ROT8: A Matlab App for Low Reynolds number Airfoil and Rotor Aerodynamic Design

ROT8: A Matlab App for Low Reynolds number Airfoil and Rotor Aerodynamic Design

Conference Paper · June 2023

DOI: 10.2514/6.2023-3379

2 authors:

Manuel Carreño Ruiz
Politecnico di Torino
20 PUBLICATIONS **67** CITATIONS
SEE PROFILE

Domenic D'Ambrosio
Politecnico di Torino
62 PUBLICATIONS **599** CITATIONS
SEE PROFILE

*ROT*8: A Matlab App for Low Reynolds number Airfoil and Rotor Aerodynamic Design

M. Carreño Ruiz * and D. D'Ambrosio †
Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, 10124, Italy

The proposed software aims to solve the increase in demand for rotor design modules generated by the exponential growth of applications involving Unmanned Aerial Systems (UASs). We offer a variety of aerodynamic solvers for rotor flows with different fidelities, including a Blade Element Momentum (BEM) method, a Free Vortex Wake (FVW) method, a Vortex Particle Method (VPM), and a data-driven approach based on experimental data. *ROT*8 allows the calculation of airfoil polars necessary for the aforementioned Reduced Order Models (ROMs) using XFOIL. This general framework can be applied to rotors, propellers, and horizontal-axis wind turbines. The code is a Matlab App that allows using different optimization algorithms for design purposes. The code is intended to be versatile, with straightforward default settings for academic use but allowing the definition of user-defined functions to increase the level of interaction with the code for advanced research purposes.

I. Nomenclature

$C_Q = \frac{Q}{\rho\pi\Omega^2R^5}$	=	rotor torque coefficient
$C_T = \frac{T}{\rho\pi\Omega^2R^4}$	=	rotor thrust coefficient
$C_P = \frac{P}{\rho\pi\Omega^3R^5}$	=	rotor power coefficient
FM	=	rotor figure of merit
$C_l = \frac{l}{\frac{1}{2}\rho V^2c}$	=	airfoil lift coefficient
$C_d = \frac{d}{\frac{1}{2}\rho V^2c}$	=	airfoil drag coefficient
c	=	airfoil chord
V	=	freestream velocity
l	=	airfoil lift
d	=	airfoil drag
D	=	rotor diameter
Ω	=	rotation rate in radians per second
RPM	=	rotation rate in revolutions per minute
R	=	rotor radius
Re	=	rotor Reynolds number computed at 75% propeller radius
Re_c	=	rotor Reynolds number based on the chord
M	=	rotor tip Mach number
T	=	rotor thrust
Q	=	rotor torque
P	=	rotor power
ρ	=	air density
γ	=	intermittency
k	=	turbulent kinetic energy
Re_θ	=	momentum thickness Reynolds number
ω	=	specific dissipation rate

*PhD Student, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10124 Torino, Italy, manuel.carreno@polito.it

†Adjunct Professor, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10124 Torino, Italy, domenic.dambrosio@polito.it

II. Introduction

The increase in the popularity of Unmanned Aerial Systems (UASs) makes necessary the development of reduced order models that allow the calculation of the loads generated by rotatory wings without the high computational costs of Computational Fluid Dynamics. The Blade Element Momentum (BEM) method is the most common and fastest methodology to estimate rotor performance. However, some advanced design phases require a more reliable prediction of load distributions and wake topology. The Free Vortex Wake (FVW) method and the Vortex Particle Method (VPM) represent higher fidelity alternatives that explicitly model the wake effect with vortex filaments and vortex particles, respectively. These methodologies offer an advantage as they can model blade vortex interaction, resulting in a better prediction of the local induction caused by the vortex system. Novel applications such as Urban Air Mobility (UAM), high-altitude, or Martian flight require the generation of new rotor geometries that respond to different restrictions compared to the traditional helicopter blade design. Therefore, rotor design still concentrates numerous research efforts.

Regarding Ultra-low Reynolds number rotor design, Koning et al. [1, 2] have presented detailed aerodynamic and optimization analyses of rotors and airfoils for the Mars environment, which led to the design of the JPL Martian helicopter and are currently aiming to improve their work, presenting the Mars Science Helicopter (MSH) [3]. Bèzard & Desert [4, 5] have performed extensive numerical and experimental research on airfoils and rotors, presenting their design of a coaxial helicopter. The previously cited works use Reduced Order Models (ROMs) and higher fidelity Computational Fluid Dynamics (CFD) to evaluate blades, but both use ROMs for their design and optimization procedure. Researchers from the US presented results using CAMRADII [6] whereas the French group uses the PUMA (Potential Unsteady Methods for Aerodynamics) code [7]. They are both Free Vortex Wake (FVW) methods that, in a time-accurate manner, resolve the evolution of the wake and can model Blade Vortex Interaction (BVI), which may have a relevant effect in hovering conditions. These codes model the rotors with a Lifting-Line approach using a 2D airfoil database.

Regarding Urban Air Mobility(UAM), there has recently been a significant increase in companies and research groups that aim to design rotors with particular restrictions. This impact was thoroughly analyzed in [8]. Several researchers have performed numerical simulations in this field. Alvarez [9] presents a reformulated Vortex Particle Method, FlowVPM, which depending on the number of particles, allows a higher or lower fidelity in the simulation of multi-rotor aircraft. Ventura-Diaz [10] shows high-fidelity simulations of UAM vehicles in forwarding flight, including a loose coupling with CAMRADII to trim the vehicle. This shows the need for reliable ROMs not only for aerodynamic assessment and design purposes but also to inform other disciplines such as aeroacoustics, flight dynamics, or aeroelasticity with rapid yet accurate aerodynamic predictions.

There are many Blade Element Momentum implementations reported in literature. PropCode [11], QBlade [12], JBlade [13], JavaProp [14] and QProp [15] are some popular BEM approaches among many other. As the complexity increases, it becomes less and less usual to find available codes, and they are often not open-source. Some Free Vortex Wake implementations include CAMRADII [6], PUMA from ONERA [7], and Qblade [16]. The same argument holds for Free Vortex Particle methods. These approaches were developed in the last decades of the past century, with great contributions from Winckelmans [17], but have recently gained popularity in the simulation of Unmanned Aerial Systems (UAS) and the field of distributed propulsion in the context of Urban Air Mobility (UAM) and other applications. Two codes implementing different variations of this technique are DUST [18] and FLOWVPM [9].

This paper presents a Matlab App that compiles several rotor ROMs to allow the user a modeling fidelity choice. The graphic interface allows an easy setup of the simulations and optimization algorithms. However, we have attempted to achieve a high level of interaction with the code using user-defined Matlab functions. For example, for the polar input, objective functions, rotor parameterizations, etc. The code is developed to handle the ultra-low Reynolds number regime found on Mars, but the employed corrections do not affect simulations for higher Reynolds numbers. The App is linked with Xfoil [19] to enable a coupled inviscid-viscous analysis of airfoils and to allow a semi-automatic polar calculation for the airfoils composing the rotor. We provide a simulation module, able to compute loads and wake characteristics and a design module in which rotors and airfoils can be optimized following user objectives and restrictions. This allows an optimization procedure in the same framework to avoid complex and time-consuming communications between different software. Furthermore, the same treatment of the geometry inputs enables an effective comparison of the effects of the wake modeling on performance.

III. Airfoil module

The Xfoil airfoil solver is employed in this application. The aforementioned software incorporates a panel method alongside an integral boundary layer formulation discussed by Drela [19]. Its creation aimed to evaluate the

efficiency of airfoils in low Reynolds number conditions. The Xfoil algorithm can determine the pressure distribution, being able to handle the presence of laminar separation bubbles and minor trailing edge separation. The present study relies on Xfoil 6.99 version, which was implemented in Matlab, applying the library outlined in reference [20]. The code has been adapted, rendering it robust to non-convergent cases. The aforementioned modifications are crucial in preventing the failure of the optimization algorithms. XFOIL uses the e^N envelope technique to forecast the onset of turbulence. The proposed approach involves analyzing the maximum amplified frequency occurring at a designated location on the airfoil downstream from the instability point to approximate the magnitude of the disturbance. Transition is assumed to occur when the integrated amplitude exceeds an empirically defined value. The following equation establishes a correlation between the value of N and the turbulence intensity in the freestream.

$$N = 8.43 - 2.4ln(Tu) \tag{1}$$

Uranga [21] has shown that Xfoil has the capability to forecast the transition caused by separation, given that the turbulence intensity is suitably calibrated. The replication of the pressure plateau and posterior recovery phenomena is achievable. The obtained skin friction fails to consider the noteworthy adverse skin friction observed within separation bubbles before the reattachment of the boundary layer.

The convergence of Xfoil tends to fail at high Mach numbers. To mitigate the possibility of rejecting good geometries only based on convergence difficulties, Xfoil operates under conditions of incompressibility. The compressibility correction proposed by Kàrman-Tsien is employed to rectify the pressure distributions. The resulting values are then integrated to obtain the coefficients of lift and drag.

At present, *ROT8* offers two distinct parametrizations for airfoils, namely the NACA 4-digit formulation and the Class Shape Transformation (CST) technique [22], which enables a greater degree of flexibility in airfoil design. The low computational expense of Xfoil enables optimization algorithms to achieve very short run times. The design module allows one to select between running optimization algorithms or performing a sweeping analysis of the exploration domain. For instance, one could run the whole NACA 4-digit family to obtain indicative values of camber and thickness distributions and their impact on a certain quantity. Currently, particle swarm and genetic algorithms implemented in Matlab can be selected for single-objective optimization, and the genetic algorithm is also available for multi-objective optimizations.

A. Example: Aerodynamic Efficiency of the NACA XX02 family in Ultra-Low Reynolds number conditions

Figure 1 shows the output of a NACA family sweep performed with *ROT8* fixing the relative thickness at 2% at Reynolds numbers 10,000 and a Mach number of 0.5. Xfoil runs for a range of angles of attack between 2 and 7 degrees with an increment of 0.1 degrees. The reason to start at 2 degrees is to avoid leading edge separation in the pressure side that thin cambered airfoils often present. At these Reynolds numbers, transition to turbulence is unlikely. If it was to occur, it would happen due to the amplification of Kelvin-Helmholtz instability in the separated shear layer. Actually, ultra-low Reynolds number flows at large angles of attack are characterized by a large-scale leading edge vortex shedding that creates a high lift regime avoiding stall defined in its classical terms. Efficiency drops, but the lift continues to increase. Clearly, Xfoil being a steady code, will be unable to resolve this regime, and therefore it would be pointless to run large angles of attack. Another recommendation when using Xfoil in this regime is to increase the N factor to avoid transition, as this might generate unrealistic reattachment of the boundary layer with large efficiency peaks. When running Xfoil in laminar mode, when leading edge separation occurs, lift remains almost constant, and the drag coefficient increases dramatically, creating a clear efficiency peak. Beyond the maximum aerodynamic efficiency peak, Xfoil results in laminar mode shall be discarded. However, the results when the flow is attached or mildly separated are in good agreement with those obtained using two-dimensional unsteady Navier-Stokes simulations [5].

B. Example: Multiobjective optimization of Airfoil operating in Martian conditions

We present the use of this APP to perform an airfoil multi-objective optimization aiming to maximize lift and aerodynamic efficiency. The simulation has been performed for a Reynolds number of 10,000 and a Mach number of 0.1. The *gamultiobj* algorithm embedded in MATLAB was used to evolve the Pareto front. A CST parametrization of the airfoil with 8 input parameters was used in the optimization loop. Figure 2 shows 10 airfoils corresponding to the resultant Pareto front at different values of the lift coefficient. We can appreciate how, as the lift coefficient increases, the optimal curvature of the airfoil increases, reducing the aerodynamic efficiency. If we were to optimize only for aerodynamic efficiency, our result would be close to the first airfoil presented. As we increase the lifting demand, airfoils start to camber beyond the optimal, causing a drop in aerodynamic efficiency.

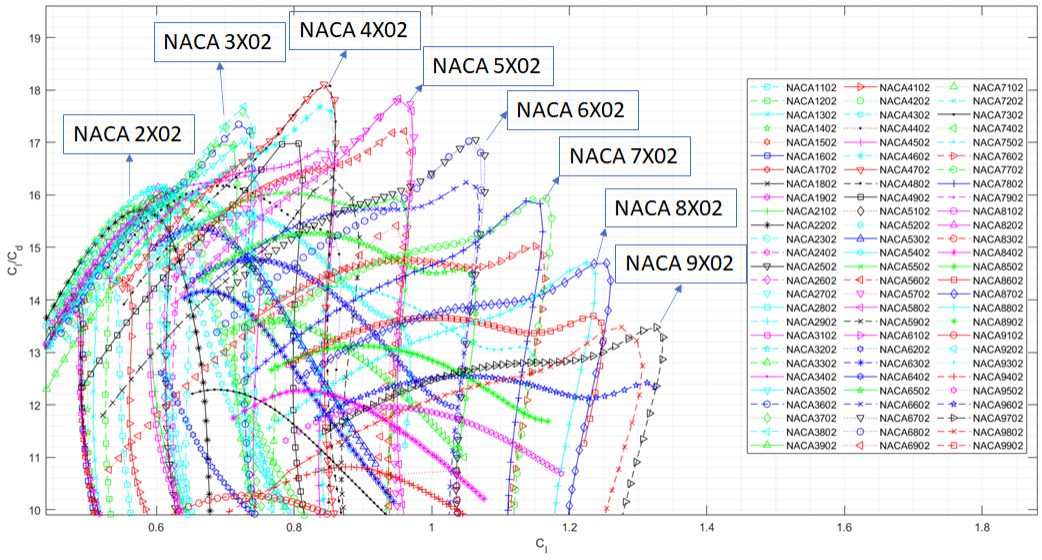


Fig. 1 NACA XX02 family sweep at Reynolds 10,000 and Mach 0.5

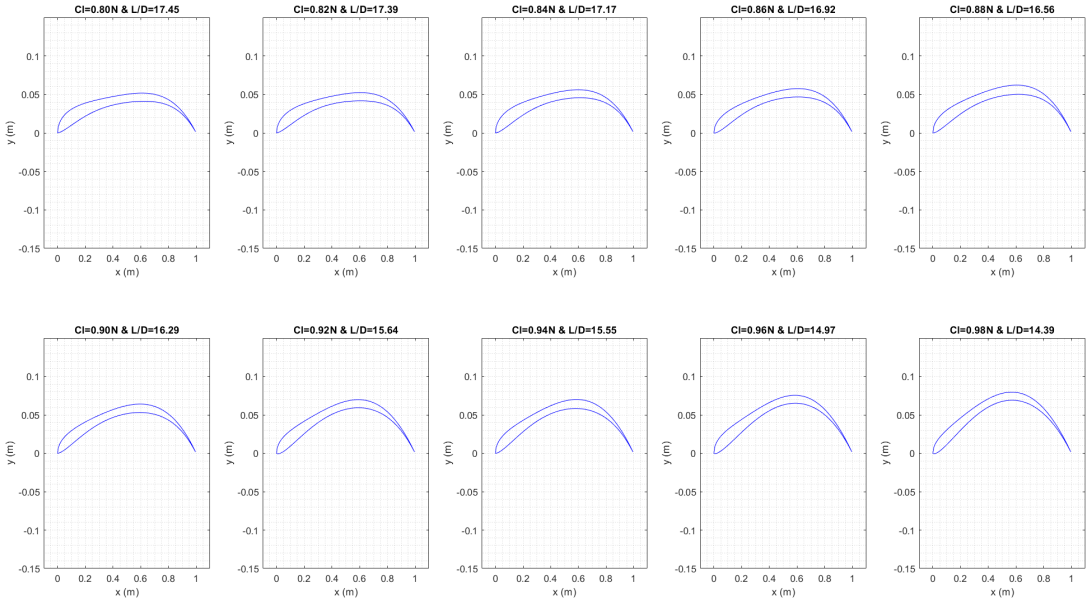


Fig. 2 Pareto optimal airfoils for different values of the lift coefficient following the CST parametrization.

IV. Rotor Module

The App offers different aerodynamic solvers to evaluate rotor performance at different computational costs. This allows the user to set the solver to be included in the optimization loop depending on their available computational resources. The solvers presented below have been developed and validated in previous publications[23–28].

A. Blade Element Method

The current implementation uses the traditional double-link methodology, which employs an iterative procedure until convergence of the momentum and blade element theory's induction factors is accomplished. At a specific radial location, the equations that represent the axial and tangential induction factors are as follows:

$$a = \frac{(V_x^2 + V_t^2)c(C_l cos(\phi) - C_d sin(\phi))B}{8\pi r V_\infty^2 (1 + a')} \tag{2}$$

$$b = \frac{(V_x^2 + V_t^2)c(C_d cos(\phi) + C_l sin(\phi))B}{8\pi r^2 V_\infty (1 + a')\Omega} \tag{3}$$

where V_∞ is the flight speed, $V_x = V_\infty(1 + a')$ is the axial velocity, $V_t = \Omega r(1 - b' - b_{visc})$ is the tangential velocity and $\phi = tan^{-1}(\frac{V_x}{V_t})$ is the inflow angle. a' and b' represent the values of the induction factors obtained in the previous iteration. These are under-relaxed to ensure convergence. This procedure is compliant with hovering and low advance ratios, as the axial induction factor con achieve values above 1.

For interpolating lift and drag coefficients, the current study includes an option to compute an aerodynamic database for a given airfoil using Xfoil, as described in the previous section. Having externally added polars through a user-defined Matlab function is also possible. Using Glauert's formulation, [29] Prandtl's tip and hub loss corrections were implemented.

$$F_{tip} = \frac{2}{\pi} \arccos\left(e^{-\frac{B}{2}\frac{R - r}{r \sin\phi}} \right) \tag{4}$$

$$F_{hub} = \frac{2}{\pi} \arccos\left(e^{-\frac{B}{2}\frac{r - R_h}{r \sin\phi}} \right) \tag{5}$$

The determination of a global correction factor can be mathematically represented as the multiplication of the correction factor for the blade tip, which is identified as F_{tip}, and the correction factor for the blade hub, which is identified as F_{hub}. This yields the overall correction coefficient, expressed as $F = F_{tip} \cdot F_{hub}$. Several methodologies can be utilized to integrate this variable within the blade element formulation. The present code applies a correction directly to the circulation, as shown in [9] for the vortex particle method, to guarantee a uniform implementation throughout the three Reduced Order Models introduced in this code.

The code incorporates the classical angular momentum correction, as described in Kunz [30], which accounts for the viscous swirl. This correction is denoted as b_{visc} and is calculated using the formula $\frac{2V_\infty a' C_d}{\Omega r C_l}$. The viscous swirl model will have minimal impact on the outcomes at typical Reynolds numbers for micro air vehicles operating in sea-level terrestrial conditions. This is because the model approaches zero asymptotically as the inverse of aerodynamic efficiency increases. Including this correction in the code is deemed necessary for completeness, as the authors have employed it to calculate the aerodynamic efficiency of Martian blades operating at a Reynolds number lower than 10,000, as shown in References [24, 25] and also in one of the examples in the following section.

The Blade Element Momentum (BEM) methodology demands geometric data that specifies the blade, including the radial distributions of chord and twist. The rotor performance prediction tool offers the highest speed among its competitors, with a computational expense of approximately 10^{-4} CPU hours. Due to its rapidity and straightforward nature, it is among the most frequently used for design applications. In the context of aerodynamics at very-low Reynolds numbers, it has been observed that blades with high aerodynamic efficiency tend to possess greater solidities, as reported in prior studies [4, 25, 31]. However, this may lead to potential errors regarding the prediction of local angles of attack, which are a crucial aspect of the design process. Bezard[4, 32] draws attention to the challenge posed by this issue.

The thrust on each panel is calculated as the sum of the vertical component of both lift and drag; the drag and lift on each panel are computed as:

$$dD = c_d(\alpha)\frac{1}{2}\rho \left[\left(\vec{V} \cdot \vec{a}_1\right)^2 + \left(\vec{V} \cdot \vec{a}_3\right)^2 \right] dA \tag{6}$$

$$dL = c_l(\alpha)\frac{1}{2}\rho \left[\left(\vec{V} \cdot \vec{a}_1\right)^2 + \left(\vec{V} \cdot \vec{a}_3\right)^2 \right] dA \tag{7}$$

Then, considering that the angle between the lift (perpendicular to the velocity in the plane defined by the vectors $\vec{a}_1$ and $\vec{a}_3$) and the vertical axis is

$$\phi = \beta - \alpha \tag{8}$$

where β is the angle between $\vec{a}_1$ and the horizontal axis, we can calculate that

$$dT = dL\cos(\phi) - dD\sin(\phi) \tag{9}$$

because drag is perpendicular to lift, as shown in figure 3.

The torque is calculated as the product of the horizontal component of both lift and drag and the radial position of the blade element control point.

$$dQ = r\left[dL\sin(\phi) + dD\cos(\phi)\right] \tag{10}$$

The thrust and torque calculations in the ROMs presented in the following sections are analogous to those shown above for the BEM model. Vectors $\vec{a}_1$ and $\vec{a}_3$ have been defined according to Van Garrel's [33] panel definition.

B. Free Vortex Wake

The NLLT-FVW code was created as part of the Master's Thesis project documented in [27]. The Matlab environment was utilized to develop the code, inspired by van Garrel's work[33]. Subsequent modifications to the code are detailed in the following lines. The employed approach utilizes the lifting line theory for rotor modeling, wherein including the airfoil's polar renders the problem non-linear. This feature is particularly crucial at low Reynolds numbers, where the lift curves deviate from linear behavior. The rotor model is integrated with the Free Vortex Wake (FVW) approach to simulate the wake, which can undergo deformation due to the velocities induced by the global vortex system created by the rotor.

The blade elements are represented as horseshoe vortices that rotate along with the blade. The determination of the circulation at every blade panel j is determined as

$$\Gamma_j = \frac{[c_l(\alpha)]_j \frac{1}{2}\left[\left(\vec{V}_j \cdot \vec{a}_{1,j}\right)^2 + \left(\vec{V}_j \cdot \vec{a}_{3,j}\right)^2\right] dA_j}{\sqrt{\left[\left(\vec{V}_j \times d\vec{l}_j\right) \cdot \vec{a}_{1,j}\right]^2 + \left[\left(\vec{V}_j \times d\vec{l}_j\right) \cdot \vec{a}_{3,j}\right]^2}} \tag{11}$$

The velocity $\vec{V}_j$ is determined based on the control points on the blade panel. The chordwise and normal versors on the panel are denoted by $\vec{a}_{1,j}$ and $\vec{a}_{3,j}$ respectively, following van Garrel's notation [33]. The panel area is represented by dA_j, while $d\vec{l}_j$ denotes the filament length or spanwise panel width.

The velocity computed at the control point of the panel j is calculated as

$$\vec{V}_j = \vec{V}_{wind} + \vec{V}_{ind,pan} + \vec{V}_{ind,fil} - \vec{\Omega} \times \vec{r} \tag{12}$$

In the previous expression $\vec{\Omega}$ denotes the angular velocity of the blade, while $\vec{r}$ represents the vector radius of the panel control point from the axis origin. $\vec{V}_{wind}$ denotes the wind velocity relative to the rotor, while $\vec{V}_{ind,pan}$ indicates

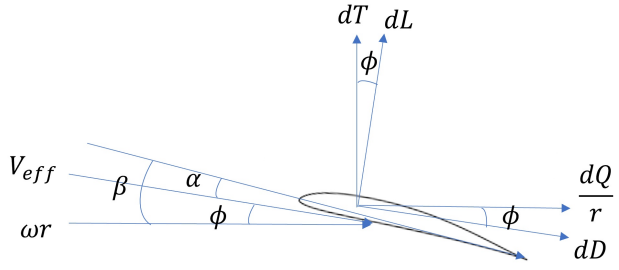


Fig. 3 Thrust and torque calculation

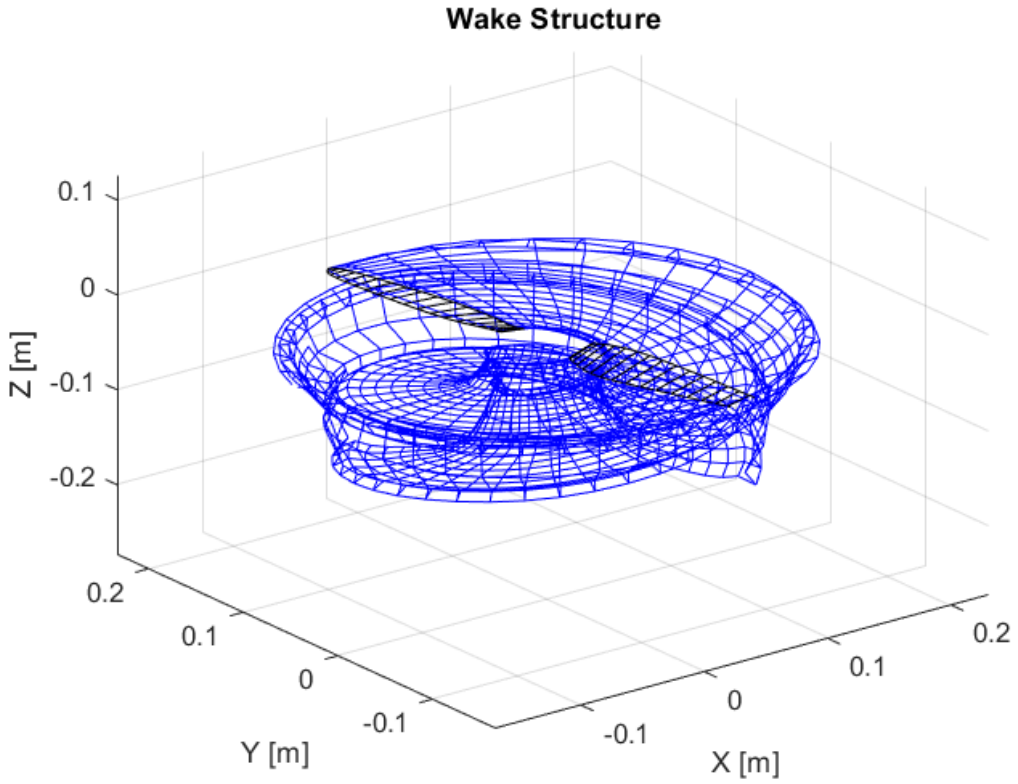


Fig. 4 Vortex filaments forming the wake behind a hovering Tmotor 15x5 after 1 revolution.

the induced velocities by the vortex system that models the blade. Additionally, $\vec{V}_{ind,fil}$ represents the induced velocity resulting from the vortex filaments in the wake. The computation of the lift coefficient is interpolated from an aerodynamic database of the airfoils composing the blade. The induced velocities are computed by applying a desingularized version of the Biot-Savart law. This study's approach employed for desingularization is based on Van-Garrel's method [33]. However, it differs from the original approach in that it incorporates a regularization technique independent of the vortex filament length, similar to the method implemented in Qblade [16]. This code adapts the approach employed by Qblade by raising the exponent of the core radius r_c to 4, ensuring the dimensional consistency of the expression below.

$$\vec{u}_\Gamma = \frac{\Gamma}{4\pi} \frac{(r_1 + r_2)(\vec{r_1} \times \vec{r_2})}{r_1 r_2 (r_1 r_2 + \vec{r_1} \cdot \vec{r_2}) + r_c^4} \tag{13}$$

This correction aims to mitigate the impact of the rotor scale on the parameters of the model. The study by Ali [27] shows how this formulation appears to enhance the prediction of the blade loading. Nevertheless, solver stability is often compromised, particularly in hovering calculations, due to the self-induced vortex system that lacks an inflow, which typically stabilizes the vortex system.

As the lift coefficient is a non-linear function of the angle of attack, a function of circulation, Eq. (11) is iteratively solved as shown in Eq. (14).

$$\Gamma_j^{k+1} = \Gamma_j^k + RF(\Gamma_j - \Gamma_j^k) \tag{14}$$

where RF is a relaxation factor, which is necessary to avoid the divergence of Γ_j^k. The convergence criterion is

$$\frac{|\max_j(\Gamma_j^{k+1} - \Gamma_j^k)|}{|\max_j \Gamma_j^k| + 1} \leq toll \tag{15}$$

The initial value for the Γ_j distribution is a value of 1 at the initial time step. Afterward, the code allocates the converged circulation distribution obtained at the preceding time interval as an initial guess.

The previous section demonstrated the implementation of Prandtl's tip and hub loss corrections, commonly utilized to account for the effects of hub and tip. Neglecting these effects would result in an incomplete analysis. Consequently, the circulation derived from Equation 14 is multiplied by the product of two factors, namely $F = F_{tip} \cdot F_{hub}$.

After achieving convergence in panel circulation, the blade undergoes rotation following the predetermined time step and rotation rate. The wake emits two distinct types of vortex filaments. The spanwise circulation variation results in the emission of trailing vortices, while the temporal variation of circulation generates shed vortices, satisfying the Bjerkness-Kelvin theorem. Determining the convection of individual vortex filaments involves numerical integration of the aggregate velocities induced on each filament by the other vortex filaments. The code implements a second-order predictor-corrector scheme for temporal integration. Incorporating a Lamb-Oseen viscous core model, as demonstrated in Bhagwat [34], facilitates the temporal growth of the vortex due to dissipation effects. Furthermore, a correction is implemented to the viscous core radius to accommodate the vortex stretching effect, as Sants [35] outlined. Following the previous revisions, the core radius denoted by r_c can be expressed as follows:

$$r_c = \sqrt{\frac{5.026 \cdot \delta_\nu \cdot \nu(t + S_c)}{1 + \epsilon}} \tag{16}$$

The present model is characterized by two different parameters, namely δ_ν and S_c. The tuning of these parameters allows a balance between reliable induction and solution stability. The impact of these parameters has been comprehensively examined in the analysis presented in references [27, 35]. The symbols t and ϵ denote the age and strain rate of the vortex, respectively.

The velocity induced by individual vortex filaments at each respective vortex node must be evaluated at every time step. The relationship between the number of vortices and time is linear, while the computational cost exhibits a quadratic increase with respect to the number of revolutions. Hence, the computational expense is susceptible to the solver setup and the inflow parameters. The default discretization uses 15 evenly distributed panels along the radial direction. The duration of the simulation corresponds to twelve complete revolutions of the rotor. The computational expense ranges from 10^{-1} to 1 CPU hour, notably less than that of CFD simulations but considerably higher than the blade element momentum method described above. Achieving complete parallelization of induction calculations results in a noteworthy decrease in simulation duration when implemented on parallel computing systems.

Figure 4 illustrates a complete revolution of the vortex wake structure acquired for the test case discussed in the following section. The hovering state results in the wake rolling around the tip vortex and passing near the blade.

C. Vortex Particle Method

The Viscous VPM software was created during a Master's thesis[26]. exaFMM [36], an open-source C++ library that implements a Fast Multipole Method (FMM), is employed to acceleration calculations. The evaluation of the circulation distribution on the blade is conducted through the use of the lifting line theory, as referenced in the free vortex wake formulation. The code creates a collection of vortex particles every time step, which are employed to discretize the vorticity field. Prandtl's Tip and Hub corrections were reintroduced to address the impact of hub and tip effects, as performed in the free vortex wake formulation.

The particles are generated from the trailing edge of the blade according to the spanwise variation (trailing vortex particles, shed chordwise) and the time variation (shed vortex particles, shed spanwise) of the circulation on the blade, as shown in Ref. [37]. Then, they are evolved according to the VPM evolution equations[38],

$$\frac{d\vec{x}^p}{dt} = -\frac{1}{4\pi}\sum_{q=1}^{N_p} \frac{|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \frac{5}{2}\sigma^2}{(|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \sigma^2)^{\frac{5}{2}}}(\vec{x}^p(t) - \vec{x}^q(t)) \times \vec{\alpha}^q(t) \tag{17}$$

$$\begin{aligned}\frac{d\vec{\alpha}^p}{dt} = \frac{1}{4\pi}\sum_{q=1}^{N_p}\Bigg[&\frac{|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \frac{5}{2}\sigma^2}{(|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \sigma^2)^{\frac{5}{2}}}\vec{\alpha}^p(t) \times \vec{\alpha}^q(t) + \\ +3&\frac{|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \frac{7}{2}\sigma^2}{(|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \sigma^2)^{\frac{7}{2}}}(\vec{\alpha}^p(t) \cdot (\vec{x}^p(t) - \vec{x}^q(t)) \times \vec{\alpha}^q(t))(\vec{x}^p(t) - \vec{x}^q(t)) + \\ &+105\nu\frac{\sigma^4}{(|\vec{x}^p(t) - \vec{x}^q(t)|^2 + \sigma^2)^{\frac{9}{2}}}(vol^p\vec{\alpha}^q(t) - vol^q\vec{\alpha}^p(t))\Bigg]\end{aligned} \tag{18}$$

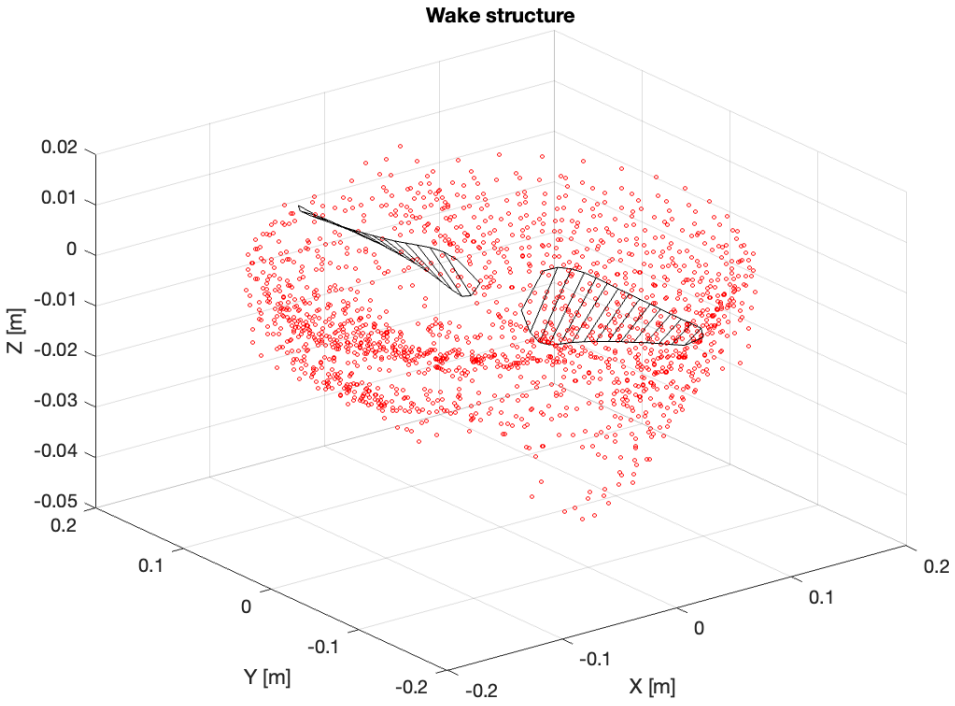


Fig. 5 Vortex particles forming the wake behind a hovering Tmotor 15x5 after 1 revolution.

The assessment of the induced velocity of particles on themselves involves utilizing a high-order algebraic regularization function for the regularization of the Biot-Savart kernel. The transpose scheme (vortex stretching term) and particle strength exchange (viscous diffusion term) model the circulation strength. The computational expense of the evolution equations is of $O(N_p^2)$ due to its formulation as an "N-body problem", where N_p denotes the number of particles present in the simulation. The Fast Multipole Method [39] algorithm exaFMM [36] is used, enabling a calculation with a computational cost of $O(N_p)$. The implementation of Pedrizzetti's relaxation scheme [40] is applied to maintain the divergence-free condition of the velocity field. The circulation strength of the particles is updated at every time step Δt as:

$$\vec{\alpha}^p(t)new = (1 - f\Delta t)\vec{\alpha}^p(t)_{old} + f\Delta t, \vec{\omega}_\sigma^p(\vec{x}^p, t)\left|\frac{\vec{\alpha}^p(t)_{old}}{\vec{\omega}_\sigma^p(\vec{x}^p, t)}\right| \tag{19}$$

The relaxation factor denoted by f is frequently fixed at 0.02. A scheme for particle splitting and merging, as outlined in Ref. [41], was implemented to prevent numerical instabilities and Lagrangian distortion. The temporal evolution of the equations is computed using a third-order low-storage four-step Runge-Kutta scheme. The dimension of the core radius is a tuning parameter that increases the fidelity of the approach; the smaller the core size, the better the vorticity field is represented at the expense of a higher computational cost. The rotor is discretized using a uniform paneling with 15 elements. The simulation duration is equivalent to twelve complete revolutions of the rotor, deemed adequate for achieving convergence of both thrust and torque on a global and local scale, given the specific rotor and operational parameters used in this work. The computational time ranges from 25 to 30 CPU hours. This duration exceeds that of the NLLT-FVW code yet falls significantly below that of CFD simulations. An example of the vortex particle wake structure obtained for the initial test case presented in the next section is illustrated in Figure 5.

D. Gaussian Process Regression Model for Fast Rotor Analysis

The aim of this model is to obtain a preliminary propeller performance using input front and top images of propellers. For this purpose, the propeller database of the University of Illinois at Urbana-Champaign (UIUC) [42] was used. In its two first volumes, around 120 propellers were experimentally tested and precisely photographed. A Machine Learning

model was trained using as inputs post-processed data from the original rotor images present in the database to match thrust and torque coefficients.

Propeller performance depends on several operational and geometrical parameters, among which the most important are the chord and twist distributions, airfoil employed, diameter, sweep, surface roughness, and angular velocity. Performing a simplified dimensional analysis neglecting sweep and roughness, the thrust and torque coefficient results to depend only on the Reynolds number, Mach number, non-dimensional twist, chord distributions, and the airfoil employed at each radial station. In fact, at the scale of propellers and rotors that we are dealing with, of only a few inches diameter, the Mach number effects are typically negligible, and the Reynolds number will be relevant, especially for the smaller propellers. These simplifications imply that the coefficient mainly depends on the chord, twist, and airfoil radial distribution. It is possible to perform an approximation of chord and twist distributions using as an input the front and top views of the propellers, as done, for example, by the open-source software PropellerScanner by Martin Hepperle. Unfortunately, no information can be induced from the images regarding the airfoil distribution. This is the biggest limitation of this approach.

Propeller images were post-processed to estimate the chord and twist in 17 different radial positions, and this was considered our input data. The variables predicted by the Machine Learning (ML) model are the thrust and torque coefficients obtained from the UIUC propeller database. The Gaussian Process Regression models provided the best validation metrics using k-fold validation, which we employed to avoid data loss considering the small database. The most performing GPR model, employing a 5/2 Matern kernel, obtained an R-squared value of 0.88 and a root-mean-square error of 0.0072. A plot showing the validation predicted against the actual value is shown in 6 for the thrust coefficients.

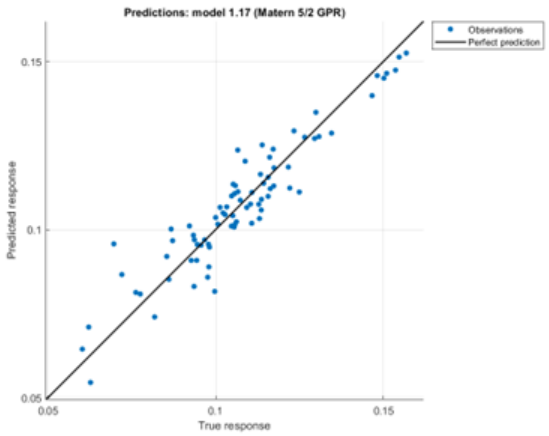


Fig. 6 Validation of a Matern 5/2 Kernel GPR model using k-fold validation for volume-1 propellers.

This module offers the possibility of performing instantaneous thrust and torque coefficient predictions simply by uploading front and side view images of the rotor.

Two random rotors which were not part of the training set and that have associated experimental and numerical work have been tested to analyze the fidelity of the predictions. These are the Tmotor 15"x5" rotor and Delft Aerocoustic Benchmark blades shown in figure 7. These blades have been extensively studied experimentally and numerically by references [43] and [44]. Table 1 compares the GPR predictions of the thrust and power coefficients for these blades hovering at 4000 rpm with the available experimental and numerical data. It is interesting how starting from an image, this machine-learning technique can predict the order of magnitude of the performance coefficients of the blade with errors below 20% compared to experiments and around 10 % if we consider numerical results.

(a) Tmotor 15"x5" cf **(b) TU Delft Aeroacoustic Benchmark [44]**

Fig. 7 Input geometries for Gaussian Process Regression model predictions.

	CFD		Experimental		GPR model	
Blade	C_T	C_P	C_T	C_P	C_T	C_P
Tmotor 15"x5"	0.00895	0.000879	0.00919	0.000912	0.01019	0.000838
TU Delft	0.0165	0.00232	0.0148	0.00206	0.0166	0.00246

Table 1 GPR model predictions of Tmotor 15"x5" and Delft Benchmark blades compared to CFD and experimental data. CFD and experimental data obtained is obtained from [43] for the Tmotor 15"x5" blade and from [44] for the Delft Benchmark blades.

E. Test case: Tmotor 15"x 5"

We present the performance predictions obtained with our previously proposed reduced order models compared with CFD simulations and experimental results described in [43, 45] at different Reynolds numbers. The polars used in the reduced order models have been computed using CFD using the γ-Re$_\theta$ transition model following the guidelines presented in [46]. The performance comparison is presented in table 2. We can appreciate a satisfactory agreement between ROMs, CFD, and experimental results, with average errors below 5% referenced to the experimental results.

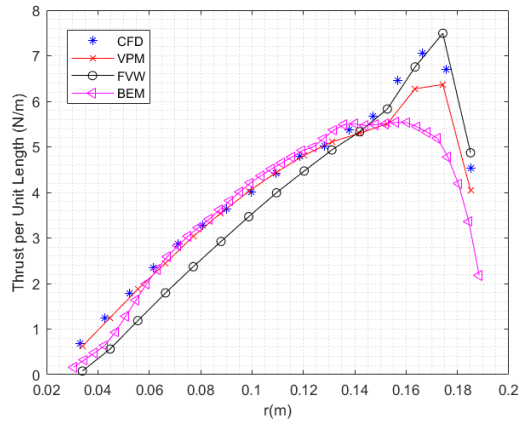
	BEM		FVW		VPM		CFD		Experimental	
Re	**T [N]**	**Q [Nm]**	**T [N]**	**Q [Nm]**	**T [N]**	**Q [Nm]**	**T [N]**	**Q [Nm]**	**T [N]**	**Q [Nm]**
24,099	1.19	0.029	1.21	0.0307	1.27	0.0310	1.33	0.030	1.26	0.029
61,539	4.18	0.082	3.95	0.0848	3.92	0.083	4.12	0.083	4.16	0.082
186,670	13.80	0.236	13.55	0.248	12.71	0.245	13.16	0.0246	13.41	0.253
CPU time(h)	0.0003		0.5		28		2000		-	

Table 2 Different fidelity numerical solvers and experimental[45] rotor performance for different Reynolds numbers.

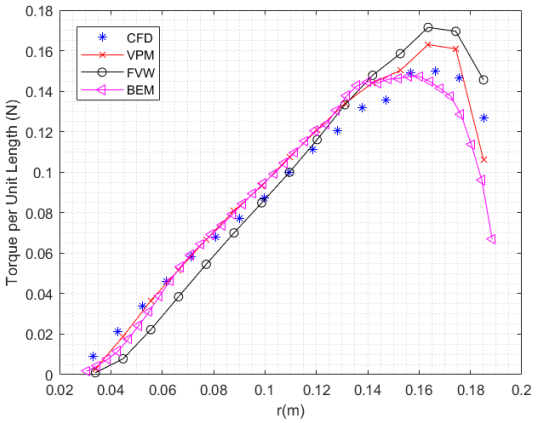
Figure 8 shows the local comparison of the local thrust and torque forces with those calculated using CFD presented in [23]. We can appreciate how the torque values present qualitatively the same behavior for all ROMs. This is possibly caused due to the smaller sensitivity of drag coefficients to effective angles of attack when these are low. On the other hand, the thrust per unit length distributions shows different trends depending on the ROM. For instance, BEM seems to overpredict the angles of attack in the mid-span of the blade. Conversely, FVW and VPM can adequately capture the inflection point on the local thrust distribution seen in CFD simulations, which is probably caused by the close passage of the tip vortex. The accuracy of the VPM thrust distributions compared with a well-resolved RANS CFD simulation [43], including a transition model, is impressive, considering the difference in computational cost.

F. Rotor Optimization

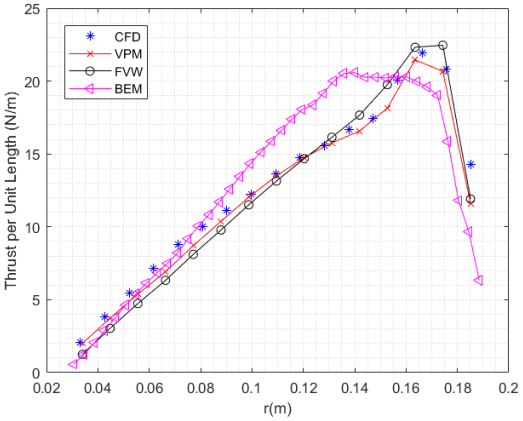
We propose a rotor parametrization based on radial chord and twist distributions at different stations, which are interpolated with an Akima spline. The code also allows the selection of a user-defined function to define a different parametrization strategy. The code allows the selection of different objective functions, such as Thrust, Power, or Torque. However, to increase the versatility of the code, we allow the users to create a Matlab function to implement the desired objective function and allow the inclusion of restrictions as penalization functions. The App allows using Matlab genetic optimization algorithms, both single-objective and multi-objective versions, and the particle swarm methodology for single-objective as in the airfoil case. An initial random sweep of the exploration domain may be performed using the genetic algorithm with a large population and a small number of generations. When using the optimization module with the BEM solver, selecting the *fmincon*, Matlab's gradient-base optimization routine can be selected. In this case, it is important to reduce the BEM relative tolerance to allow accurate gradient calculations. Once the optimization algorithm converges, the code generates an EXCEL spreadsheet equipped with a macro that automatically imports splines of the



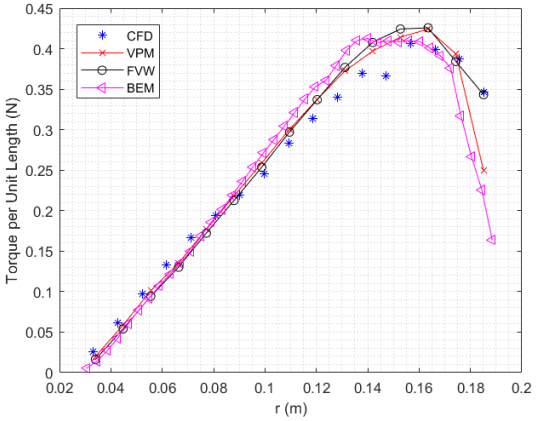
(a) Re=24,099



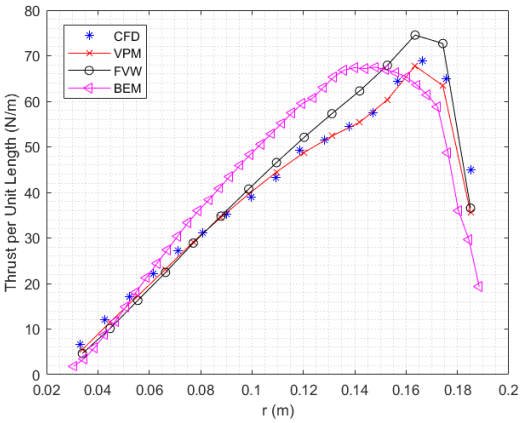
(b) Re=24,099



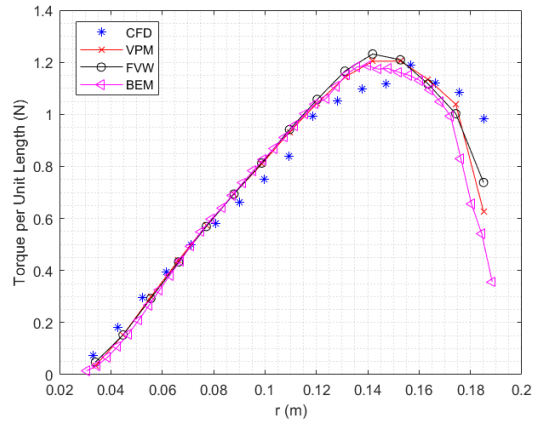
(c) Re=61,539



(d) Re=61,539



(e) Re=186,670



(f) Re=186,670

Fig. 8 Thrust and torque radial distributions for different Reynolds numbers using the CFD-based polar for the Reduced Order Models.

airfoils into Solidworks. We also provide a geometry matrix that contains the chord and twist radial distributions and the position of the airfoil's quarter chord in every section.

G. Example: Multi-Objective Rotor Optimization for Martian Conditions

We present an optimization of a hovering rotor operating in Martian conditions using the BEM solver. The restrictions are a minimum thrust of 1.1 N and a fixed diameter of 0.4 m. Twist and chord radial distributions and the rotation rate are considered design variables. In this case, multiobjective optimization is performed to maximize efficiency and minimize the weight of the blade on Mars. Figure 9 compares the Pareto front (Blue markers) and the efficiency calculated using the net thrust (Red markers). Figures 10 and 11 show the chord and twist distributions from Pareto optimal geometries with different weights. We can appreciate how as the weight increases, the blade can increase the chord reducing in this way the negative low Reynolds number effects. We can see how large chords near the hub help increase efficiency, but the improvements are small compared to the increase in weight. This can be clearly seen in the net thrust curve. For more details on this example please refer to reference [25].

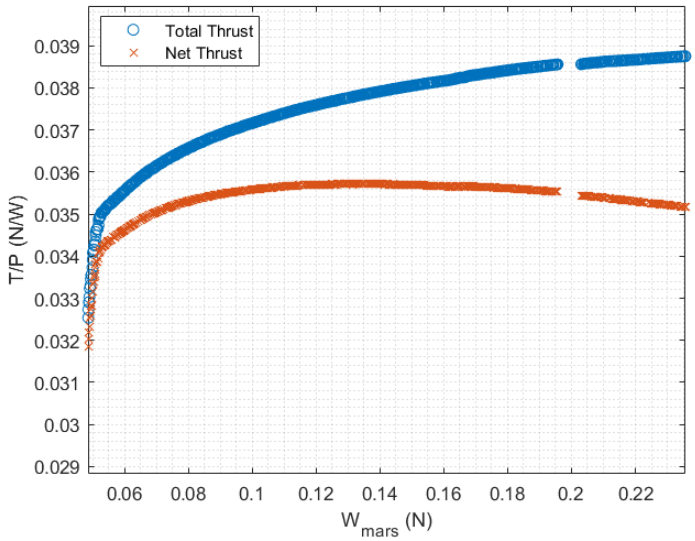


Fig. 9 Pareto front and net thrust curves.

V. Conclusions

We presented a Matlab App that allows the simulation and design of airfoils and rotatory wings. The paper shows different test cases and examples of design activities using this simulation module, including validating the code for a Tmotor 15"x5" blade. The code aims to be sufficiently intuitive to allow its educational use but has a level of user interaction that renders it an interesting research application. Combining simulation and optimization in the same framework circumvents the sometimes painful and time-consuming task of communicating different software. Furthermore, the different ROMs use the same geometrical inputs and interpolation strategies, allowing a realistic comparison of the fidelity between different approaches. The computational cost of the approaches ranges from a few milliseconds to days, exhibiting the high cost that relatively small increases in accuracy have. However, this increase in fidelity is mandatory for certain applications, and the higher cost might be justified.

The presented examples applied to the ultra-low and very-low Reynolds number regimes. We provided preliminary guidelines on using Xfoil and our code in these situations. We have demonstrated how these reduced fidelity tools with adequate settings can predict these complex regimes where even classical CFD methods have trouble aligning with experimental results.

Future works include adding 6-DOF motion to the rotor to emulate rotor maneuvers and including a swashplateless rotor design module. Also, a further generalization of the code is planned to allow even more user interaction, and the combination with other disciplines, such as aeroacoustics or aeroelasticity, will be considered.

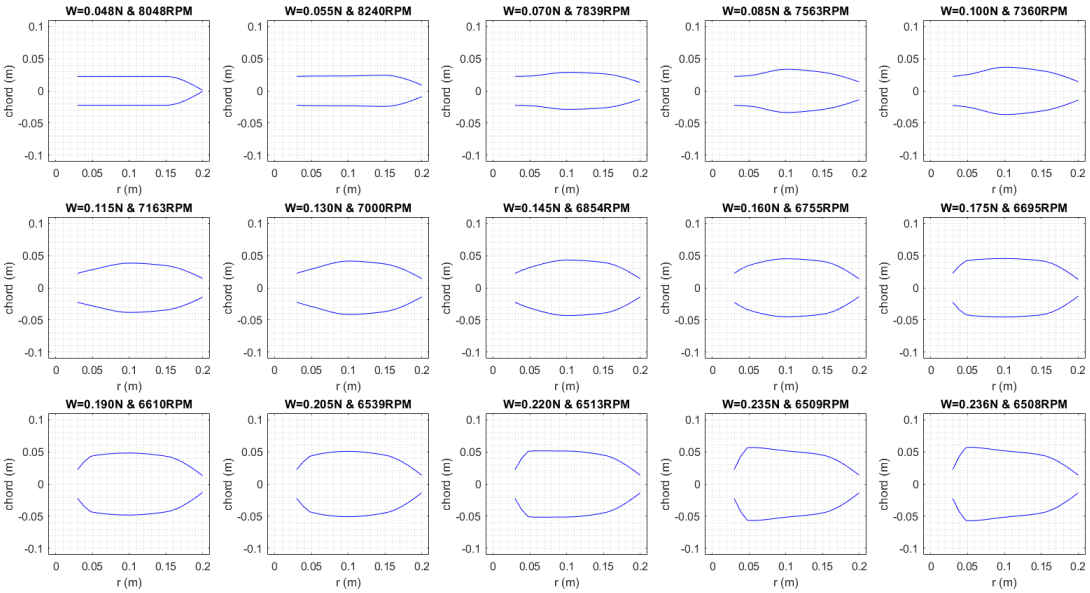


Fig. 10 Chord distributions for different Pareto optimal solutions.

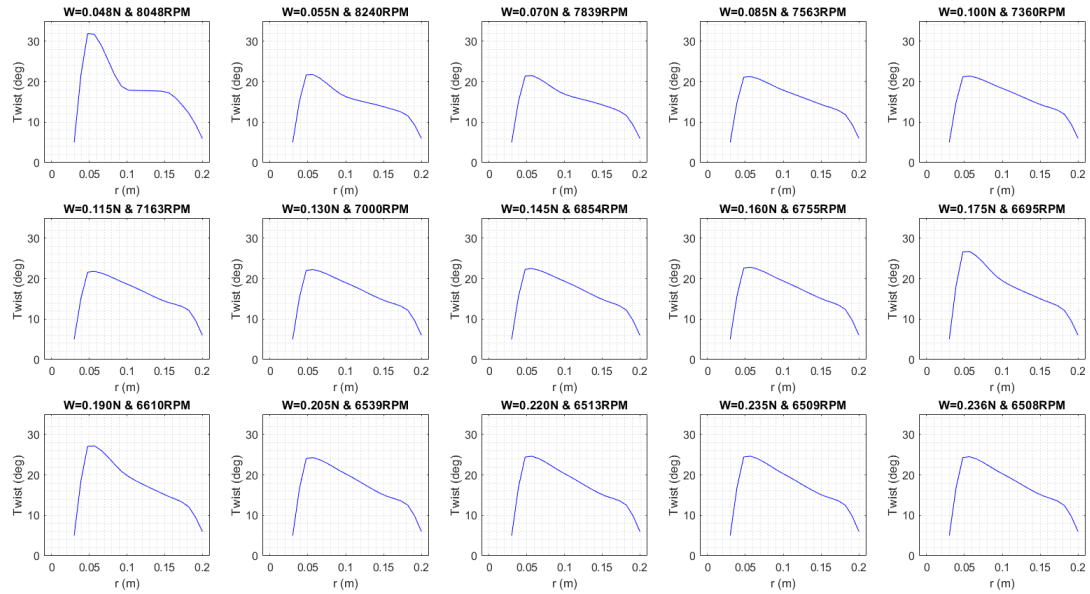


Fig. 11 Twist distributions for different Pareto optimal solutions.

References

[1] Koning, W. J., Johnson, W., and Allan, B. G., "Generation of Mars Helicopter Rotor Model for Comprehensive Analyses," *AHS Aeromechanics Design for Transformative Vertical Flight*, 2018.

[2] Koning, W. J., Romander, E. A., and Johnson, W., "Optimization of Low Reynolds Number Airfoils for Martian Rotor Applications Using an Evolutionary Algorithm," *AIAA Scitech 2020 Forum*, 2020, p. 0084.

[3] Johnson, W., Withrow-Maser, S., Young, L., Malpica, C., Koning, W. J., Kuang, W., Fehler, M., Tuano, A., Chan, A., Datta, A.,

et al., "Mars science helicopter conceptual design," Tech. rep., 2020.

[4] Bézard, H., Desert, T., Moschetta, J.-M., and Jardin, T., "Aerodynamic design of a Martian micro air vehicle," *EUCASS 2019*, MADRID, Spain, 2019. URL https://hal.archives-ouvertes.fr/hal-02397054.

[5] Desert, T., Moschetta, J.-M., and Bézard, H., "Numerical and experimental investigation of an airfoil design for a Martian micro rotorcraft," *International Journal of Micro Air Vehicles*, Vol. 10, No. 3, 2018, pp. 262–272.

[6] Nguyen, K., and Johnson, W., "Evaluation of dynamic stall models with UH-60A airloads flight test data," *Annual forum proceedings-American helicopter society*, Vol. 54, American Helicopter Society, 1998, pp. 576–588.

[7] Le Bouar, G., Costes, M., Leroy-Chesneau, A., and Devinant, P., "Numerical simulations of unsteady aerodynamics of helicopter rotor in manoeuvring flight conditions," *Aerospace Science and Technology*, Vol. 8, No. 1, 2004, pp. 11–25. https://doi.org/10.1016/j.ast.2003.08.004.

[8] Goyal, R., Reiche, C., Fernando, C., and Cohen, A., "Advanced Air Mobility: Demand Analysis and Market Potential of the Airport Shuttle and Air Taxi Markets," *Sustainability*, Vol. 13, No. 13, 2021. https://doi.org/10.3390/su13137421, URL https://www.mdpi.com/2071-1050/13/13/7421.

[9] Alvarez, E. J., "Reformulated Vortex Particle Method and Meshless Large Eddy Simulation of Multirotor Aircraft," Ph.D. thesis, Brigham Young University, 2022.

[10] Ventura Diaz, P., and Yoon, S., "High-Fidelity Simulations of a Quadrotor Vehicle for Urban Air Mobility," *AIAA SCITECH 2022 Forum*, 2022, p. 0152.

[11] Bergmann, O., Möhren, F., Braun, C., and Janser, F., "Comparison of Various Aeroacoustic Propeller Noise Prediction Methodologies in Static Operations," *AIAA SciTech 2022 Forum*, January 2022. https://doi.org/10.2514/6.2022-2529, AIAA-2022-2529.

[12] Marten, D., Wendler, J., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O., "QBLADE: an open source tool for design and simulation of horizontal and vertical axis wind turbines," *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, No. 3, 2013, pp. 264–269.

[13] Silvestre, M. A., Morgado, J. P., and Pascoa, J., "JBLADE: a propeller design and analysis code," *2013 International Powered Lift Conference*, 2013, p. 4220.

[14] Hepperle, M., "JavaProp - Design and Analysis of Propellers," , 2001. URL http://www.mh-aerotools.de/airfoils/javaprop.htm, Accessed: 2023-01-01.

[15] Drela, M., "QPROP Formulation," , 2006. URL https://web.mit.edu/drela/Public/web/qprop/qprop_theory.pdf, Accessed: 2023-01-01.

[16] Marten, D., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O., "Implementation, optimization, and validation of a nonlinear lifting line-free vortex wake module within the wind turbine simulation code qblade," *Journal of Engineering for Gas Turbines and Power*, Vol. 138, No. 7, 2016.

[17] Winckelmans, G., and Leonard, A., "Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows," *Journal of Computational Physics*, Vol. 109, No. 2, 1993, pp. 247–273.

[18] Montagnani, D., Tugnoli, M., Fonte, F., Zanotti, A., Syal, M., and Droandi, G., "Mid-fidelity analysis of unsteady interactional aerodynamics of complex vtol configurations," *45th European Rotorcraft Forum (ERF 2019)*, 2019, pp. 100–110.

[19] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.

[20] Edelman, L., "Xfoil interface updated," https://www.mathworks.com/matlabcentral/fileexchange/49706-xfoil-interface-updated, 2014. Online; 18 February 2022.

[21] Uranga, A., "Investigation of transition to turbulence at low Reynolds numbers using Implicit Large Eddy Simulations with a Discontinuous Galerkin method," Ph.D. thesis, University of California, Berkeley, 2010.

[22] Kulfan, B. M., "The CST universal parametric geometry representation method, recent extensions and applications," *Proc R Aeronaut Soc Conf*, Vol. 114, 2007, pp. 157–176.

[23] Carreño Ruiz, M., Manavella, A., and D'Ambrosio, D., "Numerical and experimental validation and comparison of reduced order models for small scale rotor hovering performance prediction," *AIAA SCITECH 2022 Forum*, 2022. https://doi.org/10.2514/6.2022-0154.

[24] Carreno Ruiz, M., and D'Ambrosio, D., "Hybrid Fidelity Optimization of Efficient Airfoils and Rotors in Ultra-Low Reynolds Numbers Conditions," *AIAA SCITECH 2023 Forum*, 2023, p. 0652.

[25] Carreño Ruiz, M., and D'Ambrosio, D., "Aerodynamic optimization and analysis of quadrotor blades operating in the Martian atmosphere," *Aerospace Science and Technology*, Vol. 132, 2023, p. 108047. https://doi.org/10.1016/j.ast.2022.108047.

[26] Bellelli, F., "Development and validation of a Vortex Particle code to evaluate Rotor and Propeller Performance," Master's thesis, Politecnico di Torino, 2022.

[27] Ali, M., "Development and validation of a reduced order method for the study of flow on a rotor," Master's thesis, Politecnico di Torino, 2021.

[28] Manavella, A., "Low Reynolds number propeller performance validation by CFD analysis and reduced order models," Master's thesis, Politecnico di Torino, 2021.

[29] Glauert, H., "Aerodynamic theory," *The Aeronautical Journal*, Vol. 34, No. 233, 1930, pp. 409–414.

[30] Kunz, P. J., "Aerodynamics and design for ultra-low Reynolds number flight," Ph.D. thesis, Standford University, 2003.

[31] Carreño Ruiz, M., and D'Ambrosio, D., "Aerodynamic Optimization of Quadrotor Blades Operating in the Martian Atmosphere," *AIAA SCITECH 2022 Forum*, 2022. https://doi.org/10.2514/6.2022-0743.

[32] Bézard, H., Désert, T., Jardin, T., and Moschetta, J.-M., "Numerical And Experimental Aerodynamic Investigation Of A Micro-UAV For Flying On Mars," *76th Annual Forum & Technology Display*, 2020.

[33] Van Garrel, A., "Development of a wind turbine aerodynamics simulation module," 2003.

[34] Bhagwat, M. J., and Leishman, J. G., "Generalized viscous vortex model for application to free-vortex wake and aeroacoustic calculations," *Annual forum proceedings-American helicopter society*, Vol. 58, American Helicopter Society, Inc, 2002, pp. 2042–2057.

[35] Sant, T., "Improving BEM-based Aerodynamic Models in Wind Turbine Design Codes," Ph.D. thesis, Delft University of Technology, 2007.

[36] Tingyu Wang, L. A. B., Rio Yokota, "ExaFMM: a high-performance fast multipole method library with C++ and Python interfaces," *The Journal of Open Source Software*, 2021. https://doi.org/doi:10.21105/joss.03145.

[37] Singh, P., "Aeromechanics of Coaxial Rotor Helicopters using the Viscous Vortex Particle Method," Ph.D. thesis, University of Michigan, 2020. https://deepblue.lib.umich.edu/handle/2027.42/163227.

[38] Wincklemans, G. S., "Topics in vortex methods for the computation of three- and two-dimensional incompressible unsteady flows," Master's thesis, California Institute of Technology, Feb. 1989.

[39] Greengard, L., and Rokhlin, V., "A fast algorithm for particle simulations," *Journal of Computational Physics*, Vol. 73, No. 2, 1987, pp. 325–348. https://doi.org/10.1016/0021-9991(87)90140-9.

[40] Pedrizzetti, G., "Insight into singular vortex flows," *Fluid Dynamics Research*, Vol. 10, No. 2, 1992, pp. 101–115. https://doi.org/10.1016/0169-5983(92)90011-k.

[41] Winckelmans, G., and Leonard, A., "Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows," *Journal of Computational Physics*, Vol. 109, No. 2, 1993, pp. 247–273. https://doi.org/10.1006/jcph.1993.1216.

[42] J.B. Brandt, G. A. O. D., R.W. Deters, and Selig, M., "UIUC Propeller Database, Vols 1-4, University of Illinois at Urbana-Champaign, Department of Aerospace Engineering," , 2001. URL https://m-selig.ae.illinois.edu/props/propDB.html, Accessed: 2023-01-01.

[43] Carreño Ruiz, M., Scanavino, M., D'Ambrosio, D., Guglieri, G., and Vilardi, A., "Experimental and numerical analysis of hovering multicopter performance in low-Reynolds number conditions," *Aerospace Science and Technology*, Vol. 128, 2022, p. 107777. https://doi.org/10.1016/j.ast.2022.107777.

[44] Casalino, D., Romani, G., Zhang, R., and Chen, H., "Lattice-Boltzmann calculations of rotor aeroacoustics in transitional boundary layer regime," *Aerospace Science and Technology*, Vol. 130, 2022, p. 107953.

[45] Scanavino, M., "Design and testing methodologies for UAVs under extreme environmental conditions," Ph.D. thesis, Politecnico di Torino, 2021.

[46] Carreño Ruiz, M., and D'Ambrosio, D., "Validation of the $\gamma - Re_\theta$ Transition Model for Airfoils Operating in the Very Low Reynolds Number Regime," *Flow, Turbulence and Combustion*, 2022, pp. 1–30. https://doi.org/10.1007/s10494-022-00331-z.