XL CILAMCE
IBERO-LATIN AMERICAN
CONGRESS ON
COMPUTATIONAL
METHODS IN
ENGINEERING

NOVEMBER
11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

AERODYNAMIC ANALYSIS OF PROPELLERS BY COMPUTATIONAL FLUID DYNAMICS

Eric Vargas Loureiro
Nicolas Lima Oliveira
eric.vargas@engenharia.ufjf.br
nicolas_limaoliveira@yahoo.com.br
Graduate Program in Computational Modeling, Federal University of Juiz de Fora
Campus Universitário, 36036 330, Juiz de Fora, MG, Brazil
Patricia Habib Hallak
patricia.hallak@ufjf.edu.br
Graduate Program in Computational Modeling, Federal University of Juiz de Fora
Campus Universitário, 36036 330, Juiz de Fora, MG, Brazil

Abstract. Propellers are widely used in our day-to-day for various purposes. Aerodynamic analyzes are necessary to evaluate the thrust generated and the power that the propeller is capable of absorbing or delivering. Among the available tools for this purpose, we highlight the computational fluid dynamics (CFD). In general, the main concerns for the use of CFD lie in the complexity of its geometry, which is often not provided by the manufacturer, the generation of the mesh, and the use of appropriate numerical techniques to solve the problem. Regarding this last observation, the computational tools must be able to accommodate the movement of the rotating domain of the propeller to the computational domain. Besides, due to the flow characteristics, turbulence models must be incorporated to represent the entire behavior of the system adequately. This study proposes computational modeling and aerodynamic analysis of a fan by CFD using the Ansys commercial program (academic version) and OpenFOAM open source. The computational domain is divided into a rotating and stationary. At interfaces between both, there is no agreement between the nodes and the elements from one mesh to the other, but the numerical technique ensures the correct transfer of information. Also incorporated into the codes are the turbulence models widely reported in the literature. The proposed methodology consists of comparing the responses in terms of thrust and torque for some operating conditions between both programs, adopting the same boundary conditions and similar numerical techniques, and verifying the quality of the results.

Keywords: propeller, CFD, turbulence, MRF, AMI

1 Introduction

Propellers are widely used in our day-to-day for various purposes, ranging from household equipment such as fans to more complex systems such as wind turbine propellers and aircraft propellers.

In general, aerodynamic analyzes are necessary to evaluate the thrust generated and the power that the propeller is capable of absorbing or delivering. Among the available tools for this purpose, we highlight the computational fluid dynamics (CFD). CFD, although requiring a high computational cost, has as its main attraction its versatility when it is desired specific analyzes in different operating conditions of the equipment.

This study proposes the computational modeling and aerodynamic analysis of a fan by CFD using the prominent Ansys CFX commercial program (academic version and hereafter CFX) and OpenFOAM (hereafter OF) open source. The idea is to compare the results in terms of aerodynamic coefficients, such as thrust and torque, obtained using both tools. OF code was chosen because it is a free source tool, released by OpenCFD Ltd in 2004, developed for a broad range of fluid dynamics applications and quickly became very popular in industrial engineering as well as in academic research; there are no limitation for parallel computing and no black boxes compare to commercial solvers like Ansys [1]. Nevertheless, OF has some disadvantages as well. According to Lysenko et al. [1] , the most crucial is the absence of quality certification, and as a consequence, the lack of high-quality documentation and references. Thus, the problem of OF validation and verification becomes more principal and fundamental compared to other commercial CFD codes.

In general, the main concerns for the use of CFD in turbomachinery lie in the complexity of its geometry, which is often not provided by the manufacturer, the generation of the mesh, and the use of appropriate numerical techniques to solve the problem. Regarding this last observation, the computational tools must be able to accommodate the movement of the rotating domain of the propeller to the computational domain. To accomplish this, a multireference frame was implemented in the CFD codes [2–5]. In this methodology, the computational domain is divided into a rotating domain and a stationary domain. At interfaces between both, there is no agreement between the nodes and the elements from one mesh to the other, but the numerical technique ensures the correct transfer of information between the two domains [6].

In addition, due to the flow characteristics, turbulence models must be considered to represent the entire behavior of the system adequately. Both CFD codes have incorporated the most traditional turbulence models, which are widely used with success in this kind of simulation [1, 4, 5, 7, 8].

The proposed methodology consists of comparing the responses in terms of thrust and torque for some operating conditions between both programs, adopting the same boundary conditions and similar numerical techniques, and verifying the quality of the results.

This paper is organized as follow. In the next section a review of the main equations of the continuum mechanics, including the theory behind rotational frames and turbulence models, is presented. The case set up, and the computational model is detailed in Section 3. Section 4 is dedicated to the description of the numerical model adopted. Results are presented in Section 5. Finally, Section 6 sum up the main results of the research.

2 Theoretical background

2.1 Continuity and momentum equations

The flow is predicted by enforcing the conservation of mass and momentum, Eq. 1 and Eq. 2, for viscous and incompressible flow in an inertial (non-accelerating) reference frame.

- Mass conservation equation

$$\nabla.\vec{v} = 0 \tag{1}$$

- Momentum conservation equation

$$\rho\frac{\partial\vec{v}}{\partial t}+\rho\nabla.\vec{v}\vec{v}=-\nabla.p+\nabla.\left(\bar{\bar{\tau}}\right)+\rho\vec{g}+\vec{F} \tag{2}$$

where ρ is the fluid density, p is the static pressure, $\vec{v}$ the velocity field, $\bar{\bar{\tau}}=\mu\left[\left(\nabla\vec{v}+\nabla\vec{v}^T\right)\right]$ is the stress tensor, μ is the viscosity, $\rho\vec{g}$ and $\vec{F}$ are the gravitational body force and external body forces.

2.2 Rotational motion

There are a variety of engineering problems involving rotational parts. The way found to treat this kind of simulation is to partitioning the entire computational domain into a moving frame and an inertial frame. The moving (rotating) frame is a reference frame that moves with the body which can translate and/or rotate. When a reference frame is either fixed or moving with a constant velocity, it is an inertial frame. Both domain are depicted in Figure 1, where rotating zone is the moving frame and stationary zone is the inertial frame.

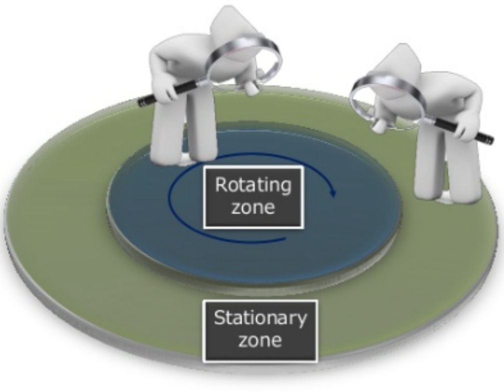


Figure 1. Moving reference frame.

CFD codes provide different techniques to simulate a rotating propeller with different physical and computational properties. MRF (The Multiple Reference Frame Model) is, perhaps, the easiest way but is a computationally efficient technique to model a rotating frame of reference. The sliding grid techniques provide the more complex way to simulate the propeller and its surrounding region, rotating and interpolate on interface for transient effects. AMI, (Arbitrary Mesh Interface), is a sliding grid implementation [9]. Both techniques are summarized in this section.

Multiple reference frame (MRF)

The MRF model is, perhaps, the simplest of the two approaches for multiple zones. It is a steady-state approximation in which individual cell zones can be assigned different rotational and/or translational speeds. The flow in each moving cell zone is solved using the moving reference frame equations [10].

Consider a coordinate system which is rotating steadily with angular velocity $\vec{\omega}$ relative to a stationary (inertial) reference frame, as illustrated in Figure 2. The origin of the rotating system is located by a position vector $\vec{r}_0$. The computational domain for the CFD problem is defined with respect to the rotating frame such that an arbitrary point in the CFD domain is located by a position vector $\vec{r}$ from the origin of the rotating frame.

The fluid velocities can be transformed from the stationary frame to the rotating frame using the following relation:

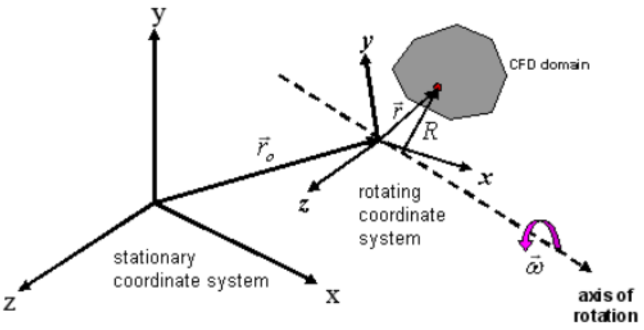


Figure 2. Stationary and Rotating Reference Frames.

$$\vec{v}_r = \vec{v} - \vec{u}_r \tag{3}$$

$$\vec{u}_r = \vec{\omega} \times \vec{r} \tag{4}$$

In the above, $\vec{v}_r$ is the relative velocity (the velocity viewed from the rotating frame), $\vec{v}$ is the absolute velocity (the velocity viewed from the stationary frame), and $\vec{u}_r$ is the ”whirl” velocity (the velocity due to the moving frame).

The original Navier-Stokes equations, Eq. 1 and Eq. 2, are modified by introducing Eq. 3 and Eq. 4. In an absolute velocity framework the governing equations of fluid flow for a steadily moving frame can be written as follows:

- Conservation of mass:

$$\nabla.\vec{v}_r = 0 \tag{5}$$

- Conservation of momentum::

$$\rho\frac{\partial \vec{v}}{\partial t} + \rho\nabla.\vec{v}_r\vec{v} + \rho\left[\vec{\omega} \times \vec{v}\right] = -\nabla p + \nabla\bar{\bar{\tau}} + \rho\vec{g} + \vec{F} \tag{6}$$

The MRF method does not rotate the mesh physically but adds rotational forces like Coriolis or centrifugal forces to selected cells of the mesh. Those forces are represented by the term $\rho\left[\vec{\omega} \times \vec{v}\right]$ in above equations.

It should be noted that the MRF approach does not account for the relative motion of a moving zone with respect to adjacent zones (which may be moving or stationary); the mesh remains fixed for the computation. This is analogous to freezing the motion of the moving part in a specific position and observing the instantaneous flowfield with the rotor in that position. Hence, the MRF is often referred to as the ”frozen rotor approach.” [10].

While the MRF approach is clearly an approximation, it can provide a reasonable model of the flow for many applications. For example, the MRF model can be used for turbomachinery applications in which rotor-stator interaction is relatively weak, and the flow is relatively uncomplicated at the interface between the moving and stationary zones [10].

When the absolute velocity formulation is used, the governing equations in each subdomain are written with respect to that subdomain’s reference frame, but the velocities are stored in the absolute frame. Therefore, no special transformation is required at the interface between two subdomains. Again, scalar quantities are determined locally from adjacent cells [10].

Sliding mesh

The sliding mesh is a computational unsteady technique to model the CFD problems where the interaction between stator and rotor is strong and more accurate computation is desired as compared with previous model [9].

Arbitrary Mesh Interface (AMI) is a sliding mesh technique where the rotational zone around the object moves in every time step and the values lying on the interface are interpolated to update the mesh in every time step. It enables to simulate across disconnected, non-conformal patches but adjacent mesh domains that have been developed based on the algorithm described in [11].

The AMI approach is based on the creation of a "supermesh". This supermesh is defined by the intersection elements of the two-consecutive time step meshes. This approach applied for the interpolation of adaptive meshes, tends to reduce continuity errors, improving its numerical efficiency [12]. With the creation of the supermesh the problems of intersection and overlapping areas have a more robust procedure, which optimizes the interpolation accuracy and consequently improves the conservation of properties [11]. Figure 3, extrated from the original reference [11], ilustrate the idea of this supermesh.

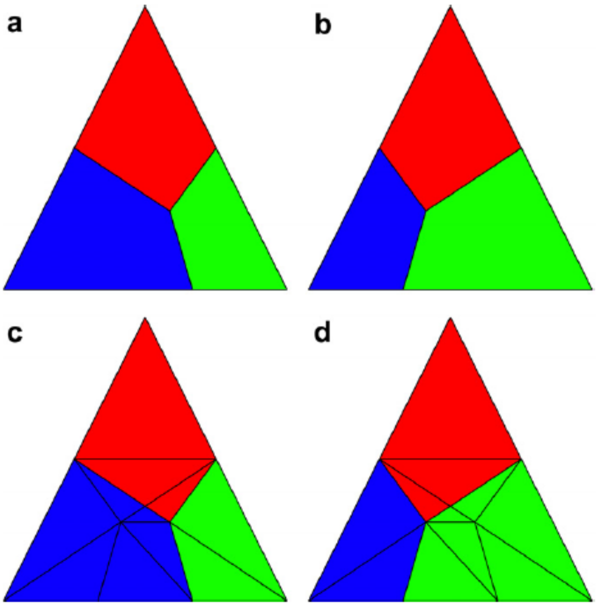


Figure 3. (a) and (b) Two quadrilateral meshes. (c) A triangular supermesh of (a) and (b), coloured to show the elements of (a). (d) The same supermesh of (a) and (b), coloured to show the elements of (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article. Extrated from [11])

The exchange of values of fluid properties takes place across the outer boundary of inner domain and the inner boundary of outer domain using a conservative interpolation method by local Galerkin projection proposed in [11]. AMI weights define the contribution as a fraction of the intersecting areas. For each face, the sum of the weights should equal 1. Conservation errors are introduced as the sum of weights deviates from 1 where the patch geometries are not well matched. Though these errors are localized and do not cause the method to fail, users can specify a lower limit to the sum of weights. When the sum of weights goes below this limit, the interpolation across the AMI boundary cells stops and a Dirichlet boundary condition is imposed to particular patch faces [13].

2.3 Turbulence model

Turbulence is normally present in real flow, and to accurately calculate the turbulent behavior would be very time-consuming and costly, hence turbulence models are often used instead. The most known and popular turbulence models are classified as Reynolds averaged Navier- Stokes (RANS). RANS uses the Reynolds decomposition which says that the variable of interest consists of an average part and a fluctuating part. Thus, the time average of the variable provides the main properties [14].

There are several models within RANS and the most popular family is the two-equation models, namely $k - \varepsilon$ and $k - \omega$ models. In those models, k represents the kinetic energy, ε their dissation and ω the specific rate of dissipation. They all rely on Boussinesq's approximation from 1877 where the stress tensor is modeled in the viscous term of Navier-Stokes Eq. 2. Readers can find in the original references the description of these most popular models [14, 15].

This section describe, briefly, the two equation models k$-\varepsilon$ and k $-\omega$ SST model, where SST stands for Shear Stress Transport [16]. These were the models adopted in present research, and corroborate with previous research [9, 12, 17–19].

It is important to report the main difference between k$-\varepsilon$ and k $-\omega$ SST model. The starting point for the development of the SST model was the need for the accurate prediction of flows with strong adverse pressure gradients and separation [16]. Over decades, the available turbulence models had consistently failed to compute these flow. In particular, the otherwise popular k-ε model model was not able to capture the proper behaviour of turbulent boundary layers up to separation [14].

The k $-\omega$ model is substantially more accurate than k$-\varepsilon$ in the near wall layers, and has therefore been successful used for flows with moderate adverse pressure gradients, but failes for flows with pressure induced separation [16]. In addition the ω-equation shows a strong sensitivity to the values of ω in the freestream outside the boundary layer [20]. The freestream sensitivity has largely prevented the ω-equation from replacing the ε-equation as the standard scale-equation in turbulence modelling, despite its superior performance in the near wall region.

This was one of the main motivations for the development of the blend method k-ωSST. Near walls the $k - \omega$SST model uses the $k - \omega$ approach, in regions far from the wall it uses the Standard $k - \varepsilon$ turbulence model. Switching between those models is controlled by a blending function.

Turbulence model k$-\varepsilon$

The k$-\varepsilon$ model is based on the eddy viscosity concept

$$\nu_t = C_\mu \rho \frac{k^2}{\varepsilon} \tag{7}$$

where C_μ is a constant, usually equal to 0.09.

The values of k and ε come directly from the differential transport equations for the turbulence kinetic energy Eq. 8 and turbulence dissipation rate Eq. 9:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j k) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon + P_{kb} \tag{8}$$

$$\frac{\partial \rho\varepsilon}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j \varepsilon) = \frac{\partial}{\partial}\partial x_j \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{\varepsilon}{k}\left(C_{\varepsilon 1}P_k - C_{\varepsilon 2}\rho\varepsilon + C_{\varepsilon 1}P_{\varepsilon b}\right) \tag{9}$$

where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k and σ_ε are constants, usually equal to 1.44, 1.92, 1.0 and 1.3, respectivelly. P_{kb} and $P_{\varepsilon b}$ represent the influence of the bouyancy forces, P_k is the turbulence production due to viscous force.

Turbulence model k$-\omega$ SST

It was designed to give a highly accurate predictions of the onset and the amount of flow separation under adverse pressure gradients by the inclusion of transport effects into the formulation of the eddy-viscosity [10]. On the other hand, the model can be considered as more complex and therefore it requires more computational time than simple two equation models [19].

The relation between the turbulent eddy viscosity ν_t with the turbulent kinetic energy k and the specific dissipation rate ω is:

$$\nu_t = \frac{\alpha_1 k}{\mathtt{max}(a_1\omega, SF_2)} \tag{10}$$

where α_1 and a_1 are constants, F_2 a blending function, S is the strain tensor.

The transport equations for the turbulent kinetic energy k and for the specific dissipation rate ω is characterized by the transport equation of the $k-\omega$ turbulence model [14]. Note that for ω equation, the model uses the blending function F_1 to switch between $k-\omega$ to $k-\varepsilon$ models.

$$\frac{\partial k}{\partial t} + u_j\frac{\partial k}{\partial x_j} = \tau_{ij}\frac{\partial u_j}{\partial x_j} - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\nu + \sigma^*\nu_t)\,\frac{\partial k}{\partial x_j}\right] \tag{11}$$

$$\frac{\partial \omega}{\partial t} + u_j\frac{\partial \omega}{\partial x_j} = \alpha S^2 + 2(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma^*\nu_t)\,\frac{\partial k}{\partial x_j}\right] \tag{12}$$

More details about the blending functions F_1, F_2 and the model constants can be found in [16, 21, 22]. One of the advantages of the formulation is the near wall treatment for low-Reynolds number computations where it is more accurate and more robust.

3 Case set-up and computational model

3.1 Geometric parameter and operation condition

The case set-up is a six blade household fan, whose geometric model is showed in Figure 4.

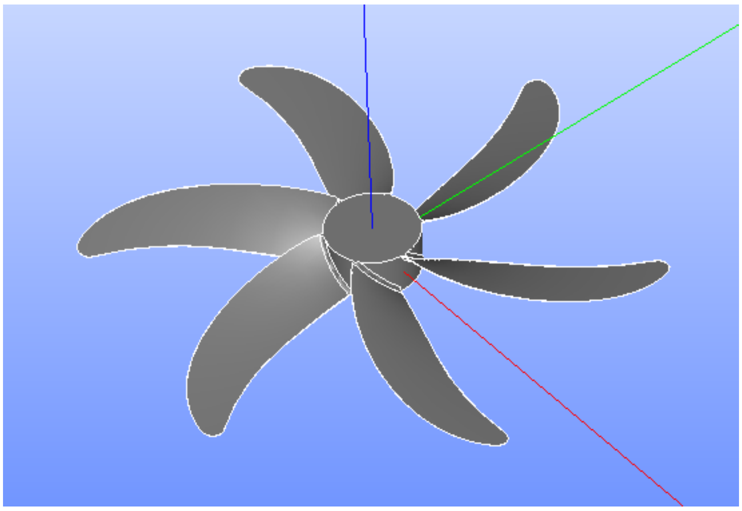


Figure 4. Case set-up.

The most important geometric features and operation conditions are presented in Tables 1 2. For this paper, the Reynolds number was defined by the rotational speed ω and chord c at the 75% blade station r. The Reynolds number is defined as:

$$Re = \frac{2\pi\omega(0.75r)c}{\nu} \tag{13}$$

Table 1. Geometric parameters.

Specification	Value
Rotor speeds (rpm)	1170
	1270
	1370
Rotor diameter D (m)	0,44m
Hub flange diameter (m)	0.066m
Blade length (m)	0,4
Tild angle (degrees)	73,3

Table 2. Operation condition.

Rotor speeds (rpm)	Reynolds numer
1170	51.749
1270	56.172
1370	60.595

One important aspect to be notice here is the Reynolds number values, since this parameter indicates the flow regime. Flows with Reynolds number below 10^5 is categorized as low turbulence level [23]. In this regime, there is a predominance of a strong and adverse gradient in the pressure field [23].

3.2 Computational domain and boundary condition

Application and analysis of the moving mesh algorithm AMI (OpenFOAM) and frozen rotor approach (Ansys CFX) were conducted under static condition. The numerical results in terms of thrust and torque were compared.

Figure 5 presents the geometric domain, similar for both simulations. The inner cylinder is the boundary of the non stationary domain (rotor), and enclosures the propeller surface. The outer cylinder is the external boundary of the stationary domain (stator). Their dimensions are, considering D as the diameter of the propeller:

- internal diameter $D_i = 1,1D$;
- internal height $h = 0,4D$;
- external diameter $D_e = 8D$;
- external height $H = 8D$.

In the inlet boundary condition, the velocity was precribed as an static case. In the external wall and in the propeller surface, a no-slip boundary condition were set. In the outlet the Dirichlet boundary condition was precribed for pressure. The intensity level I was equal to 1%.

3.3 Mesh information

In Ansys CFX the total mesh has 80769 nodes and 428728 elements. The front and back face at the interfaces surfaces has 157 nodes and 272 elements in the stationay domain and 336 nodes and 618 elements in the rotationg face. In the lateral shell that delimits both domains, there are 260 nodes and

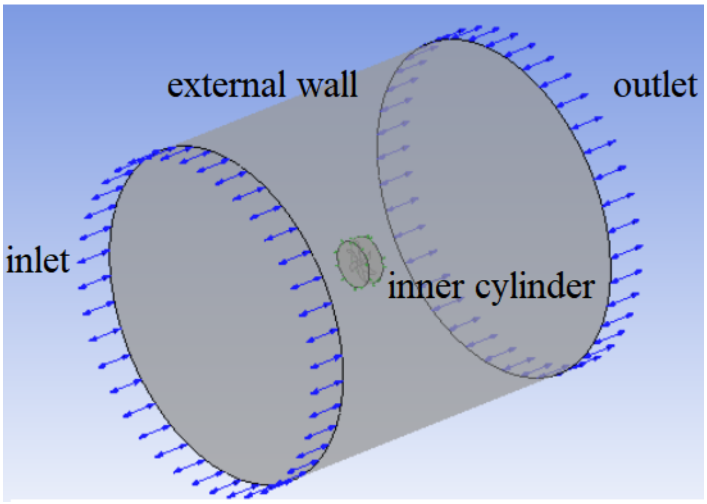


Figure 5. Computation domain.

440 elements in stationary side and 1210 nodes and 2305 elements in the rotating side. The left hand side of Figure 6 shows the entire mesh. In the right side a detail of the mesh in the rotating part. Details about meshes in the interface are shown in Figures 7.

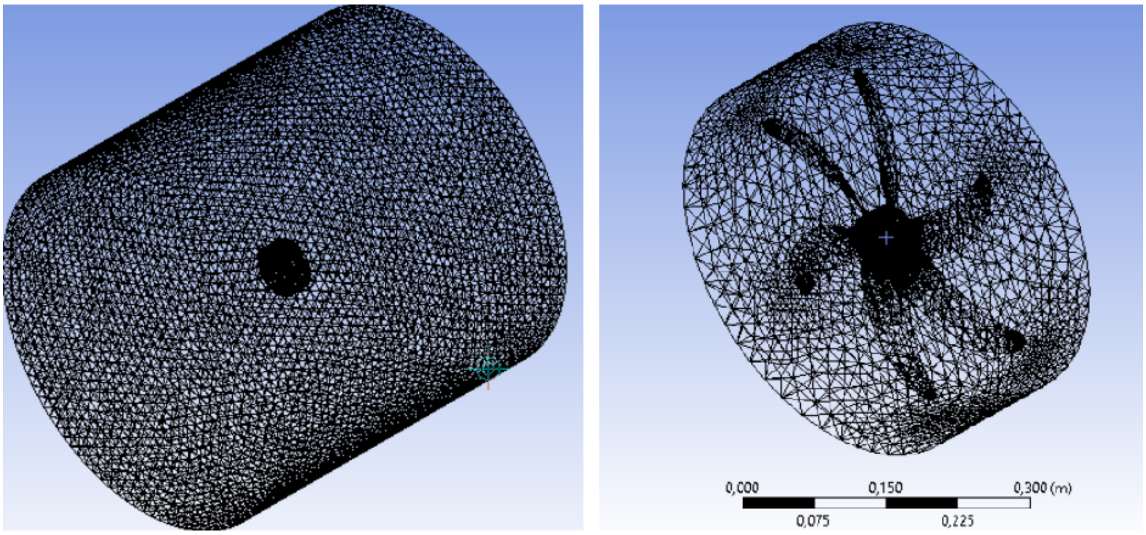


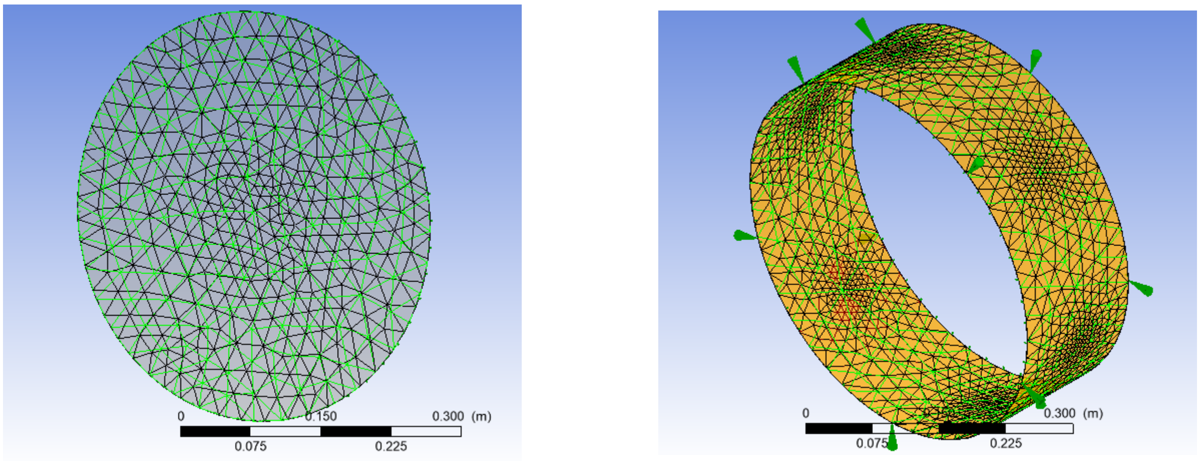
Figure 6. Mesh in Ansys model.

In OF solver the total mesh has 89641 nodes and 474382 elements. The left hand side of Figure 8 shows the entire mesh. In the right side a detail of the mesh in the rotating part, with its 44484 nodes and 254591 elements.

The inlet and outlet at the interface between the two domains have 996 and 923 elements respectively on the rotating surface; and 1056 and 990 elements on the stationary surface. On the lateral surface that delimits the domains, there are 1589 elements in the rotating interface and 1542 in the stationary interface. Although these values are close to each other, the meshes are not necessarily coincident. However, the AMI approach permit a better information exchange between each meshes in a consistent way. Details about meshes in the interface are shown in Figures 9.

4 Solvers

The purpose of this section is to briefly describe some of the numerical methods employed in the research.



(a) Rotor inlet interface.

(b) Rotor wall interface.

Figure 7. Detail of mesh interfaces in Ansys.

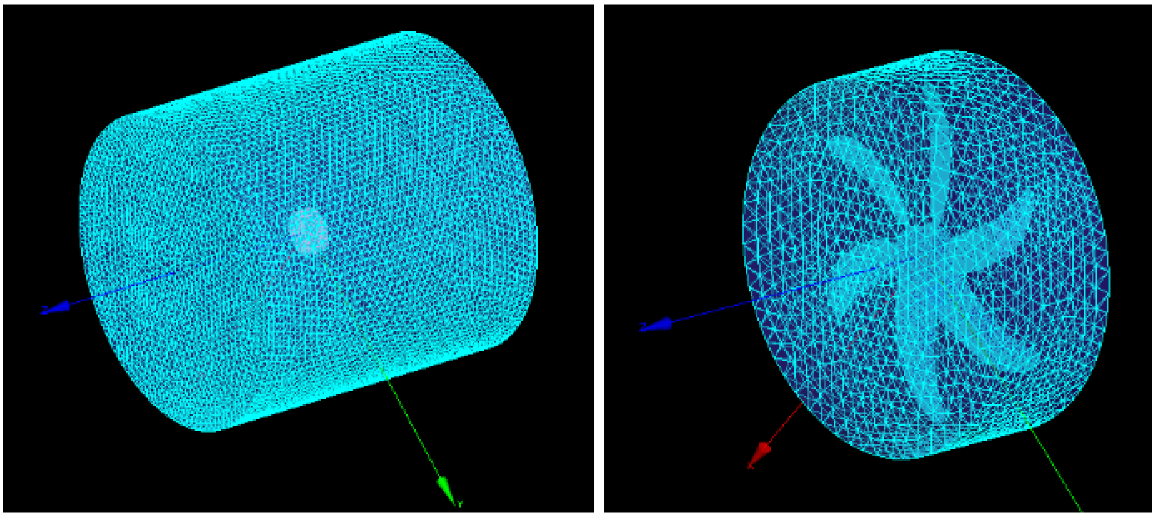


Figure 8. Mesh in OpenFOAM model.

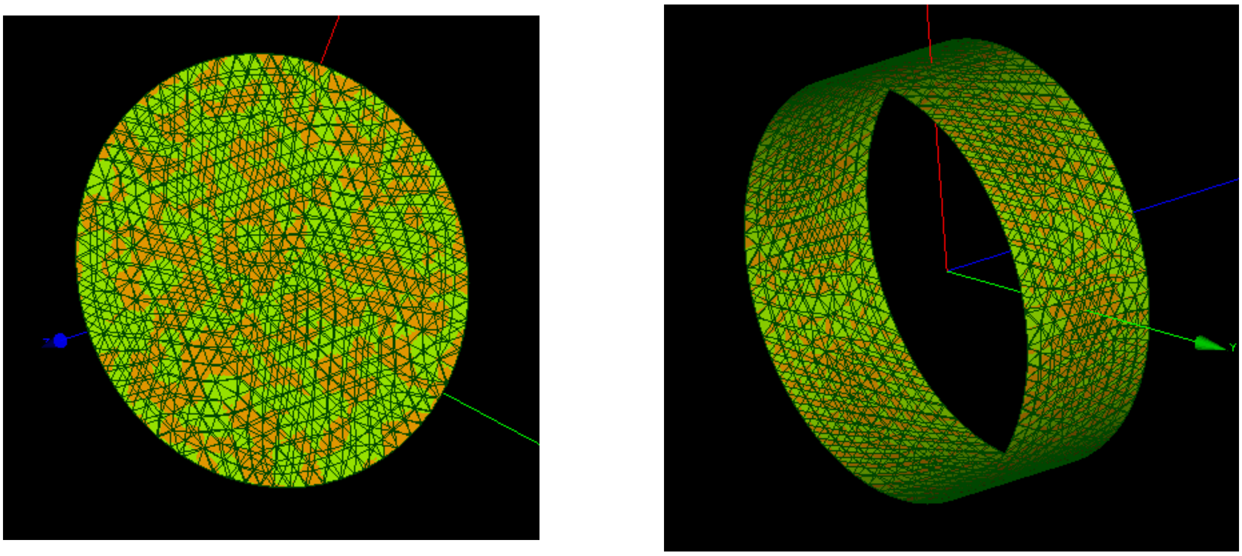
4.1 Discretization

OpenFOAM solver is based on the finite volume method, which involves discretizing the spatial domain using a mesh. The mesh is used to construct finite volumes, which are used to conserve relevant quantities such as mass, momentum, and so on. Solution fields and other properties are stored at the mesh nodes.

Ansys CFX solver uses an element-based finite volume method, which first involves the finite volume discretizing. However, to evaluate many of the terms, the solution field or solution gradients must be approximated at integration points. ANSYS CFX uses finite-element shape functions to perform these approximations.

4.2 Time schemes

In OpenFoam, time scheme solvers are generally configured to simulate either transient or steady-state. Changing the time scheme from one which is steady-state to transient, or visa versa, does not affect the fundamental nature of the solver and so fails to achieve its purpose, yielding a nonsensical solution. Some possibilities in the software are Euler, steadyState, backward, CrankNicolson and localEuler, each with its own characteristic. In this work, Euler was used for cases where the turbulence model was k-ωSST and CrankNicolson for cases where the turbulence model was k-ε. Here is some information about



(a) Rotor inlet interface.

(b) Rotor wall interface.

Figure 9. Detail of mesh interfaces in OpenFOAM.

them:

- Euler: transient, first order implicit, bounded;
- CrankNicolson: transient, second order implicit, bounded; requires an off-centeringcoefficient ψ where:

$$\psi = \begin{cases} 1 & \textit{corresponds to pure CrankNicolson,} \\ 0 & \textit{corresponds to Euler} \end{cases} \tag{14}$$

Generally ψ = 0.9 is used to bound/stabilise the scheme for practical engineering problems.

Any second time derivative $(\partial^2/\partial t^2)$ terms are specified in the *d2dt2Schemes* sub-dictionary. Only the Euler scheme is available for *d2dt2Schemes*.

An adaptive scheme was used, that is, a scheme that adjusts the time step at each instant to always work within the limits of a predetermined Courant. Even though the scheme is adaptive the time step throughout the simulation always kept around (1×10^{-5}). Because it is a costly simulation, a cluster was used to simulate all 6 cases. Each case ran on one machine, all of the same specs, Intel Xeon model E5620 CPU with 16 hyperthreading cores, 250GB RAM and Linux Ubuntu operating system.

4.3 Convective schemes

The advection schemes implemented in ANSYS CFX can be cast in the form:

$$\phi_{ip} = \phi_{up} + \theta(\nabla\phi.\delta)_{up} \tag{15}$$

where ϕ_{up} is the value of the upwind node, $\vec{\delta}$ is the vector from the upwind node to the ip.

Ansys CFX uses for the treatment of the convective term the high resolution algorithms by Barth and Jespersen [24]. This scheme uses a nonlinear adaptive choice the value for θ on each face. Is is based on border information and information of the central node itself (it is considered the minimum value of all integration point values around the node).

The treatment of advective terms is one of the major challenges in CFD numerics and so the options are more extensive. In OpenFoam the keyword identifier for the advective terms are usually of the form div(phi,...), where phi generally denotes the (volumetric) flux of velocity on the cell faces for constant-density flows and the mass flux for compressible flows, e.g. div(phi,U) for the advection of velocity, div(phi,e) for the advection of internal energy, div(phi,k) for turbulent kinetic energy, etc. The schemes are all based on Gauss integration, using the flux phi and the advected field being interpolated to the cell faces by one of a selection of schemes, e.g. linear, linearUpwind, etc. There is abounded variant of the discretisation. In this work, the following methods were used:

- linear: second order, unbounded;
- linearUpwind: second order, upwind-biased, unbounded (but much less so than linear), that requires discretisation of the velocity gradient to be specified;
- upwind: first-order bounded, generally too inaccurate to be recommended.

4.4 Pressure velocity solver

The discretization of the momentum equation Eq. 2 becames:

$$a_p u_p = \sum_{nb} a_{nb} u_{nb} + \sum pA.\hat{i} + S \tag{16}$$

where the subscript nb referes to neighbouof the r cell p, a_p and a_{nb} are linearized coefficients for velocity field u, S the soucer term, A the area of the face.

There are important issues with respect to the storage of pressure and the discretization of the pressure gradient term, since both velocity and pressure field are unknown.

To accomplish this, OpenfFOAM uses the segregated PIMPLE algorithm. This algorithm is a combination of PISO (Pressure Implicit with Splitting of Operator) and SIMPLE [25] (Semi-Implicit Method for Pressure-Linked Equations). All these algorithms are iterative solvers but PISO and PIMPLE are both used for transient cases whereas SIMPLE is used for steady-state cases. The mean features of this algorithm is the segregated meshes for store pressure and velocities.

Ansys CFX uses a co-located scheme, whereby pressure and velocity are both stored at cell centers. However, Eq. 16 requires the value of the pressure at the face between cells. Therefore, an interpolation scheme is required to compute the face values of pressure from the cell values. This the coupled approach offers some advantages over the segregated approach. The coupled scheme obtains a robust and efficient single phase implementation for steady-state flows, with superior performance compared to the segregated solution schemes.The coupled algorithm solves the momentum and pressure-based continuity equations together. The full implicit coupling is achieved through an implicit discretization of pressure gradient terms in the momentum equations, and an implicit discretization of the face mass flux, including the Rhie-Chow pressure dissipation terms, similar to procedure proposed by Rhie and Chow [26].

5 Results

This section presents the results and is divided into two parts. In the first, a comparison between the results obtained with k-ε and -k-ω turbulence model is conducted. Next, a comparative analysis of the performance of models for rotating domains is presented.

5.1 Turbulence model

Tables 3 and 4 present the thrust and torque results obtained with Ansys CFX and OF codes, respectivelly. The turbulence model is also reported.

As described in Subsection 2.3, the k-ωSST model seems to be more robust and accurate. Thus, in the last two columns, namely *Difference Thrust* $\%$ and *Difference Torque* $\%$, is the percentage difference from k-ε to k-ωSST. The minus signal means that values predict from ε model is higher than from ωSST model.

Table 3. Thrust(T) and torque (Q) - Results from Ansys CFX.

Rotation speed (rpm)	Model	T(N)	Q(Nm)	Model	T(N)	Q(Nm)	Difference Thrust $\%$	Difference Torque $\%$
1170	k$-\varepsilon$	1.53	0.063	k-ωSST	1.55	0.058	1,29	-8.62
1270		1.80	0.074		1.83	0.068	1.64	-8.82
1370		2.09	0.086		2.13	0.079	1.88	-8.86

Table 4. Thrust(T) and torque (Q) - Results from OF.

Rotation speed (rpm)	Model	T(N)	Q(Nm)	Model	T(N)	Q(Nm)	Difference Thrust $\%$	Difference Torque $\%$
1170	k$-\varepsilon$	1.23	0.044	k-ωSST	1.31	0.0478	6.10	6.38
1270		1.61	0.059		1.54	0.056	-4.54	-5.35
1370		1.97	0.071		1.79	0.065	-10.05	-9.23

Before conduct the analysis about these results, readers should keep in mind that rotational solvers employed in each analyze, which generated the results presented in Tables 3 and 4, were quite different. Ansys employed the frozen rotor techniques, without considering transient effects. On the other hand, OF employed a transient analysis. The difference between both techniques is investigated in subsection 5.2.

Table 3, Ansys CFX, shows a slight variation in thrust values. The difference in torque reach almost -9$\%$. This behavior can be attributed to the fact that ε fails to predict the detachment point and thus fails to predict the distribution of the resulting force along the blade.

On the other hand, the thrust difference shown in Table 4 is more accentuated and reach a maximum of -10$\%$ in higher rotation speed. Again, the difference in both results is attributed to the weakness of ε model to predict the distribution pressure around the blades.

In most cases, ε model overestimate the aerodynamic parameters. As reported in Section 2.3, k-ω SST model was developed to solve flows with strong adverse pressure gradients, since k-ε model was not able to capture the proper behavior of turbulent boundary layers up to separation [14]. This is the case reported here, where the Reynolds number stated in Table 2 is below 10^5, indicating a predominance of a strong and adverse gradient in the pressure field [23].

5.2 Comparison between frozen rotor and AMI techniques.

Table 5 presents a comparison between the results obtained with frozen rotor technique (Ansys CFX) and AMI tecnhique (OF), for k-ε turbulence model. In the same way, Table 6 portraits the results for k-ω turbulence model.

Since AMI incorporates the transient effects, it was used as a reference in comparisons between both techniques. In the last two columns, namely *Difference Thrust* $\%$ and *Difference Torque* $\%$, is the percentage difference from frozen rotor to AMI. The minus signal means that values predict with frozen rotor technique is higher than from AMI.

The difference observed for k-ε in Table 5 is the highest observed among all the results, and reach the maximum values of -24.39% for thrust comparison and -43.18% for torque comparison using k-ε analysis. These values were observed for the lowest value of Reynolds number (minimum rotional

Table 5. Thrust(T) and torque (Q) - Results with k-ε model.

Rotation speed (rpm)	Code	T(N)	Q(Nm)	Code	T(N)	Q(Nm)	Difference Thrust %	Difference Torque %
1170	Ansys CFX	1.53	0.063	OF	1.23	0.044	-24.39	-43.18
1270		1.80	0.074		1.61	0.059	-11.80	-25.42
1370		2.09	0.086		1.97	0.071	-6.09	-21.13

Table 6. Thrust(T) and torque (Q) - Results with k-ωSST.

Rotation speed (rpm)	Code	T(N)	Q(Nm)	Code	T(N)	Q(Nm)	Difference Thrust %	Difference Torque %
1170	Ansys CFX	1.55	0.058	OF	1.31	0.0478	-18.32	-23.40
1270		1.83	0.068		1.54	0.056	-18.83	-21.43
1370		2.13	0.079		1.79	0.065	-19.00	-21.54

speed). We reinforce the weakness of the k-ε model in this situation.

Besides the difference observed related to turbulence models, analyzed in Subsection 5.1, it is possible to highlight the great difference between frozen rotor and AMI techniques. It should be important to remark that AMI incorporates transient effects, and can be understood as the technique with higher fidelity and accuracy. Supported by this observation, in this example, frozen rotor overestimates the aerodynamic parameter. Transient effects were determinant in order to calculate the thrust and torque and should be taken into account in this kind of analysis.

6 Conclusions

In this work aerodynamic analyses of a household fan were performed using computational fluid dynamic. Two techniques for rotating meshes were investigated, the frozen rotor approach and the AMI approach. While the first kept the motion frozen, the second one took into consideration transient effects. Also, two turbulence models were also investigated.

The k-ε turbulence model overpredict the aerodynamic parameter since it was not able to predict the gradient pressure as k-ω did. Regarding the transient effects, it was possible to verify that transient analysis underestimate the aerodynamic parameters and is determinant in order to assess aerodynamic performance of propellers.

Acknowledgements

The authors would like thank to Graduate Program in Computational Modeling (PGMC), to Federal University of Juiz de Fora (UFJF) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) for the support.

References

[1] Lysenko, D. A., Ertesvåg, I. S., & Rian, K. E., 2013. Modeling of turbulent separated flows using openfoam. *Computers & Fluids*, vol. 80, pp. 408–422.

[2] Park, H., 2016. Advanced turboprop composite propeller design and analysis using fluid–structure interaction method. *Composites Part B: Engineering*, vol. 97, pp. 111–119.

[3] Kutty, H. & Rajendran, P., 2017. 3d cfd simulation and experimental validation of small apc slow flyer propeller blade. *Aerospace*, vol. 4, n. 1, pp. 10.

[4] Sodja, J., De Breuker, R., Nozak, D., Drazumeric, R., & Marzocca, P., 2018. Assessment of low-fidelity fluid–structure interaction model for flexible propeller blades. *Aerospace Science and Technology*, vol. 78, pp. 71–88.

[5] Król, P. & Tesch, K., 2018. Experimental and numerical validation of the improved vortex method applied to cp745 marine propeller model. *Polish Maritime Research*, vol. 25, n. 2, pp. 57–65.

[6] Jasak, H. & Beaudoin, M., 2011. Openfoam turbo tools: From general purpose cfd to turbomachinery simulations. In *ASME-JSME-KSME 2011 Joint Fluids Engineering Conference*, pp. 1801–1812. American Society of Mechanical Engineers.

[7] Mohamed, M., 2016. Reduction of the generated aero-acoustics noise of a vertical axis wind turbine using cfd (computational fluid dynamics) techniques. *Energy*, vol. 96, pp. 531–544.

[8] Mizzi, K., Demirel, Y. K., Banks, C., Turan, O., Kaklis, P., & Atlar, M., 2017. Design optimisation of propeller boss cap fins for enhanced propeller performance. *Applied Ocean Research*, vol. 62, pp. 210–222.

[9] Mehdipour, R., 2014. Simulating propeller and propeller-hull interaction in openfoam.

[10] Fluent, A., 2009. 12.0 theory guide. *Ansys Inc*, vol. 5, n. 5.

[11] Farrell, P. & Maddison, J., 2011. Conservative interpolation between volume meshes by local galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, vol. 200, n. 1-4, pp. 89–100.

[12] Carneiro, F., Moura, L., Rocha, P. C., Lima, R. P., & Ismail, K., 2019. Application and analysis of the moving mesh algorithm ami in a small scale hawt: Validation with field test's results against the frozen rotor approach. *Energy*, vol. 171, pp. 819–829.

[13] Mishra, V., Beatty, S., Buckham, B., Oshkai, P., & Crawford, C., 2015. Application of an arbitrary mesh interface for cfd simulation of an oscillating wave energy converter. In *Proc. 11th Eur. Wave Tidal Energy Conf*, pp. 07B141–07B1410.

[14] Wilcox, D. C. et al., 1998. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA.

[15] Pope, S. B., 2001. Turbulent flows.

[16] Menter, F., 1993. Zonal two equation kw turbulence models for aerodynamic flows. In *23rd fluid dynamics, plasmadynamics, and lasers conference*, pp. 2906.

[17] Turunen, T., Siikonen, T., Lundberg, J., & Bensow, R., 2014. Open-water computations of a marine propeller using openfoam. In *ECFD VI-6th European Congress on Computational Fluid Dynamics, Barcelona, Spain, 20-25 July 2014*, pp. 1123–1134.

[18] Chandar, D. D. & Gopalan, H., 2016. Comparative analysis of the arbitrary mesh interface (ami) and overset methods for dynamic body motions in openfoam. In *46th AIAA Fluid Dynamics Conference*, pp. 3324.

[19] Dose, B., 2013. *CFD Simulations of a 2.5 MW wind turbine using ANSYS CFX and OpenFOAM*. PhD thesis, MSc Thesis, UAS Kiel and FhG IWES, Germany.

[20] Menter, F. R., 1992. Influence of freestream values on k-omega turbulence model predictions. *AIAA journal*, vol. 30, n. 6, pp. 1657–1659.

[21] Menter, F. R., 1994. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, vol. 32, n. 8, pp. 1598–1605.

[22] Menter, F. R., 2009. Review of the shear-stress transport turbulence model experience from an industrial perspective. *International journal of computational fluid dynamics*, vol. 23, n. 4, pp. 305–316.

[23] Miley, S. J., 1982. Catalog of low-reynolds-number airfoil data for wind-turbine applications. Technical report, Rockwell International Corp., Golden, CO (USA). Rocky Flats Plant; Texas A . . . .

[24] Barth, T. & Jespersen, D., 1989. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace sciences meeting*, pp. 366.

[25] Patankar, S., 2018. *Numerical heat transfer and fluid flow*. CRC press.

[26] Rhie, C. & Chow, W. L., 1983. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, vol. 21, n. 11, pp. 1525–1532.