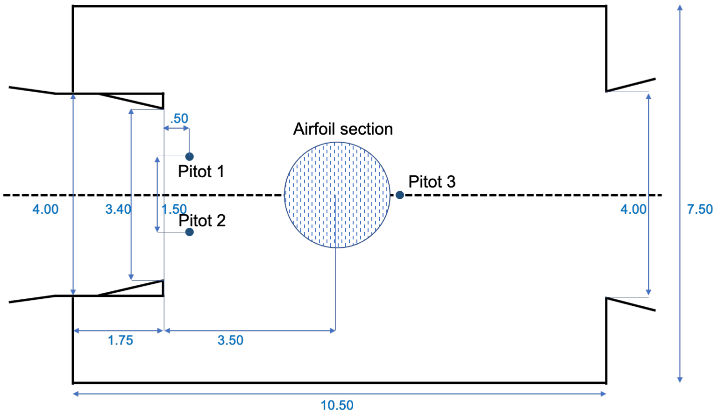


Figure 2. The wind tunnel test section with the test stand, [5].

[5] made dynamic inflow measurements and quasi-steady measurements at constantly varying angles of attack. In the center line region of the airfoil sections, 62 pressure taps of 0.5 mm inner diameter were installed. To reduce interference from upstream taps, the taps were staggered along the chord at the model's center line. The airfoil section was pitched in a harmonic motion at varied reduced frequencies up to $k = 0.15$ and amplitudes between $\pm 2° < A < \pm 5°$ with the pitch axis placed at $x/c = 0.40$ to generate dynamic inflow.

3.2. **FFA-W3-301 mesh generation procedure.** The computational domain is a two-dimensional $C-type$ domain whose radius and downstream length are 6 and 20 times the chord length, c, respectively (see Fig. 3). The outer computational domain for the $FFA-W3-301$ profile was adapted based on previous research [17] (Lines blue and red in Fig. 3 below). These dimensions were chosen to reduce the computational demand exhibited in previous $NACA$ 0015 numerical simulations. The distance from the domain inlet to the airfoil is $6c$ (similar to the experimental set-up) to avoid any nonphysical influence of the inlet boundary on the induction field upstream of the airfoil. The length behind the airfoil was chosen to be $20c$, which is larger than the length characterizing the wind tunnel test section ($\approx 8c$), in order to allow for the full development of the wake flow.

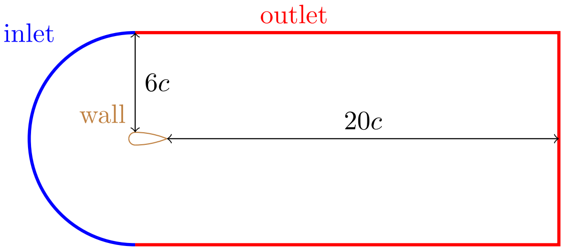


Figure 3. Schematic view of the C-type domain for `FFA-W3-301` airfoil.

One aspect is the rotation of the `FFA-W3-301` profile, thus the mesh must be decomposed into a static and a rotating part. In the rotating part, to enhance the mesh quality it is possible to use a solution based on the mesh strategy for a cylinder. It consists in putting a rectangular parallelepiped at the center of the cylinder. In this case, the parallelepiped-like shape is put after the trailing edge, as shown in Fig. 4 below.

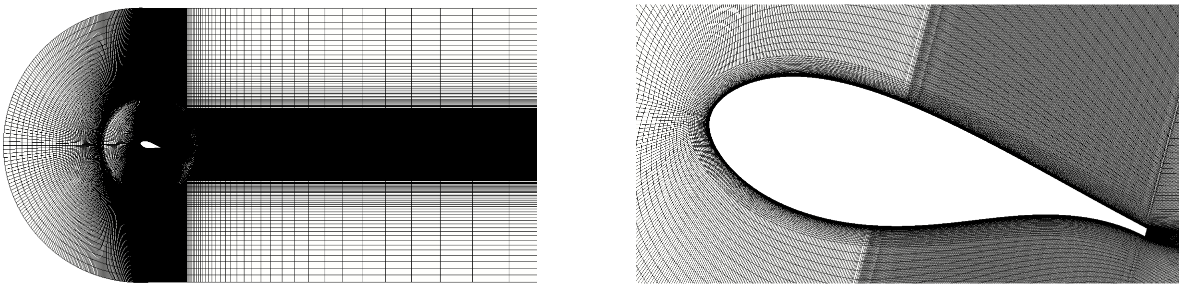


Figure 4. `OpenFOAM` mesh. On the left the $C - type$ mesh, and on the right a zoomed view of the `FFA-W3-301` airfoil profile.

The additional refinements observed on both images (left and right of the figure) are due to the structure-like mesh building block mechanism inherent to the `blockMesh` utility in `OpenFOAM`. On the left of Fig. 4, this over-refinement is conglomerated near the rotating part of the mesh (close to the airfoil) and in the wake region (behind the airfoil). On the right of Fig. 4, the refinement is better viewed close to the airfoil wall, in order to produce a refinement good enough to ensure $y+$ values around 10. To complete the mesh, a layer is added around the `FFA-W3-301` profile close to the wall. This allows up and down blocks before and after the trailing edge to have their right edges connected. Moreover, these layers can be used to control the mesh refinement near the airfoil. To construct this layer, a Python script is used to generate the layer coordinates from the airfoil profile coordinates. Figure 5 on the left shows the enlargement of the airfoil profile. For a point on the profile, the tangent is determined with a current point (p_i) and a previous one (p_{i-1}) as represented on Fig. 5 on the right.

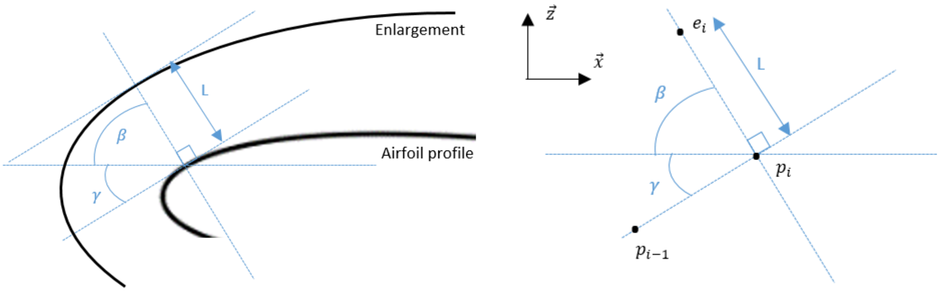


Figure 5. Layer added to the airfoil profile (left). Enlargement point associated to a point on the profile (right).

The angles λ , and β come from the mathematical description below, eq. 12. The enlargement has a length L in the tangent perpendicular direction. Thus the coordinates of the point of the enlargement associated to the point on the profile are computed as follows:

$$\begin{aligned} \gamma &= arctan\left(\frac{z_i - z_{i-1}}{x_i - x_{i-1}}\right) \quad , \\ \beta &= \frac{\pi}{2} - \gamma \quad . \end{aligned} \tag{12}$$

This process is repeated for each point on the upper and, analogously, on the lower part of the profile. The mesh smoothed transition (normal to the wall) is made by computing the number of cells (N), the grading (Q) and the length (L) on each edge. These three variables are written in the Python script. The number of cells and the grading are explicitly written as well, hence the length is deduced from the starting and ending point coordinates. Other mesh values are also linked to these values such as the size of the starting and ending cells (Δl_1 and Δl_N, respectively) and the cell-to-cell expansion ratio (q). The following lines give first the definition of the grading, the cell size evolution along the edge Δl_n, and finally the length (L) on each edge.

$$\begin{aligned} Q &= \frac{\Delta l_N}{\Delta l_1} \quad , \\ \Delta l_n &= \Delta l_1 . q^{n-1} \quad , \\ Q &= q^{N-1} \quad , \\ L &= \Delta l_1 \left(\frac{1 - q^N}{1 - q} \right) \quad . \end{aligned} \tag{13}$$

With these relations, it is possible to control the smoothness of the mesh and implement the following two functions:

$$\begin{aligned} q &= \frac{\frac{\Delta l_1}{L} - 1}{Q \frac{\Delta l_1}{L} - 1} \quad , \\ N &= \text{floor} \left(\frac{ln(Q)}{ln(q)} + 1 \right) \quad . \end{aligned} \tag{14}$$

To find the value of q, the strategy is to increment q until it satisfies the equation of L in 13. For every edge, q can have two values ($q_1 = \frac{1}{q_2}$) depending on at which point the edge starts. In this case the starting cell size is known, therefore it is possible to determine if q will be $0 \leq q \leq 1$ or $q \geq 1$ with the condition $N \times \Delta l_1 \geq L$. In fact, if $N \times \Delta l_1$ is bigger than the edge length, then the cell size will decrease from the starting cell to the ending cell, which means that q will be $0 \leq q \leq 1$. On the contrary, q will be $q \geq 1$. Finally, the last point to study before writing the `mesh` file is the interface between the static and the rotating part. As already mentioned, the two parts are meshed separately before being merged together. During the first step, the meshes are independent and what will be the interface between the two parts is currently a boundary.

3.2.1. *Another mesh generator:* ***SnappyHexMesh.*** `snappyHexMesh` is the preferred mesh generator for complex geometries available in OpenFOAM®. The idea is to transform a triangulated surface geometry (**ST**ereo-**L**ithography format) into a $3D$ hexahedral volume mesh. The meshing procedure is much simpler than the `blockMesh` procedure explained above. Firstly, a block mesh serving as base-case mesh created by `blockMesh` is used, into which the detailed triangulated surface geometry will be integrated. Secondly, the latter surface geometry is adapted to the base-case mesh in two steps: (1) by fit-cutting the airfoil profile according to existing hexahedral cells on the base-case mesh (Fig. 6 top-left), (2) by snapping the cells to properly shape the original airfoil into a smooth profile (Fig. 6 top-right). Finally, inflation layers are added around the airfoil profile to try to capture to a certain extent the boundary layer physics (Fig. 6 bottom-left). The resulting rotating mesh is shown in Fig. 6 bottom-right.

3.2.2. *Another mesh generator:* ***overSetMesh.*** Generally, a continuous mesh is employed for domain discretization. However, certain scenarios require complex mesh procedures such as 2-type meshes (e.g., hexahedral/tetrahedral/polyhedral cells), 2-zone meshes, and in this case 2-domain meshes. The latter sample is preferred for cases in which some kind of dynamic mesh movement is involved. One instance is fluid-structure simulations (FSI), which normally apply one meshed domain for the fluid part solved by the Finite Volume Method (FVM) and another meshed domain for the solid part solved by the Finite Element Method (FEM). Another example is the moving frame used in this paper for the dynamic stall study, in which the static domain can be considered as the base-case mesh serving as a background domain for the whole computational region and a foreground rotating domain made by `overSetMesh`. Between the two domains an interpolation methodology is carried out by the OverSet Grid Assembly (OGA) approach. The idea behind this approach is to improve the domain interpolation method by assigning a specific property to each pair of cells at the domain interface, thus interface cells may be called receivers/donors that will eventually weight the interpolation between each pair of receiver-donor cells. More information can be found in [35]. Nevertheless, the domain interpolation procedure has some disadvantages. Firstly, it has been previously highlighted in [36] that an intensive computational cost is associated to domain interpolation due to the overSet patches re-ensembled approach per time-step. Additionally, that issue is intensified when using parallel domain decomposition for High Power Computing (HPC), in which overSet patches are unevenly distributed between processors. Finally, some mass conservative outcomes have also been identified in [36], particularly at the domain interface. Figure

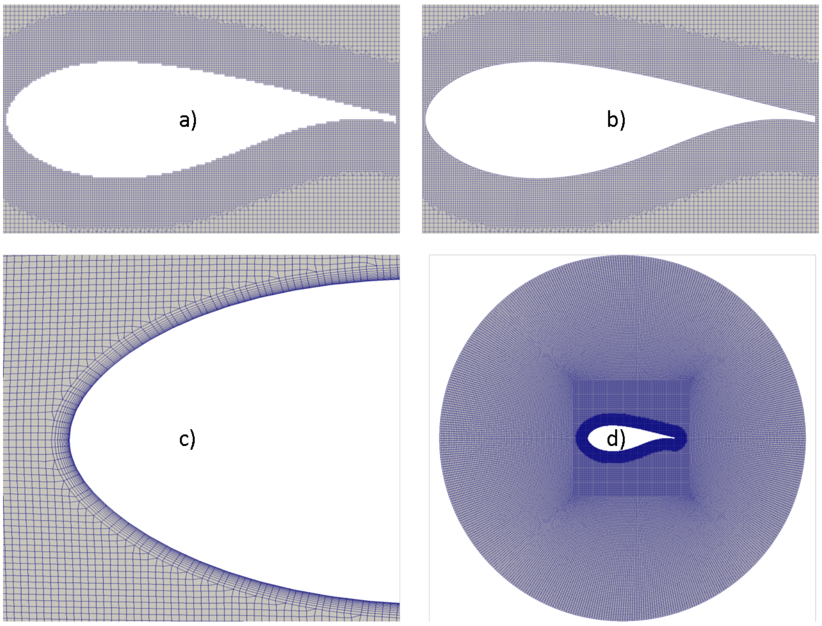


Figure 6. `SnappyHexMesh` steps creation.

7 below displays both background and foreground meshes individually made by `blockMesh` and joined together by `overSetMesh`.

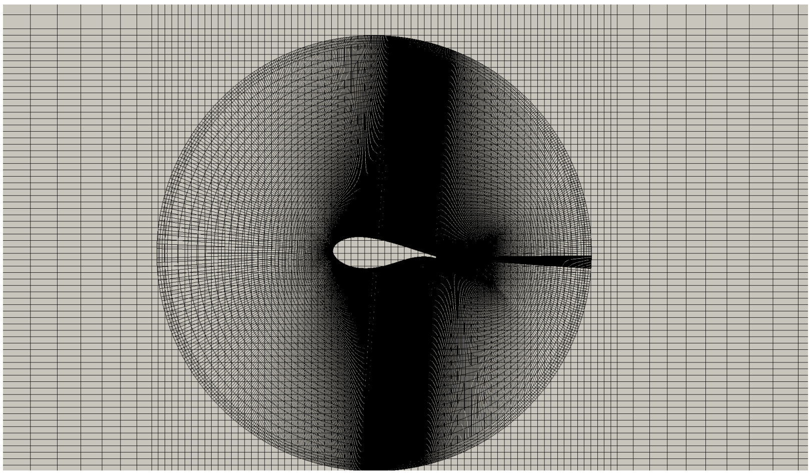


Figure 7. `overSetMesh` superimposed domains.

3.3. Boundary conditions.

The initial flow velocity and static pressure are respectively equal to U_∞ and 0. As shown in Fig. 3, an inlet boundary condition is placed on the blue-colored curved part of the exterior boundary of the computational domain. A velocity value U_∞ is imposed on this boundary with a `fixedValue` type condition and a Neumann condition is set for the static pressure with a `fixedFluxPressure` type condition. This Neumann condition makes it possible to adjust the pressure gradient to ensure equality between $\boldsymbol{U} \cdot \boldsymbol{S}_f$ ($\boldsymbol{S}_f$ being the boundary face surface) and the volumetric face flux at the boundary. On the red-colored rectangular part of the exterior boundary of the computational domain in Fig. 3, an `outlet` flow condition is placed. A static pressure value, hence $p = 0$, is imposed on this boundary with a `fixedValue` type condition and an `inletOutlet` type condition is used for the rest of the variables ($\boldsymbol{U}$, k and ω). This boundary condition provides a generic outflow condition, with specified inflow for the case of return flow. The types of boundary conditions used in the simulations are summed up in Tab. 1.

On the airfoil wall, as shown in Fig. 3, wall functions are used for k, ω and ν_t. The `kLowReWallFunction` is used for k. It provides a Neumann pure zero-gradient boundary. This boundary condition

Table 1. Boundary condition types used in the simulations.

Boundary condition	inlet	outlet	wall
$\boldsymbol{U}$	fixedValue	inletOutlet	fixedValue
p	fixedFluxPressure	fixedValue	zeroGradient
k	fixedValue	inletOutlet	kLowReWallFunction
ω	fixedValue	inletOutlet	omegaWallFunction
ν_t	calculated	calculated	nutUSpaldingWallFunction

provides a turbulence kinetic energy for low and high-Reynolds number turbulent flow cases; further details can be found in [37]. The model operates in two modes, based on the computed laminar-to-turbulent switch-over $y+$ value. The `omegaWallFunction` is used for ω. This is a special wall function that switches between viscous and logarithmic regions according to the position of $y+$. At the intersection of these two regions, the value is calculated by blending the viscous and log-law sublayer values; a broader description can be found in [37]. Recent tests have shown that the standard switching method provides more accurate results for $10 < y+ < 30$ when used with high Reynolds number wall-functions and both methods provide accurate results when used with continuous wall-functions, as verified by [38]. The `nutUSpaldingWallFunction` is used for ν_t. This wall function can operate in two modes, viscous or log, depending on the position of $y+$ [37]. While the `nutkWallFunction` could also be used for ν_t,it is better to put the first cell center in the logarithmic area with this function.

3.4. **Numerics.** For the present simulations, the `CrankNicolson` discretization scheme was chosen for numerical stability reasons compared to `backward`. The `CrankNicolson` time scheme is transient, second order implicit, and bounded. The discretization of the velocity and pressure gradients are specified to limit the scheme toward `upwind` in regions of rapidly changing gradients. The `Gauss` entry specifies the standard finite volume discretization of Gaussian integration which requires the interpolation of values from cell centers to face centers. The interpolation scheme is then given by the `linear` entry, meaning linear interpolation. Additionally, the `cellMDLimited` scheme limits the gradient such that when cell values are extrapolated to faces using the calculated gradient, the face values do not fall outside the bounds of values in surrounding cells. A limiting coefficient is specified after the underlying scheme. The coefficient has a value between 0 and 1, with 0 implying high accuracy and 1 implying good numerical stability.

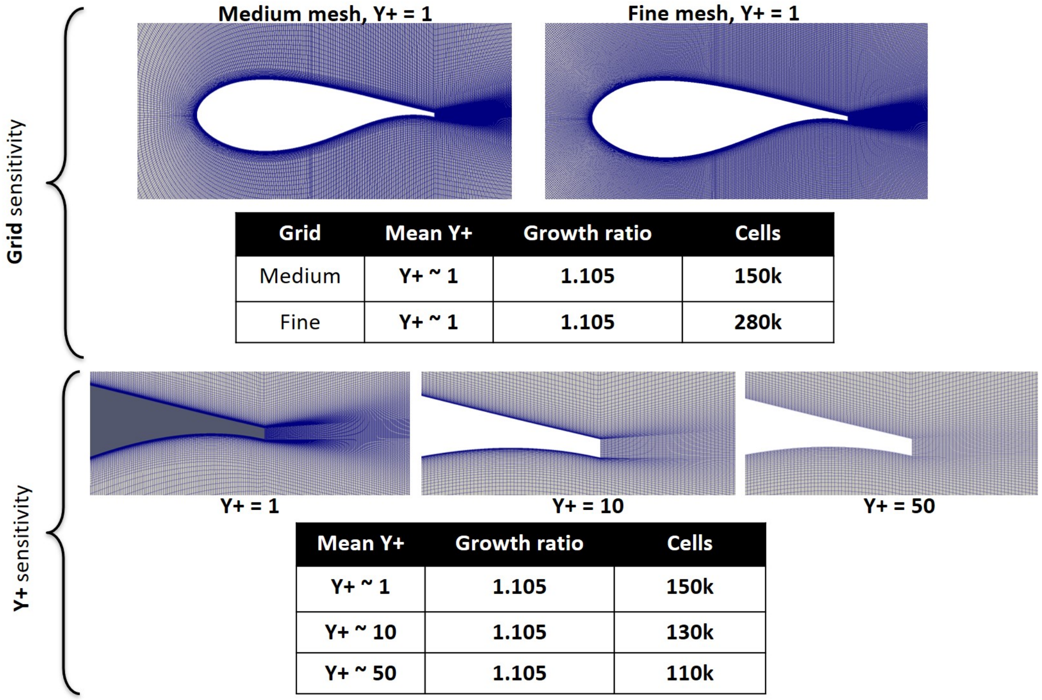
The treatment of advective terms is one of the major challenges in CFD numerics and so the options are more extensive. The schemes are all based on `Gauss` integration, using the flux (generally known as the volumetric flux of velocity on cell faces for constant-density flows and the mass flux for compressible flows) and the advected field being interpolated to the cell faces by a selection of schemes e.g., `linear`, `linearUpwind`, `upwind`, among others. In the present simulations, `linearUpwind` was chosen, which is second order, upwind-biased, and unbounded (but much less so than `linear`); it requires discretization of the velocity gradient to be specified. The `bounded` variants relate to the treatment of the material time derivative which can be expressed in terms of a spatial time derivative and convection. Particularly in steady-state simulations, it is better to include the third term (`bounded`) within a numerical solution because it helps maintain boundedness on the corresponding variable and promotes better convergence. For the other scalars (k,ω and ν_t), the second order `linearUpwind` scheme is also used to ensure boundedness of the solution.

The flow is numerically solved with a solver named `pimpleFoam`, an incompressible fluid flow algorithm of the OpenFOAM® library. The solver is transient, permitting relatively large time steps thanks to the hybrid `PISO-SIMPLE` (`PIMPLE`) algorithm. Most fluid dynamics solver applications in `OpenFOAM` use either the pressure-implicit split-operator (`PISO`), the semi-implicit method for pressure-linked equations (`SIMPLE`) algorithms, or a combined `PIMPLE` algorithm. These algorithms are iterative procedures for coupling equations for momentum and mass conservation, `PISO` and `PIMPLE` being used for transient problems and `SIMPLE` for steady-state. Both steady-state or transient algorithms solve a pressure equation to enforce mass conservation, with an explicit correction to velocity to satisfy momentum conservation. They optionally begin each step by solving the momentum equation, the so-called momentum predictor. While all the algorithms solve the same governing equations, the algorithms differ mainly in how they loop over the equations.

For the numerical simulations of the present paper, it was chosen (as a first estimation) up to 50 iterations of the momentum, pressure equation before moving to the next time step. After a while, all the variables converged within the residual restrictions at the 10^{th} iteration. In the paper, as previously explained, the method of choice is the `PIMPLE` algorithm due to the possibility of speeding up the process in order to reduce the computational effort i.e., $Co > 1$. In fact, each time step can be much more expensive than a time step in the `PISO` calculation but the time step itself can be much larger. In addition, significant information may be lost, which changes the flow pattern. This is why a time-step sensitivity analysis was conducted (explained below) to overcome this issue. In this paper, numerical issues are solved until certain criterion values are reached. These values are 10^{-6} for P and 10^{-5} for $\boldsymbol{U}$, k and ω. It is important to reach these pre-defined values for each angle of attack. The quantitative results strongly depend on these convergence criteria. The relaxation factors, specific to a steady state simulation, are set to 0.7 for $\boldsymbol{U}$, k and ω, and 0.5 for the pressure. The initial conditions for the transient URANS models were initialised with the results from a steady-state RANS simulation to accelerate the convergence.

4. Sensitivity analysis

Before post-processing the dimensionless numbers previously introduced in section 2.2, different types of mesh refinement far from and close to the wall are generated with the same mesh topology in order to perform a grid sensitivity analysis. Figure 8 illustrates the grid (top of the figure) and the $y+$ (bottom of the figure) used for the sensitivity analysis, respectively. In the grid analysis, two cell sizes were proposed, namely $fine$ and $medium$ mesh with $280k$ and $150k$ cells, respectively. The cell size close to the wall (i.e., $y+ = 1$) was kept the same for both configurations, only the cell size away from the wall was changed. In the region close to the wall, three $y+$ configurations were made: $y+ \approx 1$, $y+ \approx 10$, and $y+ \approx 50$, to assess the boundary layer when it is fully resolved or modeled by wall functions. It is noticeable that the same cell growth ratio was used for all simulations (≈ 1.1), in order to preserve a fairly good mesh quality.



Grid	Mean Y+	Growth ratio	Cells
Medium	Y+ ~ 1	1.105	150k
Fine	Y+ ~ 1	1.105	280k

Mean Y+	Growth ratio	Cells
Y+ ~ 1	1.105	150k
Y+ ~ 10	1.105	130k
Y+ ~ 50	1.105	110k

Figure 8. Computational meshes for the grid (top) and $y+$ (bottom) sensitivity analysis.

4.1. **2D URANS.** In Fig. 9 below, static $URANS$ results with a low angle of attack (up to 12^o) by means of the $k-\omega\ SST$ turbulence model are first compared against the experimental measurements made by [5] (black) using the above mesh set-ups. Lift (top-left), drag (top-right), and momentum (bottom) coefficients are displayed. Globally, numerical results exhibit fairly good agreement with experiments, particularly for the lift coefficient from 0^o to 9^o. From 9^o to 12^o better agreement was found for drag and momentum coefficients. $y+ = 50$ (yellow) was the least satisfactory agreement for the momentum coefficient compared to experimental results, hence indicating the need for of a proper mesh refinement close to the wall, as proved by both $y+ = 1$ (fine and medium meshes) with best agreement for all force coefficients. Moreover, before proceeding with a higher angle of attack (AOA) to study flow separation, a dynamic study with the airfoil undergoing pitching motion (in which the sliding mesh method is employed to couple the rotational sub-domain to the rest of the domain) is preferred to study stall conditions, i.e., dynamic stall.

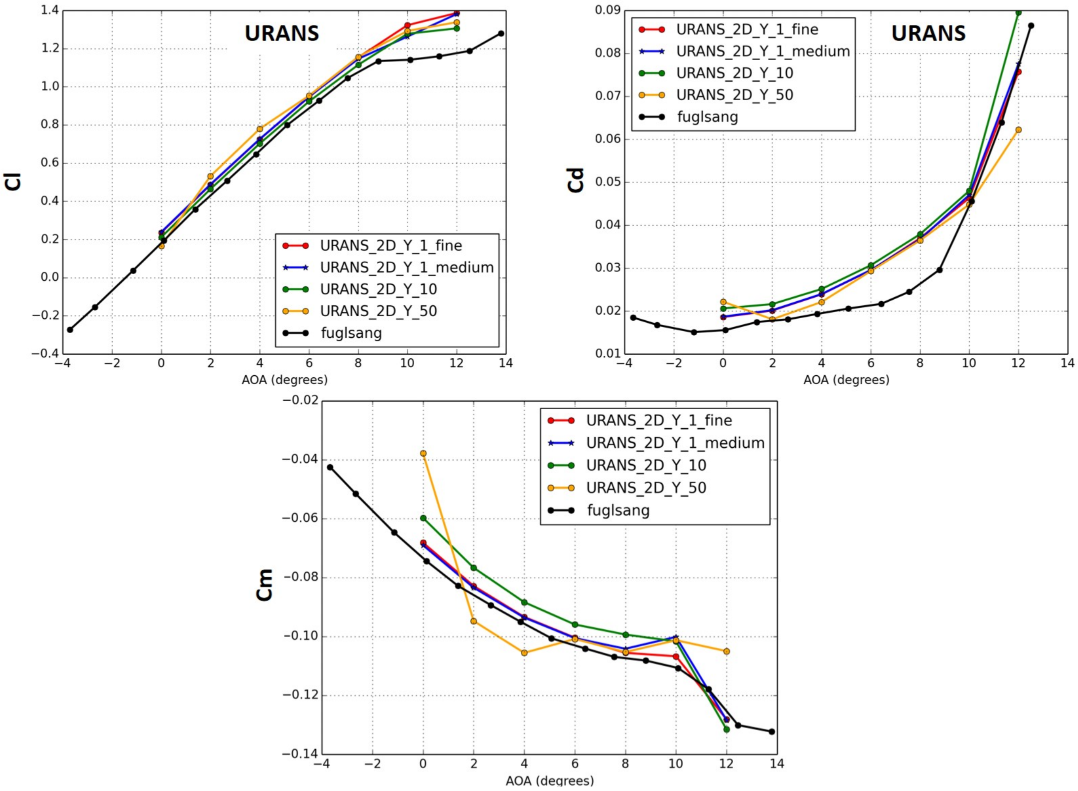


Figure 9. $y+$ and mesh sensitivity analysis for lift (top-left), drag (top-right) and momentum (bottom) coefficients, respectively, under $URANS$ non-stall conditions, compared with experimental measurements (black), [5].

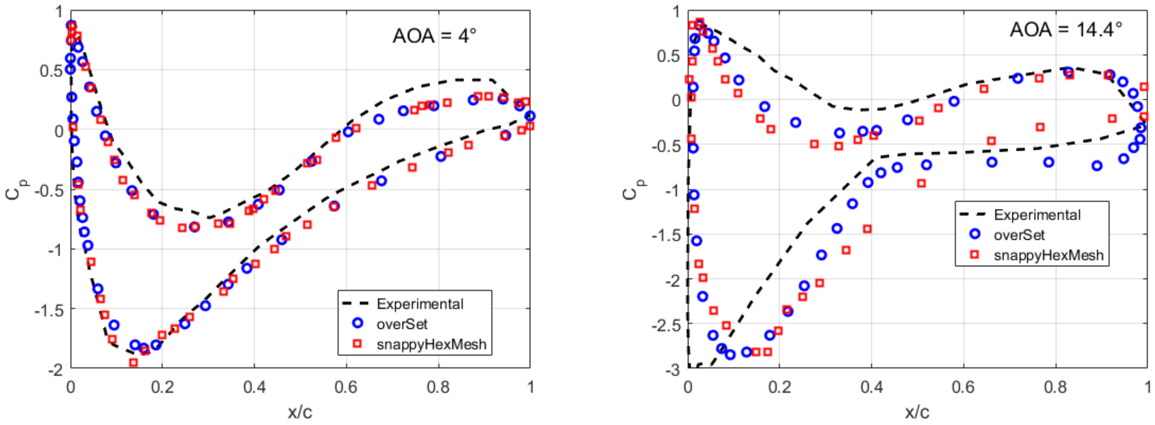
Before the dynamic stall study, a mesh generator comparison analysis was carried out using $URANS$. With regard to the rotating domain, it was fully meshed using `snappyHexMesh`, and `blockMesh` for the static domain. For the case of `overSetMesh`, both static and rotating domains were meshed with `blockMesh` and then later joined together by `overSetMesh`, as explained in section 3.2.2. Finally for the case of `blockMesh`, both domains were meshed by `blockMesh`. Table 2 displays some relevant characteristics of the three meshes, such as *Non-Orthogonality*, *Skewness*, $y+$ and number of cells. Regarding $y+$, two regions close to the wall were generated, namely $y+ = 10$ and $y+ = 20$ for the case of `snappyHexMesh`.

Table 2. Mesh generators comparison.

Mesh generator	Non-Orthogonality	Skewness	$y+$	Cells
`blockMesh`	58	0.84	10	130K
`snappyHexMesh`	46	2.95	20	100K
`overSetMesh`	58	0.84	10	130K

The advantageous feature behind `snappyHexMesh` is the possibility of local mesh refinement, for instance close to the airfoil walls, without the need to refine far from the focused region as in the structured approach used by `blockMesh`. Consequently, the expected number of cells should be by definition much lower in this case. It should be noted that *skewness* and $y+$ increase to 2.95 and 20 respectively, for the case of `snappyHexMesh`. The *skewness* represents the deviation of a vector **d** joining two cell-centers, from the face-center crossed by the vector. This deviation is represented by a vector δ. In OpenFOAM® the skewness is computed as $S_k = \frac{|\delta|}{|\mathbf{d}|}$. The downside of the local refinement made by `snappyHexMesh` is these highly skewed cells at the interface of the refined and non-refined regions. Nonetheless, even though the skewness value for `snappyHexMesh` is appreciably higher than that of `blockMesh` and `overSetMesh`, it is still within an acceptable quality tolerance. Regarding the non-orthogonality, it can be defined as the angle made by the vector between two adjacent cell centers and the common face normal. Acceptable standardized values in OpenFOAM® for a reasonable quality mesh are below 60^o.

Figure 10 below displays the pressure coefficient distribution along the airfoil profile (i.e., as a function of x/c) for an angle of attack (AOA) of $\alpha = 4^o$ (on the left) and $\alpha = 14.4^o$ (on the right). Good agreement is found for both AOAs and both `snappyHexMesh` and `overSetMesh` mesh approaches, and the pressure tendency along the profile is well captured, specifically for lower AOAs. This behavior is expected, since flow instabilities that develop for higher AOAs are not being properly captured. In the next section a detailed analysis of dynamic stall will better explain this issue.



(a) - AoA $\alpha = 4°$ (b) - AoA $\alpha = 14.4°$

Figure 10. Pressure coefficient distribution using `snappyHexMesh` and `overSetMesh` for $\alpha = 4°$ (left) and $\alpha = 14.4°$ (right). `blockMesh` results, closely mirroring those of `snappyHexMesh`, have been omitted from the plot for the sake of visual clarity and conciseness.

With regard to `overSetMesh`, the simulation time to converge is greatly increased compared to `blockMesh` and `snappyHexMesh` by a factor of 5, even while using a simplistic approach such as *RANS*. As pointed out in section 3.2.2 based on findings in [36], the `overSetMesh` drawback associated with the domain interpolation time delay issue is replicated here by using two parallel decomposition approaches (i.e., *simple* and *scotch*), in which the pseudo-random distribution of processors greatly increased the simulation time using *scotch*, as was observed by the escalated pressure iteration loops per time step. Table 3 gives the relative error percentage calculated in the following form: $error = \frac{|P-O|}{O}.100$, where P and O stand for numerically predicted and experimentally observed values respectively. The relative error percentage is comparable between both methods for the lift coefficient (Cl); however, it is notably diminished for the drag coefficient (Cd) when employing overSetMesh. This phenomenon is likely attributed to the substantial mesh point density inherent in overSetMesh, which affords a more comprehensive resolution of phenomena closely associated with the drag coefficient (Cd). While the meshing procedure with `overSetMesh` is decidedly simpler than that with `blockMesh` and `snappyHexMesh`, it falls short in terms of simulation time and accuracy of results, when compared with existing experimental data. Accordingly, the focus of subsequent simulations will shift towards a comparative analysis of `blockMesh` and `snappyHexMesh`.

Table 3. Mesh generators comparison.

Mesh generator	Cl, $AOA = 4$			Cd, $AOA = 4$		
	URANS	Exp.	Error [%]	URANS	Exp.	Error [%]
`snappyHexMesh`	0.67	0.64	4.7	0.026	0.019	36.8
`overSetMesh`	0.67	0.64	4.7	0.021	0.019	10.5

4.2. **Convergence analysis for dynamic stall.** Identical set-up parameters, detailed earlier, are utilized to investigate statistical convergence during an airfoil pitching motion. This pitching motion is described by the sinusoidal equation $\alpha = A.\sin(wt) + \alpha_m$, where $k = \frac{wc}{2V_\infty} = 0.070$, and amplitudes span between 1.4° and 2°. Here, w represents the pitching frequency, c denotes the chord length, and V_∞ signifies the freestream velocity. Figure 11 displays the instantaneous lift coefficients C_l at various AOAs during the pitching cycles. A total of nine pitching cycles were examined, across three different ranges of AOAs: $\alpha_m = 3.8^o$ (left image), $\alpha_m = 8.9^o$ (center), and $\alpha_m = 14.8^o$ (right). As an illustration, the third image on the right reveals instantaneous C_l during nine pitching cycles at different α values: $15^o \uparrow$, $16^o \uparrow$, $16^o \downarrow$, $15^o \downarrow$, $14^o \downarrow$, $14^o \uparrow$. Here, $\uparrow$ denotes upstroke, while $\downarrow$ indicates downstroke (labeled as up/down in the figure). From Fig. 11, it is clear that the first three C_l values (up to the third pitching cycle) diverge from those after the fourth pitching cycle. Further, discrepancies between the C_l in the fourth cycle and the subsequent cycles (up to the ninth pitching cycle) are minimal. Thus, ensuing pitching airfoil analyses in this work will consider averages from the fourth cycle to the sixth.

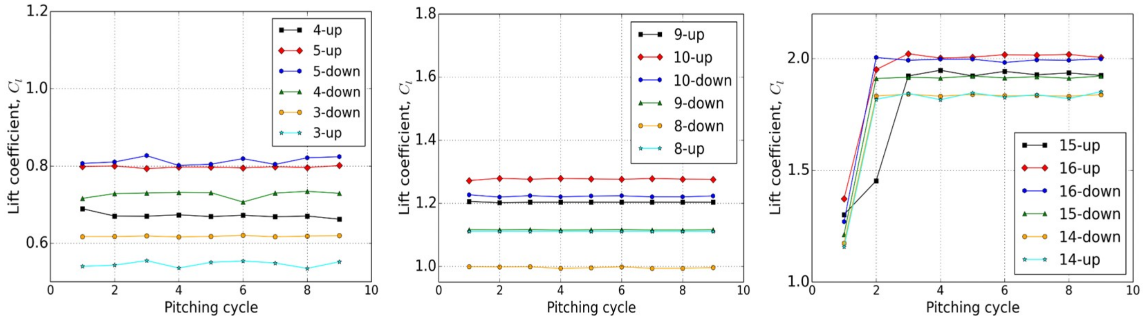


Figure 11. Convergence sensitivity analysis. Instantaneous lift coefficient at six angles of attack (during upstroke/downstroke motion) as a function of pitching cycle for three ranges of angle of attack (AOA): $\alpha_m = 3.8^o$ (left), $\alpha_m = 8.9^o$ (center), and $\alpha_m = 14.8^o$ (right).

4.3. **Grid sensitivity study.** The medium and fine computational grids depicted in Fig. 12 with the grid properties listed in Fig. 8 were used for a grid-sensitivity analysis. For low AOA (left) the dynamic motion is counter-clockwise and for AOA higher than or equal to 8^o (right) the dynamic motion is clockwise. It is found that the downstroke motion of the fine and medium meshes (red and blue, respectively) with a low angle of attack (left) is quite similar to experiments. However, a downside peak for both upstroke motions is observed. On the right of the figure, both meshes failed to predict a good agreement with experiments. Such behavior may be due to the unsuitability of $RANS$ turbulence models when solving completely the boundary layer. Firstly, the noisy behavior observed on both the left and right images in the figure is due to incomplete statistical convergence, caused by an increase in numerical instabilities. The computed mean value exhibits an overestimation when juxtaposed with the measured data. Notable deviations are discernible during the downstroke motion, attributable, at least in part, to inherent limitations associated with unsteady RANS turbulence modeling - a phenomenon corroborated in other numerical simulations [39]. Analogous findings have also been reported in related literature by Qian et al. for example [40]. Consequently, even though $k - \omega$ SST is highly recommended in the literature, when it is not accompanied with a proper mesh resolution close to the walls (i.e., $y+ \approx 1$, and

$\Delta y/\Delta z \approx 1$, which does not apply to the present case) under the actual dynamic stall conditions, strong numerical instabilities emerge, which slow down the convergence rate, and therefore the simulation, and additionally prompt the use of lower-order numerical time schemes.

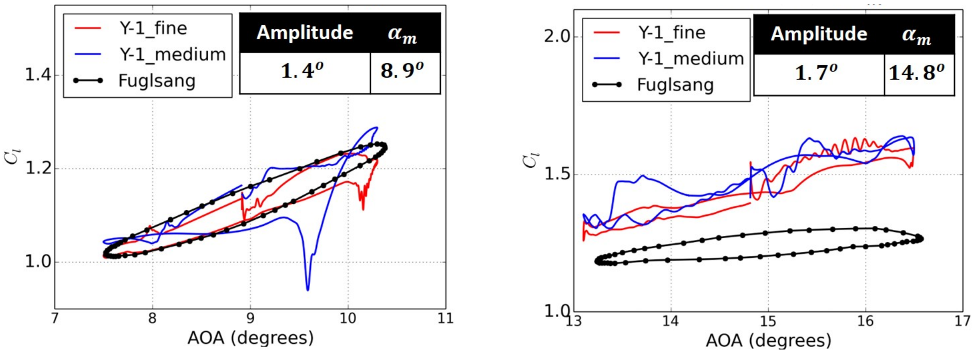


Figure 12. Fine (red) and medium (blue) mesh refinements, both using $y+ = 1$, compared against experimental measurements (black) [5] under dynamic stall conditions $\alpha_m = 8.9^o$ (left), and $\alpha_m = 14.8^o$ (right).

As the $y+ \approx 1$ mesh cases were rather problematic, switchable wall-functions were employed for $y+ > 1$ that are capable of adapting to the sub-laminar viscous and log-region of the boundary layer [37]. Results are shown in Fig. 13 below. From left to right a range of AOA starting at α_m were made: $\alpha_m = 3.8^o$ (left), $\alpha_m = 8.9^o$ (center), and $\alpha_m = 14.8^o$ (right). Two $y+$ were evaluated, namely $y+ = 10$ (red) and $y+ = 50$ (blue). Good agreement was found between numerical results and experimental measurements for low and medium AOA ($\alpha_m = 3.8^o$ (left), $\alpha_m = 8.9^o$ (center), respectively).

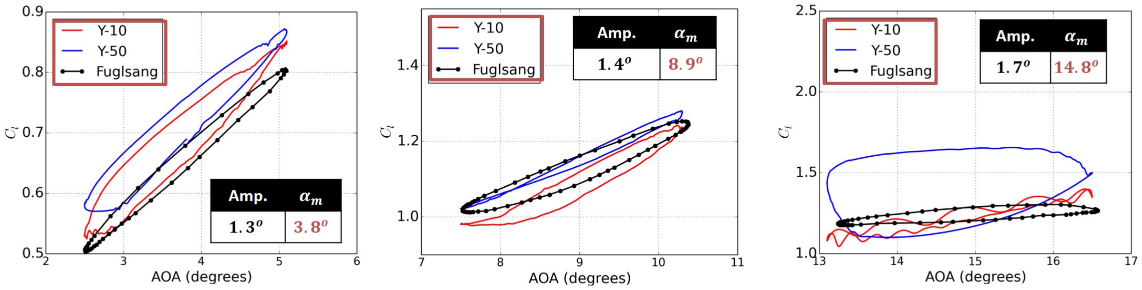


Figure 13. $y+$ sensitivity analysis in dynamic stall condition, compared with experimental measurements (black) [5].

Nonetheless, in these relatively strong dynamic stall conditions ($\alpha_m = 14.8^o$ on the right) a proper mesh refinement close to the walls becomes necessary due to a more complex flow pattern such as flow separation in the upper part of the airfoil, as verified with higher $y+$ values i.e., $y+ = 50$ (blue) which largely mismatched experiments. The present results coincide with previous numerical results [17], in which $y+$ values lower than and equal to 10 were found to be below the limit at which results become inaccurate and therefore gave similar agreement with experiments, which confirms the selection of $y+ = 10$ for the rest of the test cases in the present study.

4.4. **Time step size study.** As explained in the numerics section 3.4, the method of choice is the `PIMPLE` algorithm due to the possibility of going faster in time in order to reduce the computational effort i.e., $Co > 1$. From the physical point of view, the time step is associated with the turbulent time-scale and the integral time-scale of the flow field. A proper selection of the time-step has to be made in order to properly capture the physics on the airfoil such as the leading and trailing edge vortices and the wake downstream. Figure 14 below shows on the top different time-step values that correspond to maximum CFL numbers of 4, 6, 8, 10, 15 and 20. From left to right, a range of AOA starting at α_m were made: $\alpha_m = 3.8^o$ (top-left), $\alpha_m = 8.9^o$ (top-center), and $\alpha_m = 14.8^o$ (top-right). No significant differences were observed for CFL numbers between 4 and 10. However, for medium and high AOA, top-center and top-right respectively, larger CFL numbers tend to destabilize the simulation numerically. Consequently, adapted time and convective schemes were used instead.

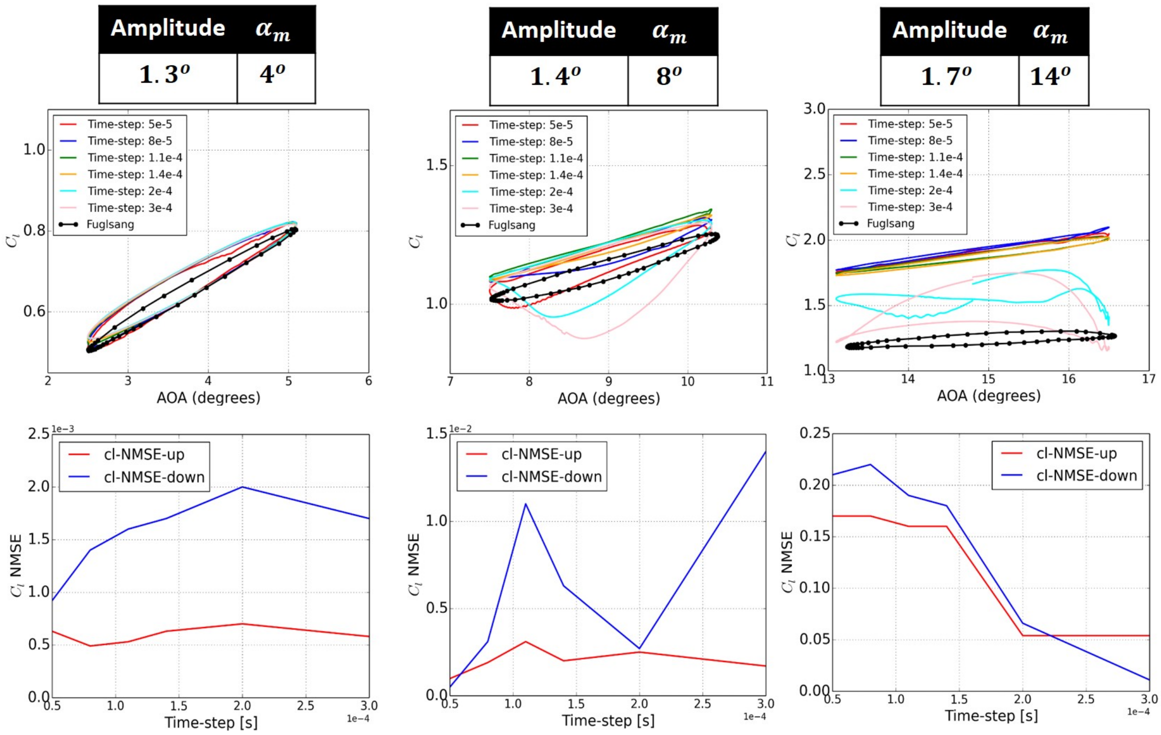


Figure 14. Time-step sensitivity analysis under dynamic stall conditions (top) for lift coefficient (C_l). Increasing initial angle of attack (α_m) from left to right, 4^o, 8^o and 14^o, respectively, compared with experimental measurements (black) [5]; Normalized Mean Square Error ($NMSE$) during upstroke (red) and downstroke (blue) motion (bottom).

Quantitatively speaking, a performance measure was used, the Normalized Mean Square Error, $NMSE = \frac{\langle (O-P)^2\rangle}{\langle O\rangle\langle P\rangle}$, where O, and P stand for *observed* (experimental) and *predicted* (numerical) values respectively. The ideal value indicating a perfect match between CFD and experiments would be $NMSE = 0$. At the bottom of Fig. 14 above, the results are separated into downstroke and upstroke motion errors (blue and red respectively). From left to right, a range of AOA starting at α_m were made: $\alpha_m = 3.8^o$ (bottom-left), $\alpha_m = 8.9^o$ (bottom-center), and $\alpha_m = 14.8^o$ (bottom-right). In general the downstroke dynamic motion presents a greater error than its upstroke dynamic counterpart, which is consistent with the hysteresis loop behavior observed in the experimental measurements. As expected, the biggest disagreement is observed for the most unfavorable flow separation condition ($\alpha_m = 14.8^o$, top-right). Even though lower values of $NMSE$ are obtained with higher CFL, those values correspond to lower order time and convective schemes. Ultimately, the chosen value would be $CFL = 8$, with which high order time and convective schemes have been used and presented a fairly good agreement with experiments.

Lastly, a dynamic stall sensitivity study was performed using the following parameters: $y+$ (1, 10, 50), grid spacing (fine, medium mesh), time-step (CFL from 4 to 20), for a range of AOA starting at α_m : $\alpha_m = 3.8^o$ (AMP_1), $\alpha_m = 8.9^o$ (AMP_2), and $\alpha_m = 14.8^o$ (AMP_3). Since the focus in an industrial approach is to find a balance between accuracy and simulation time efficiency, Tab. 4 below displays the simulation time per dynamic cycle. Globally, the simulation time increases as the AOA increases, due to a rise in flow separation complexity. Additionally, as expected, with lower mesh refinement $y+ \approx 10$, the total converged simulation time is smaller, even comparable with the other cases $y+ \approx 1$ for only one cycle (i.e., six times faster).

Once the sensitivity analysis had been conducted, a mesh comparison survey was performed under dynamic stall conditions using a $2D$ URANS pitching airfoil via the $k-\omega$ SST turbulence model. The final computational setup described above (time-step, $y+$, convergence time and grid-sensitivity) is exploited in subsequent results. Figure 15 below compares the lift (left) and drag (right) coefficients obtained with the two mesh generators `blockMesh` (blue line) and `snappyHexMesh` (orange line) versus the experimental data (black dashed-line). It can be seen that `snappyHexMesh` (orange line) provides the best fit with experiments (black dashed-line) on both lift (left) and drag (right) coefficients, particularly for the

Case	Simulation time
$y+ = 1$, fine mesh, AMP_2	$17h$ / cycle
$y+ = 1$, fine mesh, AMP_3	$26.5h$ / cycle
$y+ = 1$, medium mesh, AMP_2	$14h$ / cycle
$y+ = 1$, medium mesh, AMP_3	$23h$ / cycle
$y+ = 10$, AMP_2	$18h$ / **6 cycles**
$y+ = 10$, AMP_3	$22h$ / **6 cycles**

Table 4. Simulation time per cycle under dynamic stall conditions using different $Y+$ and mesh refinement.

downstroke motion (lower part of the hysteresis loop shape), which agrees quite well for the lift (left) coefficients.

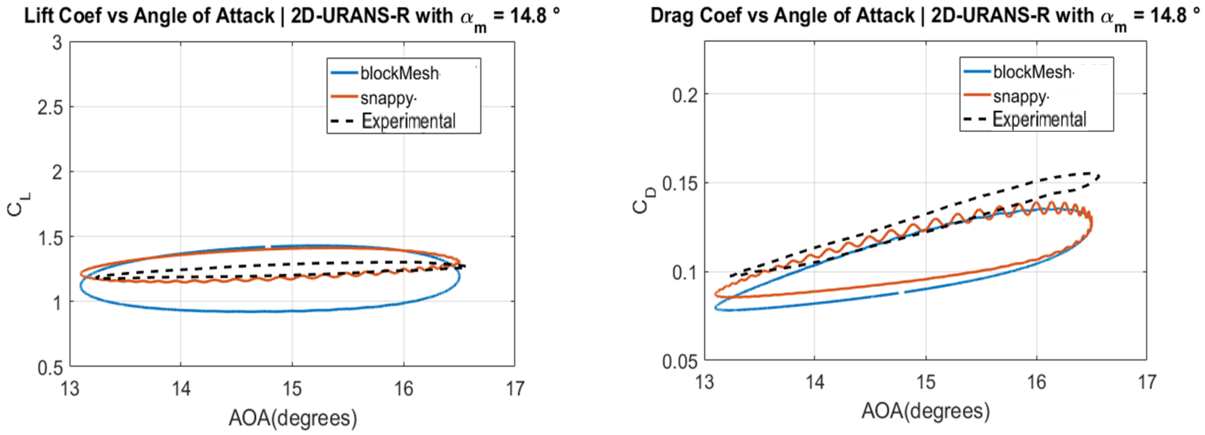


Figure 15. Mesh comparison between `blockMesh` (blue line), `snappyHexMesh` (orange line) and experimental data (black dashed-line) by means of lift (left) and drag (right) coefficients.

A reasonable source could be the O-grid mesh shape generated by `snappyHexMesh` via its local refinement feature compared to the C-grid mesh arrangement produced by `blockMesh`. As previously verified by other studies in [41], the O-grid displays a pronounced advantage over C-grid when large flow gradients appear in the vicinity of the trailing edge. Moreover, the O-grid typically exhibits a clustering of grid points around the trailing edge, and the gradient metrics is considerably smoother. For certain critical flow conditions, e.g., a combination of high Reynolds number and high AOA, the O-grid displays superior numerical stability when compared with the C-grid style by `blockMesh`. Lastly, the two mesh generators `blockMesh` (blue line) and `snappyHexMesh` (orange line) mismatch the experimental hysteresis loop for the drag (right) coefficient. Furthermore, some numerical instabilities can still be seen (wavy orange line) for `snappyHexMesh` on both lift (left) and drag (right) coefficients. While one may think that highly skewed cells might produce such behavior, some turbulence modeling issues are highlighted instead below.

4.5. **Turbulence models comparison.** Figure 16 compares the turbulence models against experiments (black) using Spalart-Allmaras (SA, red), Scale-Adapting Simulation (SAS, blue), $k-\omega$ SST (SST, green), and transitional-SST ($transSST$, yellow) for lift (left) and drag (right) coefficients, respectively. Firstly, the transitional-SST ($transSST$, yellow) is highlighted as the model with the least agreement with experiments (black). One reason for this behavior may be the incorrect modeling of the laminar to turbulent boundary layer transition based on the intermittency term (γ, Eqn. 7), as the onset of transition could be affected by (1) $3D$ effects that might trigger a different transition boundary layer behavior, and (2) the non-resolved/modeled boundary layer by using wall-functions that might affect both laminar, transition, and turbulent development within the boundary layer.

Secondly, regarding the Scale-Adapting Simulation (SAS, blue), results depend on the local flow inhomogeneities, which in the case of $2D$ simulations, are not properly captured, especially for high AOA beyond stall conditions ($14^o < \alpha_m < 16^o$). Finally, both the Spalart-Allmaras (SA, red) and $k-\omega$ SST (SST, green) models display the best match with experiments. The Spalart-Allmaras (SA, red) turbulence model is the only one with a hysteresis loop width that is comparable with experiments, particularly

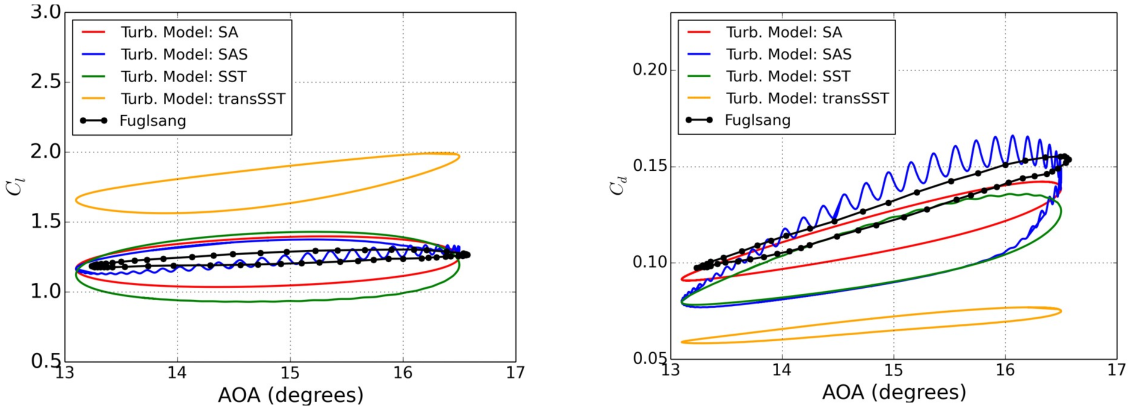


Figure 16. Turbulence models sensitivity analysis under dynamic stall conditions for lift (left) and drag (right) coefficients, respectively, compared with experimental measurements (black) [5].

for the drag (right) coefficient. Both the Spalart-Allmaras (SA, red) and $k-\omega$ SST (SST, green) models overestimate the hysteresis loop for the lift (left) coefficient. In general, it may be concluded that the results for Spalart-Allmaras (SA, red) are the closest to the experimental results.

Once the turbulence modeling sensitivity analysis had been done over the airfoil in order to find the best set-up using also previous parameters such as: $y+$, grid refinement, time-step, convergence cycles, the two mesh-procedure comparison was performed again and is displayed in Fig. 17 below. `blockMesh` (blue) and `snappyHexMesh` (orange) are compared against experimental data (black dashed-line) under dynamic stall conditions for lift (left) and drag (right) coefficients, respectively. Simulations were made using the Spalart-Allmaras (SA) turbulence model. In Fig. 17, both methods `blockMesh` (blue) and `snappyHexMesh` (brown) show fairly good agreement with experiments. The hysteresis loop is overestimated with `blockMesh` (blue) for the lift (left) coefficient, while during downstroke motion `snappyHexMesh` (orange) matches quite well with experimental results (black). On the other hand for the drag (right) coefficient, `snappyHexMesh` (orange) fails to predict the experimentally observed results, which are captured slightly better by `blockMesh` (blue). Lastly, it can be seen that the simulation time is much faster ($4x$) with `snappyHexMesh` (orange) compared to `blockMesh` (blue).

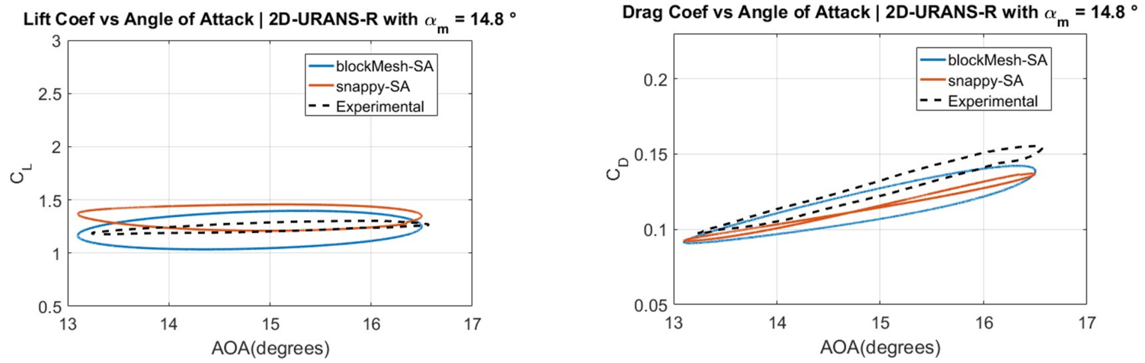


Figure 17. SnappyHexMesh (orange) and blockMesh (blue) under dynamic stall conditions with $\alpha_m = 14.8^o$ using the Spalart-Allmaras (SA) turbulence model on both meshes, compared against experimental measurements (black) [5], for lift (left) and drag (right) coefficients, respectively.

As a reminder, the idea behind `snappyHexMesh` (orange) is the advantageous feature of local refinement close to the airfoil walls, without mesh over-refinement far from the wall, which is indeed the method of the well-known structured mesher `blockMesh` (blue). This local pseudo-structured refinement close to the wall prompted a prolific O-grid style as previously explained in Fig. 15. Nonetheless, numerical instabilities can be seen (wavy orange line) on Fig. 15 for `snappyHexMesh` on both lift (left) and drag (right) coefficients, which is no longer the case using the Spalart-Allmaras (SA) turbulence model. A possible reason is the development of vortices close to the trailing edge that may not have been accurately envisaged by the $k-\omega$ turbulence model in $2D$ simulations. It is to be recalled that $k-\omega$ turbulence model switches from $k-\epsilon$ to $k-\omega$ from outside to inside of the boundary layer, respectively. On the contrary, the Spalart-Allmaras (SA) turbulence model is more indulgent in drastic mesh changing scenarios by dumping any gradient calculation causing numerical instabilities. Certainly, the thickness of the locally refined mesh region on the O-grid mesh generated by `snappyHexMesh` is nearly constant, while the boundary layer thickness is increasing downstream of the flow separation point due to positive pressure gradient. As will be noticed in next sections below, growing vortices from the separation point to the trailing edge and beyond were observed in $2.5D$ comparisons. Those vortices near the edge could be numerically underpredicted by the $k-\omega$ turbulence model while using `snappyHexMesh` in blade-pitching motion conditions. Conversely, these instabilities on the lift and drag coefficients were not observed using the `blockMesh` generation method in which a smoother meshing transition is prompted from the airfoil surface to the free stream.

4.6. $2D$ / $2.5D$ **Comparison.** Figure 18 below displays the $2D/2.5D$ comparison, using the same mesh generator i.e., `snappyHexMesh` for the $2D$ $URANS$ (blue line), $2.5D$ $URANS$ (red star), and $2.5D$ **Hybrid** turbulence models ($SA-IDDES$ [42], green star) compared against experimental measurements (black line). A closer agreement with experiments (black line) is obtained by adding the span-wise effect into the numerical results (red star compared to blue line). Even better numerical results are obtained, however, if a more elaborate turbulence model based on the Detached Eddy Simulation (DES) approach is used. (DES) models, which are an improved version of $URANS$, switch explicitly between $RANS$ and LES model formulations based on the local grid spacing and turbulent length scale. The original intent of DES was to be run in $RANS$ mode for attached boundary layers (low AOA) and to switch to LES mode in large separated (detached) flow regions (high AOA). The explicit switch to the LES model is however not accompanied by a corresponding transfer of modeled (RANS) turbulence to resolved (LES) turbulence. As with SAS (explained in section 6), DES relies on inherent flow instability for a quick generation of such resolved coherent structures. Due to the direct impact of the grid spacing on the $RANS$ model, DES models require more carefully crafted grids to avoid inappropriate behavior [42]. A better development of vortices is observed on the airfoil upper part, from the leading to the trailing edge.

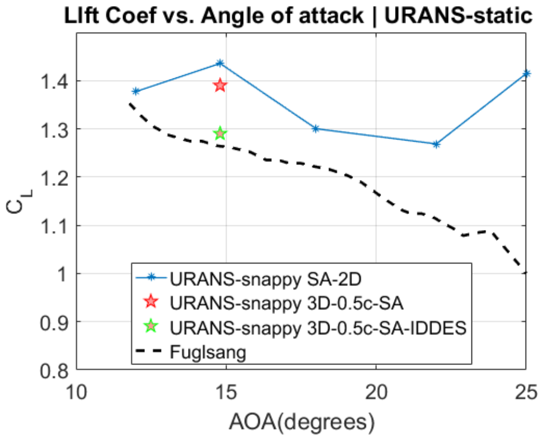


Figure 18. $2D/2.5D$ Comparison. $2D$ $URANS$ (blue line), $2.5D$ $URANS$ (red star), and $2.5D$ **Hybrid** turbulence model ($SA-IDDES$, green star) compared against experimental measurements (black line), [5].

4.7. **Span-wise sensitivity analysis.** As the Detached Eddy Simulation (DES) approach provided better agreement with experiments, a span-wise study was carried out with $\Delta x/\Delta y \approx 6$, and a constant mesh refinement in the normal direction, thus $\Delta z \approx 5/\Delta x$. Figure 19 below shows the different widths used for the span-wise sensitivity analysis in the chord-normal direction (c). The three widths - $0.2c$ width (top), $0.3c$ width (center), and $0.5c$ width (bottom) - are colored by pressure. Blue and light blue represent negative pressure values (i.e., presence of vortices), while yellow and light red represent positive pressure values. A different behavior can be observed for the smallest width of $0.2c$ (top), in which a more extended negative pressure region is found on the upper part of the airfoil, compared to the other two cases, namely $0.3c$ width (center), and $0.5c$ width (bottom). The second case, namely the $0.3c$ width (center), could be the limiting case from which vortex behavior remains constant in the span-wise direction. Nevertheless, as these are only static results, pitching airfoil simulations need to be compared with experimental data to corroborate the findings.

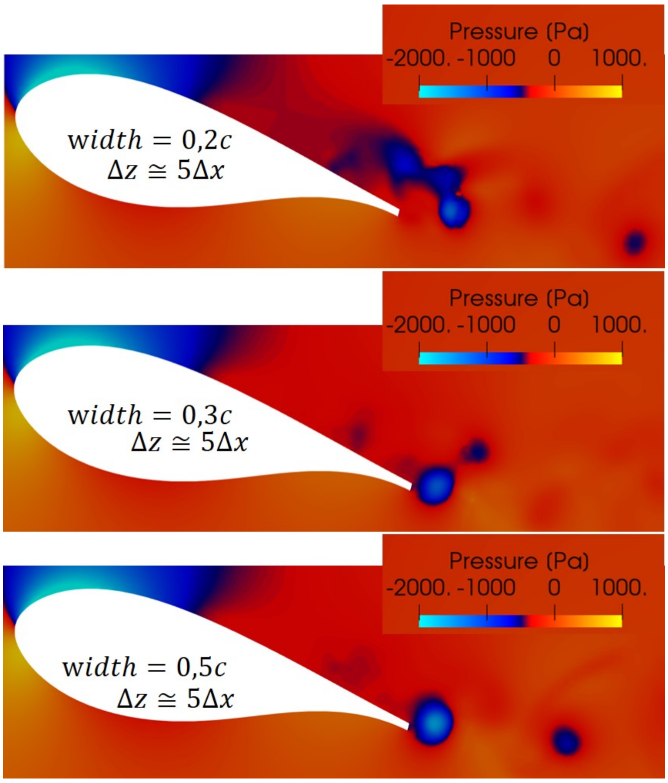


Figure 19. Pressure color map on a plane for span-wise sensitivity analysis. $0.2c$ Width (top), $0.3c$ width (center), and $0.5c$ width (bottom).

5. **Conclusion**

The behavior of a boundary layer in a positive pressure gradient (i.e., pressure increasing with distance downstream) has been studied for a pitching airfoil (i.e., during dynamic stall) namely, `FFA-W3-301`, in which the integral forces and moments experienced substantial variations and large hysteresis. A validation study of unsteady flow around a pitching `FFA-W3-301` airfoil at a Reynolds $1.6 * 10^6$ using OpenFOAM® was performed. The sensitivity analysis included $y+$, computational grid resolution, time step size, convergence pitching cycles, turbulence models, $2D/2.5D$ comparison and finally span-wise analysis. Numerical results were compared with the experimental data available in the literature ([5]), in which quasi-steady measurements at continuously varying angles of attack as well as dynamic inflow measurements were made. Firstly, an automatic mesh-generation procedure was produced using `blockMesh`, followed by `snappyMesh` and `overSetMesh`. These mesh generation procedures enabled any $2D$ and $2.5D$ airfoil shape to be automatically generated directly from its profile points, with a refinement close to the wall. Secondly, regarding dynamic stall conditions, a convergence study was conducted, and showed that differences between the force coefficients in the fourth cycle and the rest of the cycles (up to the 9^{th} pitching cycle) are negligible, thus values were averaged between the fourth and the sixth cycle. With regard to the grid sensitivity, it was found that $RANS$ turbulence models were unsuitable when solving completely the boundary layer (i.e., $y+ \approx 1$). Switchable wall-functions were therefore employed for $y+ > 1$, able to adapt to the sub-laminar viscous and log-region of the boundary layer ([37]), and were found to improve the experimental agreement (i.e., $y+ \approx 10$). In the time-step analysis, the most stable solution was obtained with up to $CFL = 8$ for which high-order time and convective schemes were used. Concerning the turbulence models, both Spalart-Allmaras (SA) and $k-\omega$ SST (SST) displayed the best match with experiments. The Spalart-Allmaras (SA) turbulence model was the only one, however, with a hysteresis loop width that was comparable with experiments. Even though numerical results using `snappyHexMesh` gave comparable results to `blockMesh`, much faster simulation times ($4x$) were found with `snappyHexMesh` compared to `blockMesh` and `overSetMesh`. A greater difference was spotted regarding accuracy when comparing `snappyHexMesh` and `overSetMesh`, as the latter exhibited numerical instabilities related to domain interpolation issues. A much better development of vortices was observed from the $2D/2.5D$ comparison. Finally, with respect to the span-wise sensitivity study, the case with $0.3c$ width could be the limiting case from which vortex behavior remains constant in the span-wise direction. Nevertheless, these results are only static results, and pitching airfoil simulations need to be compared with experimental data to corroborate our findings.

Acknowledgements

This work was granted access to the HPC resources of IDRIS, TGCC and CINES under the allocation A0052B06153 made by GENCI (Grand Equipement National de Calcul Intensif). The work reported here was performed using the computing resources of CRIANN - 2017008 (Normandy, France). This work was supported by Adwen Offshore.

Author Contributions: Conceptualisation, J.A. and J.R.; methodology, J.A. and J.R.; software, J.A. and H.H.; validation, J.A. and H.H.; formal analysis, J.A. and H.H. and J.R.; investigation, J.A. and H.H. and J.R.; resources, J.R. and F.X.D. and B.D.; data curation, J.A. and H.H.; writing—original draft preparation, J.A. and H.H. and J.R. and F.X.D.; writing—review and editing, J.A. and H.H. and J.R.; visualisation, J.A. and H.H. and J.R.; supervision, J.R. and F.X.D.; project administration, J.A. and J.R.; funding acquisition, J.R. All authors have read and agreed to the published version of the manuscript.

References

[1] M. O. L. Hansen, *Aerodynamics of wind turbines.* Routledge, 2015, isbn : 9781138775077.

[2] F. R. Menter, M. Kuntz, and R. Langtry, “Ten years of industrial experience with the sst turbulence model,” *Turbulence, heat and mass transfer*, vol. 4, no. 1, pp. 625–632, 2003.

[3] P. Kekina and C. Suvanjumrat, “A comparative study on turbulence models for simulation of flow past naca 0015 airfoil using openfoam,” in *MATEC web of conferences*, vol. 95. EDP Sciences, 2017, p. 12005.

[4] S. R. Allmaras and F. T. Johnson, “Modifications and clarifications for the implementation of the spalart-allmaras turbulence model,” in *Seventh international conference on computational fluid dynamics (ICCFD7)*, vol. 1902. Big Island, HI, 2012.

[5] P. Fuglsang, I. Antoniou, K. S. Dahl, and H. A. Madsen, “Wind tunnel test of the FFA-w3-241, FFA-w3-301 and NACA 63-430 airfoils,” Risoe National Lab., Roskilde (Denmark). Wind Energy and Atmospheric Physics Dept., Tech. Rep., 1998.

[6] M. L. Shur, P. R. Spalart, M. K. Strelets, and A. K. Travin, “A hybrid rans-les approach with delayed-des and wall-modelled les capabilities,” *International Journal of Heat and Fluid Flow*, vol. 29, no. 6, pp. 1638–1649, 2008.

[7] Y. Kim and Z.-T. Xie, “Modelling the effect of freestream turbulence on dynamic stall of wind turbine blades,” *Computers & Fluids*, vol. 129, pp. 53–66, 2016.

[8] W. J. McCroskey, “The phenomenon of dynamic stall.” National Aeronautics and Space Administration, Tech. Rep., 1981.

[9] N. Guillaud, G. Balarac, and E. Goncalvès, “Large eddy simulations on a pitching airfoil: Analysis of the reduced frequency influence,” *Computers & Fluids*, vol. 161, pp. 1–13, 2018.

[10] K. Gharali and D. A. Johnson, “Dynamic stall simulation of a pitching airfoil under unsteady freestream velocity,” *Journal of Fluids and Structures*, vol. 42, pp. 228–244, 2013.

[11] T. C. Corke and F. O. Thomas, “Dynamic stall in pitching airfoils: aerodynamic damping and compressibility effects,” *Annual Review of Fluid Mechanics*, vol. 47, pp. 479–505, 2015.

[12] L. Carr, K. McAlister, and W. McCroskey, “Analysis of the development of dynamic stall based on oscillating airfoil experiments. report tn d-8382,” National Aeronautics and Space Administration, Tech. Rep., 1977.

[13] T. Lee and P. Gerontakos, “Investigation of flow over an oscillating airfoil,” *Journal of Fluid Mechanics*, vol. 512, pp. 313–341, 2004.

[14] S. Wang, D. B. Ingham, L. Ma, M. Pourkashanian, and Z. Tao, “Turbulence modeling of deep dynamic stall at relatively low reynolds number,” *Journal of Fluids and Structures*, vol. 33, pp. 191–209, 2012.

[15] K. M. Almohammadi, D. B. Ingham, L. Ma, and M. Pourkashanian, “2-d-cfd analysis of the effect of trailing edge shape on the performance of a straight-blade vertical axis wind turbine,” *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 228–235, 2014.

[16] J. Tank, L. Smith, and G. Spedding, “On the possibility (or lack thereof) of agreement between experiment and computation of flows over wings at moderate reynolds number,” *Interface focus*, vol. 7, no. 1, p. 20160076, 2017.

[17] F. Geng, I. Kalkman, A. Suiker, and B. Blocken, “Sensitivity analysis of airfoil aerodynamics during pitching motion at a reynolds number of 1.35× 105,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 183, pp. 315–332, 2018.

[18] Y. Kim, “Wind turbine aerodynamics in freestream turbulence,” Ph.D. dissertation, University of Southampton, 2013.

[19] O. Wiki, “Naca0012 by michael alletto — openfoam wiki,,” 2022, [Online; accessed 19-August-2022].

[20] M. Seyednia, S. Vakilipour, and M. Masdari, “Numerical investigation of dynamic stall over a wind turbine pitching airfoil by using openfoam,” *International Journal of Mechanical and Mechatronics Engineering*, vol. 11, no. 8, pp. 1490–1501, 2017.

[21] M. Seyednia, M. Masdari, and S. Vakilipour, “Numerical assessment of a deformable trailing-edge flap on aerodynamic load control of a pitching s809 airfoil using openfoam,” *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 233, no. 7, pp. 890–900, 2019.

[22] E. L. Houghton and P. W. Carpenter, *Aerodynamics for engineering students.* Elsevier, 2003, isbn : 9780750651110.

[23] P. Spalart and S. Allmaras, “A one-equation turbulence model for aerodynamic flows,” in *30th aerospace sciences meeting and exhibit*, ser. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, 1992.

[24] C. Rumsey, B. Smith, and G. Huang, “Description of a website resource for turbulence modeling verification and validation,” in *40th Fluid Dynamics Conference and Exhibit*, ser. Fluid Dynamics and Co-located Conferences, 2010.

[25] D. C. Wilcox, *Turbulence modeling for CFD.* DCW Industries, Inc, 1993, isbn : 9780963605108.

[26] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*, 2nd ed. Pearson Education Ltd, 2007, isbn : 9780131274983.

[27] Y. Egorov and F. Menter, “Development and application of sst-sas turbulence model in the desider project,” in *Advances in Hybrid RANS-LES Modelling.* Springer, 2008, pp. 261–270.

[28] F. Menter, R. Langtry, and S. Völker, “Transition modelling for general purpose cfd codes,” *Flow, turbulence and combustion*, vol. 77, no. 1-4, pp. 277–303, 2006.

[29] R. Mayle and A. Schulz, “Path to predicting bypass transition,” *Journal of Turbomachinery. Transactions of the ASME*, vol. 119, no. 3, pp. 405–411, 1997.

[30] M. V. Morkovin, “On the many faces of transition,” in *Viscous drag reduction.* Springer, 1969, pp. 1–31.

[31] H. M. McIlroy and R. S. Budwig, “The boundary layer over turbine blade models with realistic rough surfaces,” *Journal of turbomachinery*, vol. 129, no. 2, pp. 318–330, 2007.

[32] R. E. Mayle, “The role of laminar-turbulent transition in gas turbine engines,” in *ASME 1991 International Gas Turbine and Aeroengine Congress and Exposition.* American Society of Mechanical Engineers, 1991, pp. V005T17A001–V005T17A001.

[33] A. M. Savill, “One-point closures applied to transition,” in *Turbulence and Transition Modelling: Lecture Notes from the ERCOFTAC/IUTAM Summerschool held in Stockholm, 12–20 June, 1995*, ser. ERCOFTAC Series. Springer Netherlands, 1996, pp. 233–268.

[34] F. R. Menter, “Two-equation eddy-viscosity turbulence models for engineering applications,” *AIAA journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

[35] E. Autio, “Dynamic overset cfd simulation of a pneumatic impact device,” Master’s thesis, Tampere University of Technology, 2018.

[36] G. Zagaris, M. T. Campbell, D. J. Bodony, E. Shaffer, and M. D. Brandyberry, “A toolkit for parallel overset grid assembly targeting large-scale moving body aerodynamic simulations,” in *Proceedings of the 19th international meshing roundtable.* Springer, 2010, pp. 385–401.

[37] F. Liu, “A thorough description of how wall functions are implemented in openfoam.” *Proceedings of CFD with OpenSource Software*, 2016.

[38] F. Menter and T. Esch, “Elements of industrial heat transfer predictions,” in *16th Brazilian Congress of Mechanical Engineering (COBEM)*, vol. 109, 2001, pp. 117–127.

[39] F. Barnaud, P. Bénard, G. Lartigue, V. Moureau, and P. Deglaire, “Flow around thick airfoils at very high reynolds number. stall and dynamic stall applications,” in *Direct and Large-Eddy Simulation XI*, M. V. Salvetti, V. Armenio, J. Fröhlich, B. J. Geurts, and H. Kuerten, Eds. Cham: Springer International Publishing, 2019, pp. 359–365.

[40] Y. Qian, Y. Zhang, Y. Sun, and T. Wang, “Numerical investigations of the flow control effect on a thick wind turbine airfoil using deformable trailing edge flaps,” *Energy*, vol. 265, p. 126327, 2023.

[41] M. J. Lutton, “Comparison of c-and o-grid generation methods using a naca 0012 airfoil,” Air Force Inst. of Tech Wright-Patterson AFB OH School of Engineering, Tech. Rep., 1989.

[42] M. S. Gritskevich, A. V. Garbaruk, J. Schütze, and F. R. Menter, “Development of ddes and iddes formulations for the k-ω shear stress transport model,” *Flow, turbulence and combustion*, vol. 88, no. 3, pp. 431–449, 2012.