Physics of Fluids

RESEARCH ARTICLE | JULY 15 2024

Eulerian–Lagrangian hybrid solvers in external aerodynamics: Modeling and analysis of airfoil stall

R. Pasolari ; C. J. Ferreira ; A. van Zuijlen

Physics of Fluids 36, 077135 (2024)
https://doi.org/10.1063/5.0216634

Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Eulerian–Lagrangian hybrid solvers in external aerodynamics: Modeling and analysis of airfoil stall

Cite as: Phys. Fluids **36**, 077135 (2024); doi: 10.1063/5.0216634
Submitted: 30 April 2024 · Accepted: 24 June 2024 ·
Published Online: 15 July 2024

View Online Export Citation CrossMark

R. Pasolari,[a] C. J. Ferreira, and A. van Zuijlen

AFFILIATIONS

Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

[a]Author to whom correspondence should be addressed: r.pasolari@tudelft.nl and r.pasolari@gmail.com

ABSTRACT

Hybrid computational solvers that integrate Eulerian and Lagrangian methods are emerging as powerful tools in computational fluid dynamics, particularly for external aerodynamics. These solvers rely on the strengths of both approaches: Eulerian methods efficiently handle boundary layers, while Lagrangian methods excel in reducing numerical diffusion in flow convection. Building on our prior development of a two-dimensional hybrid solver that combines OpenFOAM with vortex particle method, this paper extends its application to the complex phenomena of airfoil stall at low Reynolds numbers. Specifically, we examine both static and dynamic stall conditions of a National Advisory Committee for Aeronautics (NACA) airfoil series 0012 (NACA0012) across a wide range of attack angles and oscillation frequencies, comparing our results with established data. The findings demonstrate the accuracy of hybrid Eulerian–Lagrangian solvers in replicating known stall behaviors, underscoring their potential for advanced aerodynamic studies. This work not only confirms the capability of hybrid solvers in accurately modeling challenging flows but also paves the way for their increased involvement in the field of external aerodynamics.

I. INTRODUCTION

Hybrid Eulerian–Lagrangian solvers have gained increasing attention in the last few decades, especially in the field of external aerodynamics. Eulerian–Lagrangian solvers combine the advantages of both Eulerian and Lagrangian solvers, and at the same time, they are leveraging their disadvantages. Specifically, Eulerian solvers perform great on resolving boundary layers, but they introduce numerical diffusion in the flow, especially in the wake region. On the other hand, Lagrangian solvers can convect the wake behind the aerodynamic bodies, eliminating the artificial diffusion, but they have critical bottlenecks in resolving near-wall regions. The idea behind the hybrid solver is to use the Eulerian solver for the near-wall region, while the evolution of the wake is modeled solely by the vortex particles. The computational domain can be split in such a way using the domain decomposition technique, introduced by Cottet[1] and later improved by Daeninck.[2] Up today, many hybrid Eulerian–Lagrangian solvers have been developed.[1–8] Stock *et al.*[3] coupled a vortex particle method (VPM) with OVERFLOW, a fully compressible solver, and later[9] a high-order spectral finite difference method with an open-source VPM. Palha *et al.*[4] coupled the finite elements method (FEM) FEniCS software with a VPM. Papadakis and Voutsinas[6] developed a strongly coupled compressible Eulerian–Lagrangian solver and used it for external compressible flows and flows, including FSI.[10] Recently, Pasolari *et al.*[8] coupled OpenFOAM[11] with a VPM and then demonstrated the dynamic mesh capabilities of the solver in Ref. 12.

This paper deals with the solver developed in Ref. 8 and extends its applications into the simulations of airfoils. In this way, the solver dives into more realistic scenarios than the cylinder validation cases in the previous work. The airfoil is a more complex geometry, introducing new physics phenomena into the flow, like that static stall. Static stall is a critical phenomenon in aerodynamics that occurs when the angle of attack of the airfoil increases beyond a certain point, leading to a sudden decrease in lift and a significant increase in drag, and occurs because of the separation of the airflow from the surface of the airfoil. Malhotra *et al.*[13] studied the static stall of the National Advisory Committee for Aeronautics (NACA) airfoil series 0012 (NACA0012) at Reynolds number of three million, using the $k-\omega$ SST (shear stress transport) turbulence model. Kurtulus[14] focused on investigating the same airfoil, but in lower Reynolds number and specifically $Re = 1000$. This low Reynolds number has extensive application in micro-air vehicles (MAVs). Using

the same case, Di Ilio *et al.*[15] validated the hybrid lattice-Boltzmann method they developed, and Nguyen *et al.*[16] validated their low-order Brinkman penalized vortex method and data-driven dynamic decomposition in angles of attack (aoa) 20°, 60°, and 90°.

Another critical phenomenon observed in airfoils is the dynamic stall. Unlike static stall, which is observed in fixed aoa, dynamic stall involves the rapid variation of the angle of attack, introducing additional complexities into the airflow behavior around airfoils. The dynamic stall is characterized by delayed stall onset, a temporary lift overshoot, and complex vortex-shedding patterns that significantly influence the lift and drag experienced by the airfoil. This phenomenon is particularly relevant in wind turbines, both horizontal axis wind turbines (HAWTs)[17] and vertical axis wind turbines,[18] helicopters,[19] highly maneuverable fighters,[20] and MAVs. Wang *et al.*[21] investigated computationally the dynamic stall of the NACA0012 airfoil at $Re = 10^5$ and compared the computational fluid dynamics (CFD) results with experimental. They used two different sets of oscillating patterns. Akbari and Price[22] used a vortex method to investigate the dynamic stall for the NACA0012 airfoil at $Re = 10^4$ and different oscillation patterns.

This paper focuses on the numerical investigation of the static and dynamic stall of the NACA0012 airfoil at $Re = 1000$, using the solver introduced in Ref. 8. The first aim of the paper is to validate the solver in a more complex case, such as the airfoil case, ensuring that the solver can predict complex phenomena occurring and the stalling aoa. While focused on our developed solver, this study extends its relevance to hybrid Eulerian–Lagrangian solvers, aiming to validate their application in such aerodynamic analyses. After the static stall validation, the solver is applied to investigate the dynamic stall of the NACA0012 at $Re = 1000$, a Reynolds number significant for many applications, such as the MAVs, which has not been investigated thoroughly in the bibliography. Up to our knowledge, works that have been published on the topic concern higher Reynolds numbers,[21,22] or the investigation was not into the dynamic stall, like the work of Kurtulus,[23] where small amplitude oscillation of 1° pitch amplitude were applied in 1 and 4 Hz oscillations, and compared to the non-oscillatory case. For this reason, pure OpenFOAM simulations accompany the hybrid simulations to validate the results. These two cases (static and dynamic stall) are essential to assess the accuracy of the hybrid solvers compared to pure Eulerian solvers in more complex and realistic flows.

The structure of the present paper is as follows: Sec. II briefly introduces the solver used for this work and gives reference to the readers who want to read it in more detail. Section III deals with the validation case of the static stall of the NACA0012. In this section, the computational case is discussed and validated through the results in the bibliography. Moreover, an analytical convergence test is applied to verify the validity of the results. After this, Sec. IV describes the applied case of the dynamic stall of the NACA0012 airfoil in different oscillation patterns, and the results are compared with the corresponding pure Eulerian results. The paper closes with Secs. V and VI, where the summary and a discussion on the potentials of the hybrid solver are taking place, respectively.

II. NUMERICAL METHOD

The numerical method employed in this study is a hybrid Eulerian–Lagrangian solver. The Eulerian part is based on OpenFOAM,[11] a cell-centered, finite volume, open-source software widely utilized in industry and academia. Being open-source, it permits modifications to existing solvers and the incorporation of new ones. The implementation of the finite volume method (FVM) within OpenFOAM is documented on the OpenFOAM website,[24] as well as in the books.[25,26] The core solver used is PimpleFOAM,[27] specifically version OpenFOAMv9.[28] Details on modifications made to OpenFOAM for coupling with the Lagrangian solver are available in Refs. 8 and 12.

The Lagrangian component utilizes the vortex particle method (VPM), a technique well suited for evolving flow fields without artificial diffusion. VPM naturally satisfies far-field boundary conditions and leverages acceleration techniques such as the Fast multipole method,[29] along with parallelization across CPUs and GPUs. VPM addresses the Navier–Stokes equations through a velocity–vorticity formulation [Eq. (1)], where the velocity ($\boldsymbol{u}$) and vorticity (ω) fields are obtained by summing the effects of each particle [Eq. (2)]. In this formulation, g_σ and ζ_σ are smoothing functions, Γ_p is the strength of the particle, and $\boldsymbol{U}_{inf}$ is the freestream velocity,

$$\frac{\partial \omega}{\partial t} + (\boldsymbol{u}\cdot\nabla)\omega = \nu\nabla^2\omega \quad \text{in 2D}, \tag{1}$$

$$\boldsymbol{u}_p(\boldsymbol{x}) = -\frac{1}{2\pi}\sum_p \frac{g_\sigma(|\boldsymbol{x}-\boldsymbol{x}_p|)}{|\boldsymbol{x}-\boldsymbol{x}_p|^2}(\boldsymbol{x}-\boldsymbol{x}_p)\times \boldsymbol{e}_z\Gamma_p + \boldsymbol{U}_{inf}, \tag{2a}$$

$$\omega_p(\boldsymbol{x}) = \sum_p \zeta_\sigma(|\boldsymbol{x}-\boldsymbol{x}_p|)\Gamma_p. \tag{2b}$$

The simulations progress through discrete advection and diffusion steps [Eq. (3)],[30] where ν is the kinematic viscosity of the fluid. Comprehensive discussions on vortex methods are provided by Cottet and Koumoutsakos,[31] with recent advancements reviewed by Mimeau and Mortazavi.[32] The specific application of VPM in this solver is detailed in Refs. 8 and 12,

$$\frac{\partial \omega}{\partial t} + \boldsymbol{u}\cdot\nabla\omega = 0 \quad \text{advection step}, \tag{3a}$$

$$\frac{\partial \omega}{\partial t} - \nu\nabla^2\omega = 0 \quad \text{diffusion step}. \tag{3b}$$

The present solver's approach combines the Eulerian method for resolving boundary layers with the Lagrangian solver for the remainder of the computational domain, thereby avoiding artificial diffusion in the flow. This effective coupling is achieved through domain decomposition strategies initially proposed by Cottet,[1] refined by Daeninck,[2] and successfully applied in Palha *et al.*[4] The domain decomposition for the NACA0012 airfoil case is illustrated in Fig. 1.

The Eulerian solver resolves the region very close to the wall, while the Lagrangian particles resolve the rest of the domain. The two solvers are coupled in a two-way manner, with the Lagrangian particles first providing boundary conditions for the outer Eulerian boundary (numerical boundary in Fig. 1), and in a later step, the Eulerian solution is used to correct the Lagrangian solution in the near-wall region. The steps employed during a hybrid time step can be summarized in the flow chart of Fig. 2. Readers are encouraged to consult the original publications[8,12] for a deeper understanding of this methodology.

III. VALIDATION—STATIC STALL

First, the solver is validated in the case of the static NACA0012 airfoil at $Re = 1000$. Different angles of attack are tested here, from the

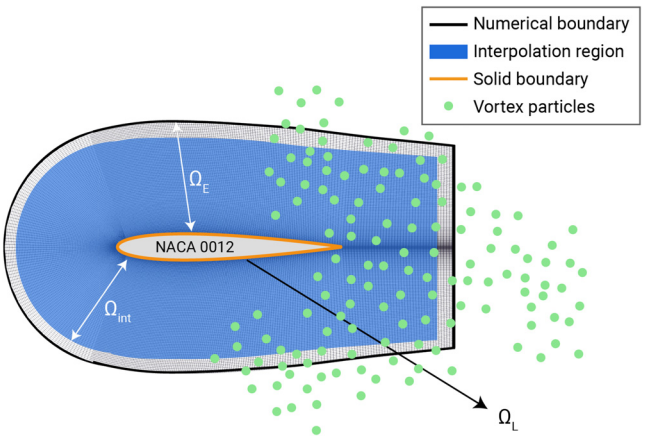


FIG. 1. The domain decomposition technique employed in the hybrid solver, presented for the case of a NACA0012 airfoil. The Eulerian solver resolves the near-wall region (Ω_E) from the solid boundary (orange line) up to the numerical boundary (black line), while the Lagrangian solver is responsible for evolving the wake downstream and is used for the entire computational domain (Ω_L). The Lagrangian solver is then corrected inside the interpolation region (Ω_{int}), which is illustrated in cyan.

symmetrical configuration of 0° aoa up to 30° aoa. This case aims to predict the flow field, as well as the static stall of the airfoil. The results obtained here will be compared with the corresponding results from Kurtulus,[14] Di Ilio *et al.*,[15] and Liu *et al.*[33]

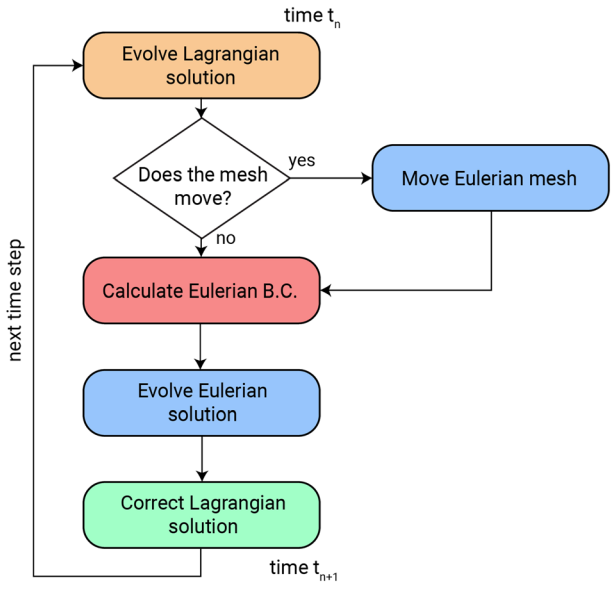


FIG. 2. The flow chart of the hybrid solver. Assuming that both solvers are in time t_n, the Lagrangian solver is evolved to t_{n+1}. Then, if the mesh is dynamic, the Eulerian mesh is updated, and the Eulerian boundary conditions are computed into the updated coordinates. If the mesh is static, then the boundary conditions are computed directly after the Lagrangian step. After this, the Eulerian solution can be evolved to t_{n+1}, and the final step is to correct the Lagrangian solution inside the interpolation region.

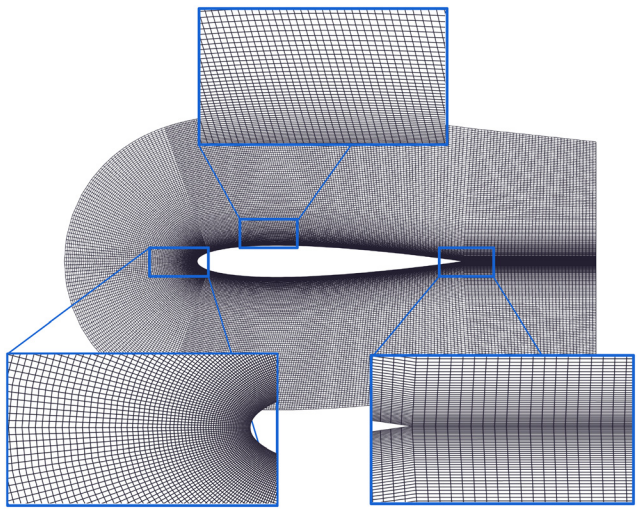


FIG. 3. The mesh of the NACA0012 airfoil created in Gmsh.[34] Snippets of the leading edge, the trailing edge, and the top surface of the airfoil are also present.

A. Case parameters and convergence tests

To ensure the accuracy of the simulation, conducting a convergence test is crucial. This involves identifying the optimal mesh resolution, time step, and particles' resolution (in terms of the particles' spacing h). The convergence test focuses on the airfoil at an 4° angle of attack. The selected mesh is C-shaped, entirely composed of hexahedra, and was generated using Gmsh.[34] Visual representations of the mesh, including detailed views of the leading edge, trailing edge, and top surface, are presented in Fig. 3. Additionally, a comprehensive summary of all simulation parameters employed in this study is provided in Table I.

For the spatial convergence, the mesh resolution and the particle spacing change simultaneously, while the time step is chosen to be small enough that even for the most refined mesh, the Courant number remains around 1.0. The mesh size and the particle size are refined by a factor of $\sqrt{2}$. However, since an exponential expansion is used in the normal direction of the airfoil, the total number of particles does not increase by the same factor. The findings of the spatial convergence study are detailed in Table II, where h, Cl, Cd, and p represent the particle spacing, the lift coefficient, the drag coefficient, and the convergence order, respectively.

TABLE I. The simulation parameters used for the case of the flow around a NACA0012 at $Re = 1000$ and aoa ranging from 0° to 30°.

Parameter	Symbol	Value	Dimension
Reynolds number	Re	10^3	...
Freestream velocity	U_{inf}	1.0	m/s
Chord length	C	1.0	m
Simulation time	t_{sim}	100	s
Interpolation domain offset from Eulerian boundary	d_{bdry}	0.1	m

TABLE II. Grid and particles' spacing convergence study for the case of the flow around a NACA0012 at $Re = 1000$ and 4° aoa.

Case	Mesh size	Time step	h	C_l	C_d	p
Coarse	3864	7.5×10^{-4}	0.0424	0.1641(−)	0.1239(−)	···
Refined (level 1)	6766	7.5×10^{-4}	0.0300	0.1819(9.8%)	0.1244(0.4%)	···
Refined (level 2)	11 844	7.5×10^{-4}	0.0212	0.1867(2.6%)	0.1244(0.0%)	3.8
Refined (level 3)	20 956	7.5×10^{-4}	0.0150	0.1900(1.7%)	0.1244(0.0%)	1.2
Refined (level 4)	36 544	7.5×10^{-4}	0.0106	0.1920(1.0%)	0.1244(0.0%)	1.3

The relative error in parentheses is computed as $\frac{Cl_{new} - Cl_{old}}{Cl_{new}} \cdot 100\%$. It can be seen in Table II that the aerodynamic coefficients converge to a solution, allowing us to determine them as $C_l = 0.1920$ and $C_d = 0.1244$. Therefore, it can be concluded that the fourth level of refinement is sufficient for running the rest of the simulations. For an additional check, we can calculate the order of convergence, p, using the formula in Eq. (4), where f_{level_i} is the solution at the current refinement level and r is the refinement factor,

$$p = ln \frac{f_{level_{i-1}} - f_{level_{i-2}}}{f_{level_i} - f_{level_{i-1}}} \Big/ ln(r). \tag{4}$$

From the value of p in Table II, it can be seen that the first set of solutions is outside the asymptotic range, while the next two sets are within the asymptotic range, with the order of convergence between 1.2 and 1.3. Although the discretization schemes used here are second-order, this order of convergence is not achieved due to several factors, such as the dependence of particle resolution on the mesh, and grid quality.

Now, having identified the mesh for which the solution can be considered independent, a time step convergence study can be conducted. As before, a second-order temporal discretization scheme is used. The convergence study results are presented in Table III.

The solution appears to converge more quickly for the time step convergence study. From the results in Table III, we can conclude that for the second level of time step refinement ($dt = 7.5 \times 10^{-4}$), the solution is independent of the time step.

B. Results—Static stall

First, we explore the flow fields. Figure 4 showcases the vorticity and velocity magnitude fields across various aoa. At aoa of 8° or lower, the wake remains steady without any flow instabilities. This observation aligns with the prediction by Di Ilio *et al.*,[15] who suggested that instability begins at an aoa of 8°. The present findings indicate that instability onset occurs between 8° and 9°. Specifically, at 8°, the flow is stable, but at 9°, instability is already evident. Beyond these angles, the wake becomes unsteady, exhibiting a Von Kármán vortex street. This phenomenon also reflects on the aerodynamic forces, leading to oscillations as depicted in Fig. 5 where the lift and drag coefficients are plotted for seven different aoa. At higher aoa, such as 25° and 28°, the wake no longer exhibits its sinusoidal periodicity and transitions into a more complex form of periodicity, as shown in both the contour plots (Fig. 4) and the aerodynamic coefficients (Fig. 5). Flow separation, which initiates at lower aoa (for instance, 4°) near the trailing edge, gradually progresses toward the leading edge as the aoa increases.

In order to assess the aerodynamic performance of the hybrid solver and its capability to predict the static stall of the airfoil, the lift coefficient is plotted over the angle of attack. The plot can be seen in Fig. 6. It must be mentioned here that a mean value is calculated for the cases where the wake is unsteady, excluding the first seconds of the simulation where the transient phenomena dominate. The results are compared with those from three bibliographical Refs. 14, 15, and 33. It is observed that the proposed solver shows great agreement with the references. Up to 12° aoa, all the solvers predict the lift coefficient similarly. However, there are slight deviations within the acceptable range after this point. It can be seen that up to 22° aoa, the present solver has a better agreement with the results from Di Ilio *et al.*,[15] while for the rest, the agreement is closer to the results of Kurtulus.[14] The proposed solver and the results from Kurtulus[14] and Di Ilio *et al.*[15] predict that the static stall occurs between 25° − 26° aoa, while Liu *et al.*[33] predicted a much higher lift coefficient for 27° aoa, and then the airfoil stalls. The results from Di Ilio *et al.*[15] differ from the other two references at 29°, while the present solver agrees with them. Finally, the result for 30° aoa has been included for the proposed solver. It can be mentioned here that slight differences in the aerodynamic coefficient in the high aoa regime may be present due to the high unsteadiness of the solution, and it matters from which point the mean value is derived.

Figure 7 shows the ratio between the lift and drag coefficient for the different aoa. Again, there is a very nice agreement between our solver and the references.[14,15] The present solver is in slightly better agreement with the results from Di Ilio *et al.*,[15] especially for the aoa between 5°and12°, where the results from Kurtulus[14] differ slightly,

TABLE III. Time step convergence study for the case of the flow around a NACA0012 at $Re = 1000$ and 4° aoa.

Temporal refinement	Spatial refinement	Time step	C_l	C_d
Coarse	Refined (level 4)	1.50×10^{-3}	0.1956(−)	0.1244(−)
Refined (level 1)	Refined (level 4)	1.06×10^{-3}	0.1935(1.1%)	0.1244(0.0%)
Refined (level 2)	Refined (level 4)	7.50×10^{-4}	0.1920(0.8%)	0.1244(0.0%)
Refined (level 3)	Refined (level 4)	5.30×10^{-4}	0.1916(0.2%)	0.1244(0.0%)

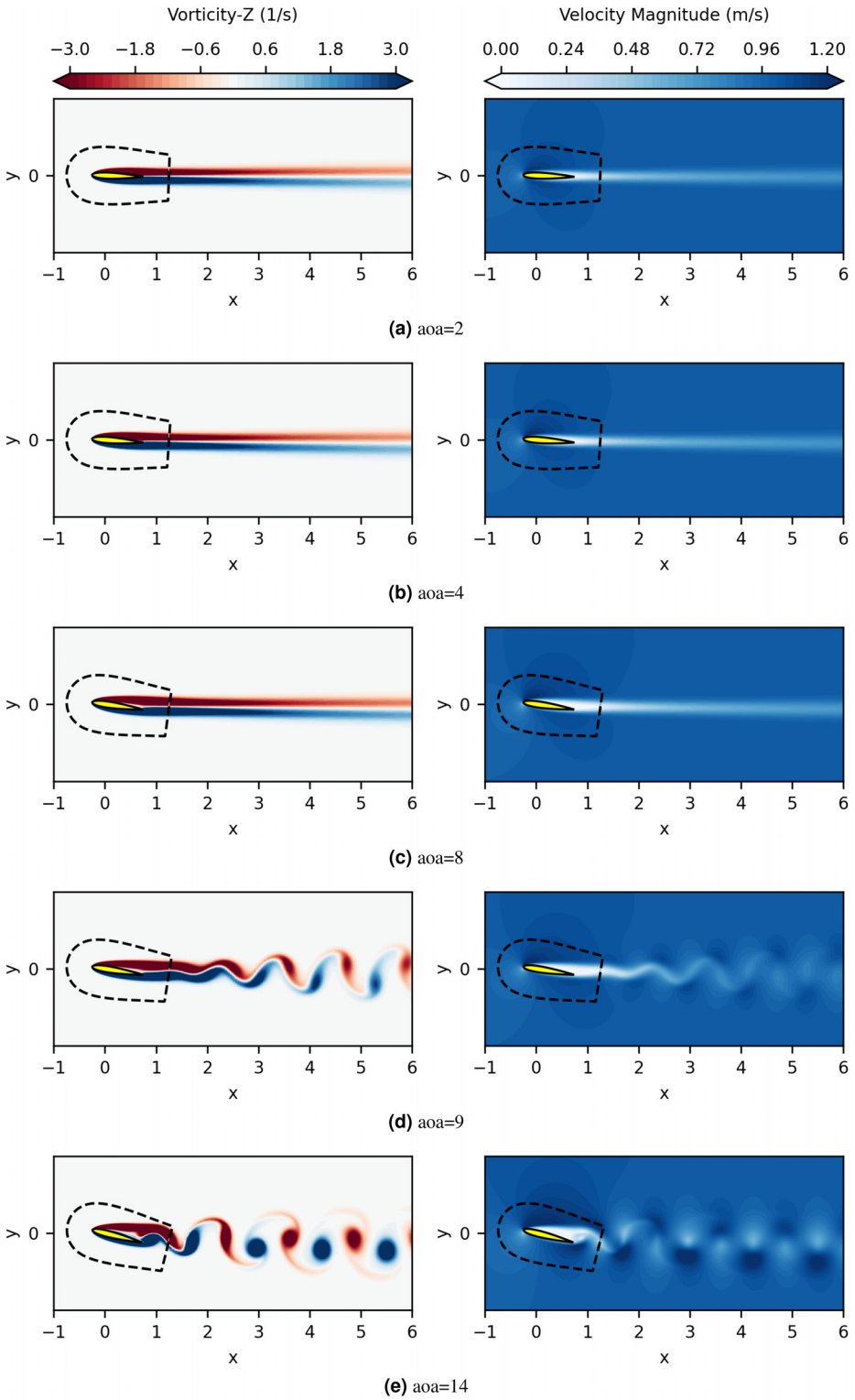
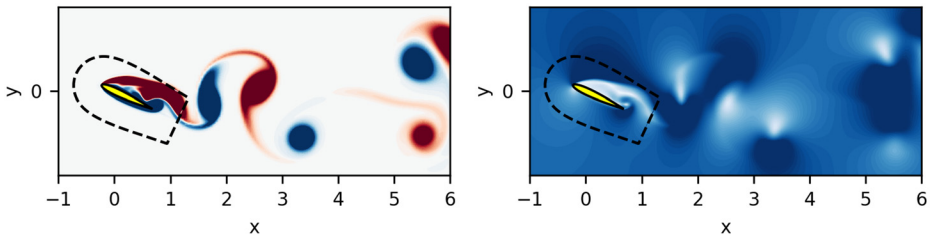
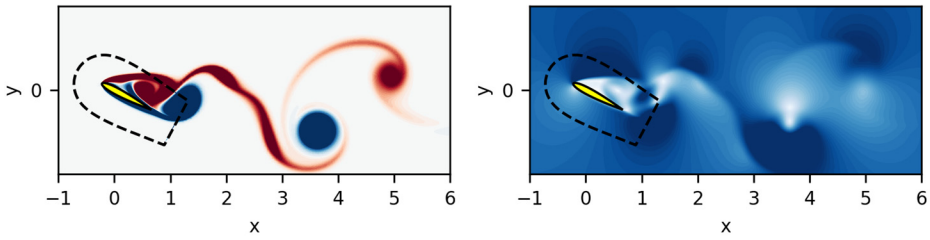


FIG. 4. Flow field of NACA0012 airfoil at $Re = 1000$ for varying angles of attack. The left panels illustrate the vorticity fields, while the right panels display velocity magnitude contours.



(f) aoa=25



(g) aoa=28

FIG. 4. (*Continued.*)

within acceptable limits again. All the solvers predict similarly that the highest C_l/C_d occurs for aoa around 10° to 12°.

IV. APPLICATION—DYNAMIC STALL

Section III shows that the present solver can simulate the flow around the NACA0012 airfoil at $Re = 1000$ and predict the aerodynamic coefficients and the static stall point, aligning closely with published results. Next, we will explore a more complex scenario involving a pitching airfoil. In this case, the airfoil undergoes an oscillatory pitching motion, transitioning between low and high angles of attack. This motion can lead to significant aerodynamic phenomena, such as the dynamic stall of the airfoil. Dynamic stall occurs due to the rapid change in the angle of attack, resulting in a delayed stall onset. This delay leads to increased lift coefficients compared to what would be expected if the airfoil were static and at the same angles of attack.

A. Case configuration

The airfoil is set to oscillate around its aerodynamic center, located a quarter chord ($0.25C$) from the leading edge. This oscillation is characterized by a mean angle of attack (α_0), an oscillation amplitude (α_{amp}), and a reduced frequency ($k = \frac{\omega}{2}\frac{C}{U_{\text{inf}}}$), where C is the chord length and U_{inf} is the freestream velocity. The angle of attack varies over time as follows:

$$aoa = \alpha_0 - \alpha_{amp} \cdot \sin(\omega t). \tag{5}$$

In this analysis, the initial angle of attack is specified as $\alpha_0 = 15°$, with an amplitude also set at 15° ($\alpha_{amp} = 15°$). The investigation will incorporate four distinct reduced frequencies: $k = 0.1, 0.2, 0.4, 0.6$ (Fig. 8).

Experimental data are not present in the bibliography for this Reynolds number, so the results will be compared to those of a pure Eulerian simulation here in OpenFOAM. The computational domain

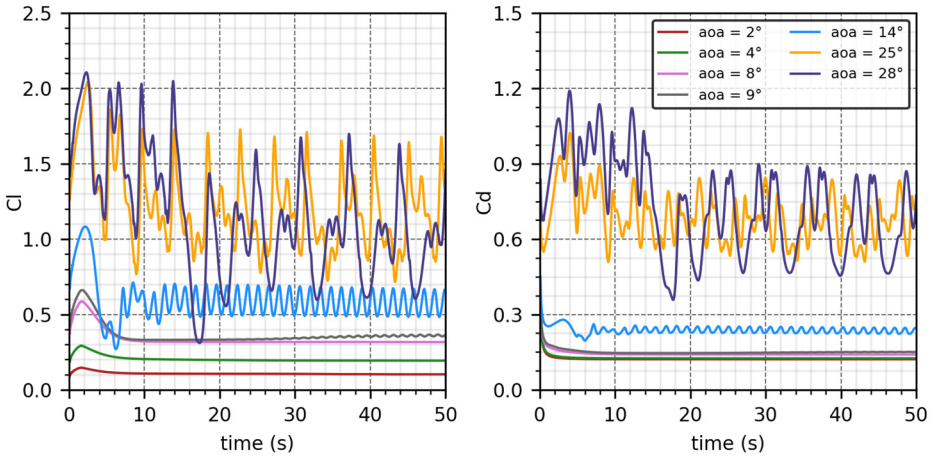


FIG. 5. Aerodynamic coefficients of NACA0012 airfoil at $Re = 1000$ for varying angles of attack. The left plot illustrates the lift coefficient over the aoa, while the right plot displays the drag coefficient over the aoa.

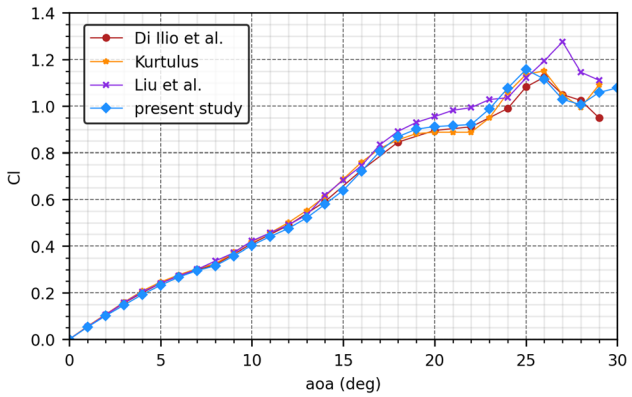


FIG. 6. Lift coefficient as a function of the aoa. The present results are compared with the corresponding results from Kurtulus,[14] Di Ilio *et al.*,[15] and Liu *et al.*[33]

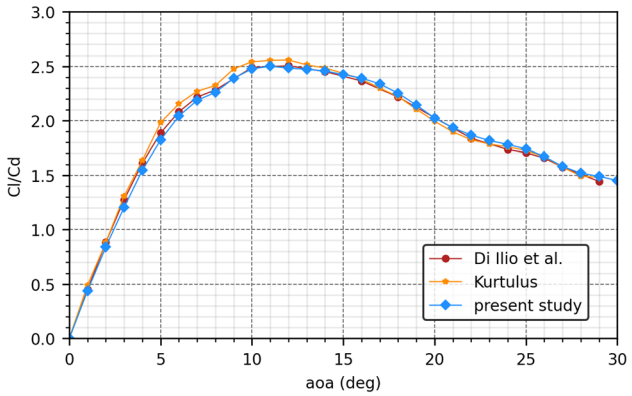


FIG. 7. Ratio of the lift and drag coefficients as a function of the aoa. The present results are compared with the corresponding results from Kurtulus[14] and Di Ilio *et al.*[15]

used in the pure OpenFOAM simulations is illustrated in Fig. 9. The oscillatory rotating motion of the airfoil is modeled using the cyclic AMI feature of OpenFOAM, which is widely used in many applications for airfoil simulations, like the work of Pan *et al.*[35] A mesh and

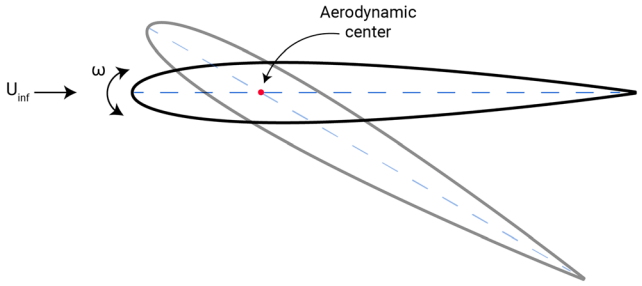


FIG. 8. The NACA0012 airfoil oscillates from the initial aoa a_0, with an angular frequency ω and amplitude a_{amp}.

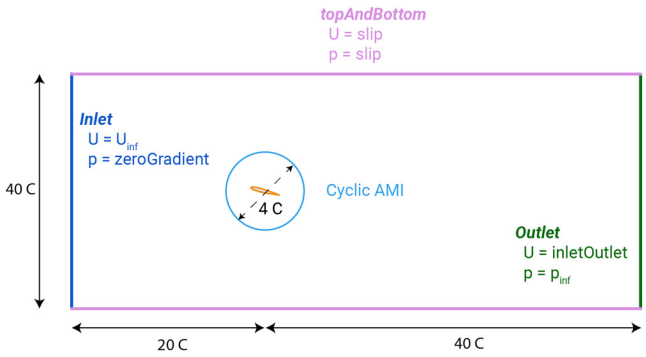


FIG. 9. The computational domain for the pure OpenFOAM simulation of the dynamic stall of a NACA0012 airfoil.

time step convergence test has been also employed for the OpenFOAM case to have a fair comparison of the converged OpenFOAM and hybrid solver solutions. The OpenFOAM results refer to the converged cases for all the cases presented in this section.

Dynamic stall is a highly complex phenomenon affected by various parameters, one of which is the mesh. To ensure consistency between the OpenFOAM solution and the solution obtained from the hybrid solver, we employ a different mesh than that used in the static case. Specifically, we utilize the inner mesh of the full OpenFOAM solution, which, in OpenFOAM, is the region that rotates, as the complete mesh for the hybrid solution. This mesh is depicted in Fig. 10. It is a circular mesh with a radius twice the chord length, created using the snappyHexMesh tool of OpenFOAM. The mesh features four levels of refinement as it approaches the surface.

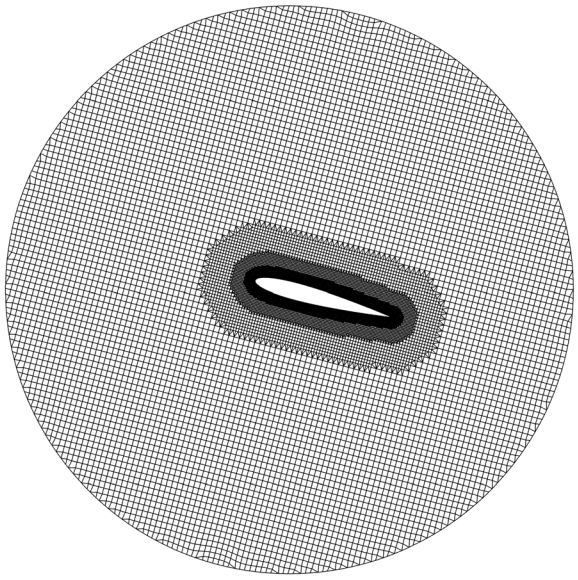


FIG. 10. The inner mesh of the full OpenFOAM simulation, which is here utilized as the entire mesh for the hybrid simulation.

B. Results—Dynamic stall

1. Flow fields

Figure 11 illustrates the vorticity fields for different pitching reduced frequencies and angles of attack. On the left, the results obtained from pure OpenFOAM simulations are depicted, while the results from the present solver are illustrated on the right panel of Fig. 11. Across all scenarios, similar wake structures are observed from both solvers. The only noticeable difference is that in the OpenFOAM case, the wake vortices are somewhat more diffused than those in the

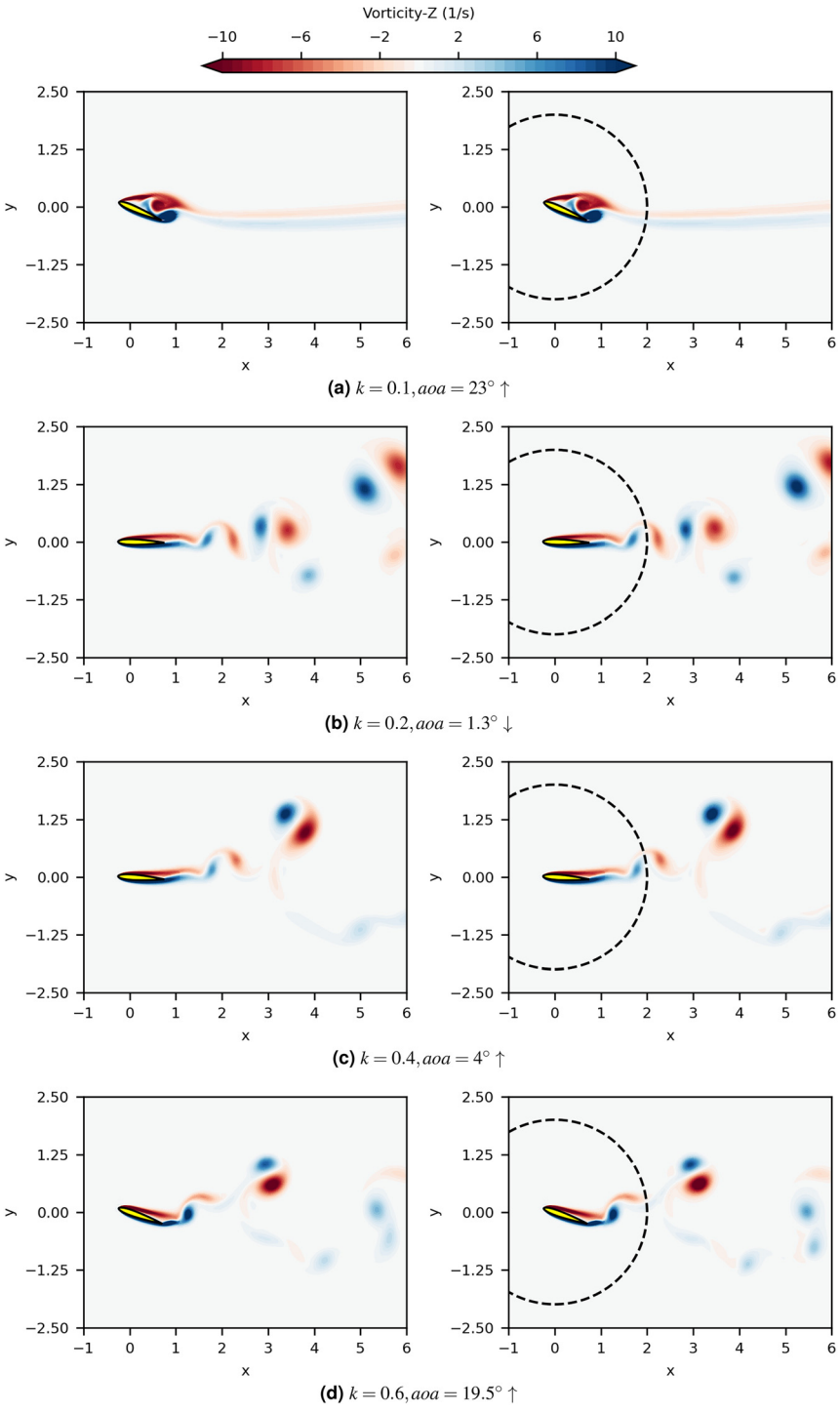


FIG. 11. Vorticity fields for the dynamic case of the NACA0012 airfoil at $Re = 1000$. The contours represent different pitching reduced frequencies and time instances. On the left panel, the results from the present solver are illustrated, while on the right panel, the OpenFOAM results are depicted.

hybrid solver. This was expected, as the vortex particles in the hybrid solver ensure no numerical diffusion is added to the flow. The artificial diffusion into the wake is more obvious in the wake structures depicted in Fig. 12. This is the wake for the case of $k = 0.6$ and aoa $= 19.5°$, with the airfoil ascending. In this contour, the absolute vorticity is illustrated. It can be seen that the wake vortices in the OpenFOAM case (left) are more diffused than the corresponding vortices in the hybrid solver (right). Even with three levels of refinement in the wake for the OpenFOAM case, and despite the aerodynamic coefficients being converged, the artificial diffusion has an impact on the wake structures.

2. Hysteresis loops

Figure 13 depicts the hysteresis loops for the four different reduced frequency cases. The initial observation here is that, as expected, the pitching airfoil can reach higher angles of attack before stalling compared to the static airfoil. This is particularly evident in the cases of $k = 0.4$ and $k = 0.6$, where the airfoil reaches $29° - 30°$ without stalling, while in the static case, stalling occurs between $25°$ and $26°$. As the frequency increases, the airfoil does not have sufficient time to stall, allowing the lift coefficient to increase beyond the static stall point. In lower reduced frequencies, like the case of $k = 0.1$, a more complex situation arises, as depicted in Fig. 13(a), where the airfoil has enough time to adapt to its state.

Now, when comparing the hysteresis loops obtained from the pure OpenFOAM simulations with those from the hybrid solver, no differences are observed. The hysteresis loops for all the reduced frequencies match exactly, demonstrating that the hybrid solver can accurately reproduce the results that OpenFOAM produces. Some spikes present in the hysteresis loops of the current solver are due to the dynamic mesh simulation, where the Eulerian mesh moves relative to the distribution of particles. In some cases, particles close to the solid boundary may end up inside the body in the next time step and are therefore deleted. This phenomenon was studied in Ref. 36.

V. SUMMARY

This study aimed to test the validity of hybrid Eulerian–Lagrangian solvers used in the field of external aerodynamics, particularly for complex fluid phenomena that are closer to real-life scenarios. Specifically, it examined the static and dynamic stall of a NACA0012 airfoil. We utilized the hybrid Eulerian–Lagrangian solver developed in our previous work,[8] a coupled solver that employs OpenFOAM for resolving near-wall phenomena and a vortex particle method (VPM) for the rest of the computational domain. This approach efficiently and accurately evolves the wake without introducing artificial diffusion into the flow. All cases presented were for the same Reynolds number, $Re = 1000$, as turbulence modeling has not yet been incorporated into the hybrid solver.

Initially, the airfoil's static stall was examined across angles of attack ranging from $0°$ to $30°$. The results showed great agreement with the literature used for comparison. Specifically, the present solver predicts the stalling angle to be between $25°$ and $26°$ angle of attack, the same value that was previously predicted by Kurtulus[14] and Di Ilio *et al.*[15]

Following this, the solver was applied to dynamic stall cases, where the airfoil pitches around its aerodynamic center at different reduced frequencies and within the same angle of attack range as before. These results were compared with those obtained from pure OpenFOAM simulations, given the absence of external data for this scenario. In OpenFOAM, the cyclic AMI feature was employed to model the rotation of the airfoil. To ensure a fair comparison between the two solvers, we utilized the rotating region of OpenFOAM as the Eulerian mesh for the hybrid solver. In all cases, the airfoil loading between OpenFOAM and the hybrid solver demonstrated identical patterns, with the hysteresis loops matching exactly. The flow fields produced by the two solvers show excellent agreement, with the only differences being that in the pure OpenFOAM case, some wake vortices appeared more diffused, due to the solver's diffusive nature.

VI. DISCUSSION

The primary advantage of the hybrid solver is its ability to capture wake structures effectively. In the hybrid approach, it is not necessary to resolve the entire region behind the airfoil. The adaptive nature of the particles, which are free to move, allows them to focus on regions with vorticity while leaving the rest unresolved. This reduces computational cost significantly.

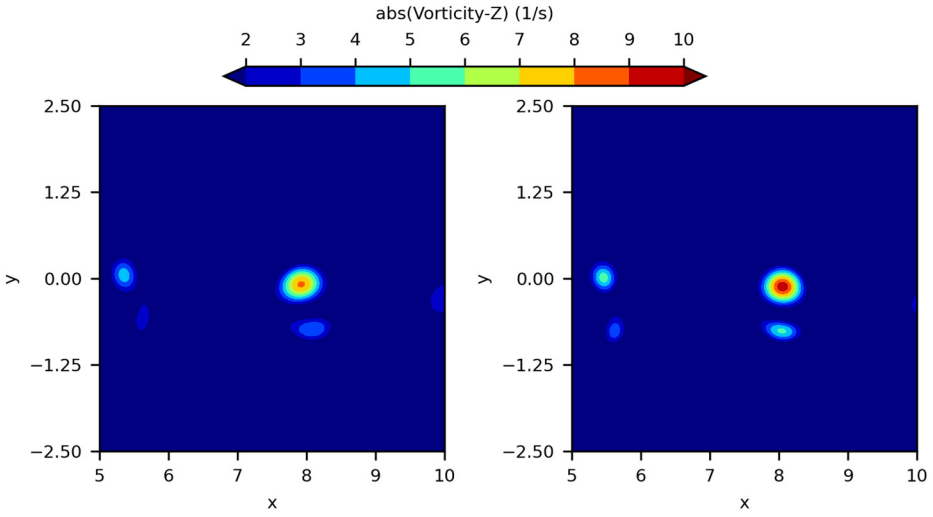


FIG. 12. The wake for the case of $k = 0.6$ and aoa $= 19.5°$, with the airfoil ascending. On the left is the result from OpenFOAM, while on the right is the result from the hybrid solver. The absolute vorticity is plotted.

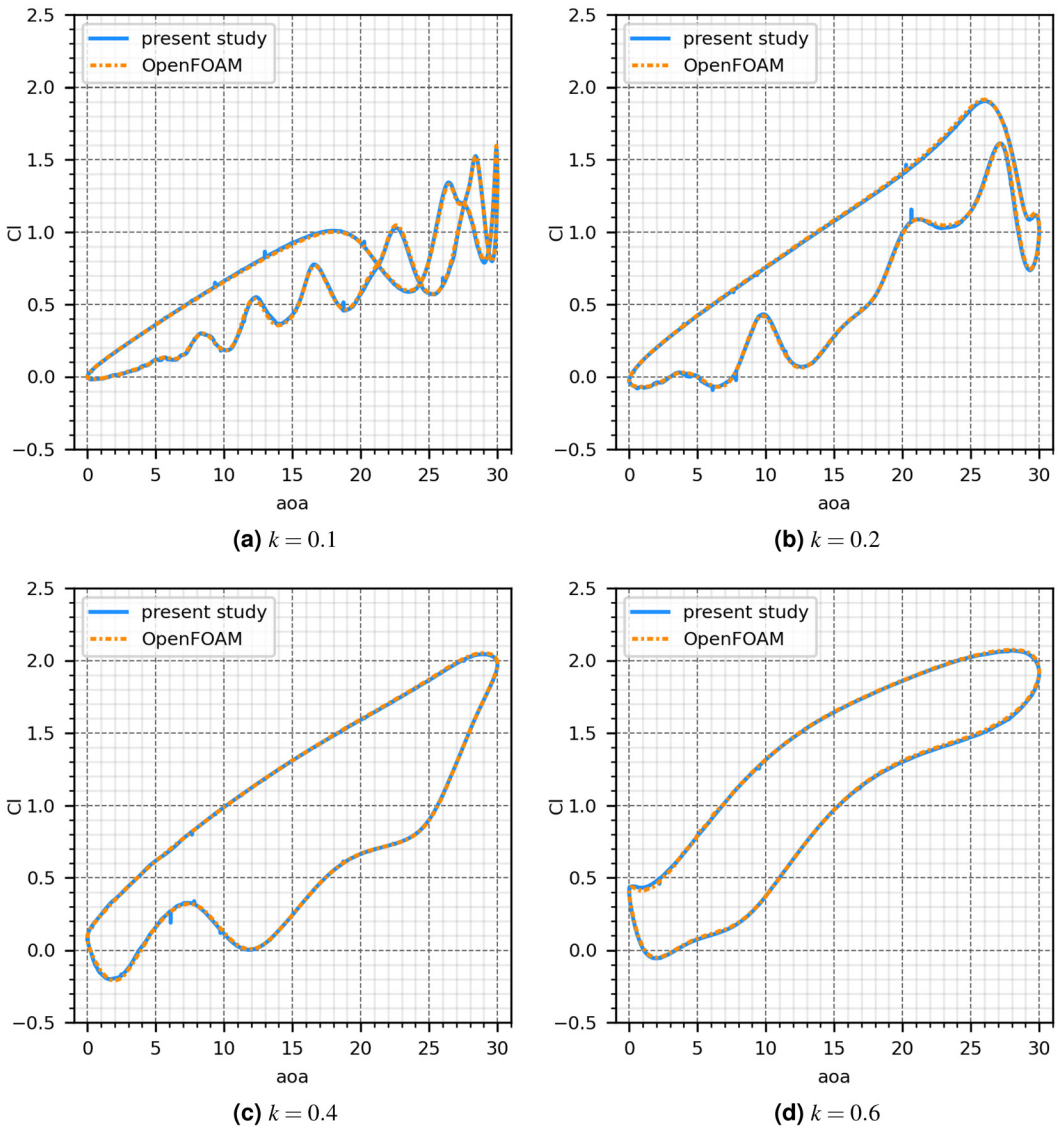


FIG. 13. Hysteresis loops for the dynamic stall case of a NACA0012 airfoil at $Re = 1000$ and reduced frequencies $k = 0.1, 0.2, 0.4, 0.6$.

Moreover, it has been shown that OpenFOAM, despite many levels of refinement in the wake, can achieve convergence on aerodynamic forces but still introduces artificial diffusion into the wake. As a result, the wake is not accurately captured due to the diffusive nature of Eulerian solvers. To accurately capture wake structures, extremely dense and computationally heavy simulations are required. This issue is mitigated in the hybrid solver, which preserves the local vorticity of the structures without introducing numerical diffusion. This is particularly important in multibody cases, where the wakes of different bodies interact, making wake accuracy crucial.

Another advantage of our hybrid solver is its approach to mesh motion. Very small meshes are generated and move as solid bodies, eliminating the need for morphing meshes or AMI. This elegant method of simulating motion is especially beneficial when simulating multiple bodies, as each body can be solved separately by the OpenFOAM component of the hybrid solver, while the particles interconnect the different regions.

The findings presented here suggest that hybrid Eulerian–Lagrangian solvers are capable of simulating complex flow scenarios, establishing them as reliable tools in computational fluid dynamics (CFD). While assessing the solver's efficiency falls outside the scope of this paper, future analysis to evaluate its potential, particularly in wake regions where Lagrangian solvers can achieve comparable results to Eulerian solvers with fewer computational resources, would be valuable. Furthermore, we recommend incorporating turbulence modeling into the hybrid solver as a future step. The solver can leverage existing turbulence modeling modules in OpenFOAM, with the main challenge

being the application of appropriate boundary conditions at the Eulerian field's numerical boundary and addressing turbulence in the Lagrangian solver.

ACKNOWLEDGMENTS

The authors acknowledge the use of computational resources of DelftBlue supercomputer,[37] provided by Delft High Performance Computing Centre (https://www.tudelft.nl/dhpc).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

R. Pasolari: Conceptualization (equal); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead). **C. J. Ferreira:** Conceptualization (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal). **A. van Zuijlen:** Conceptualization (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal).

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

[1]G.-H. Cottet, "A particle-grid superposition method for the Navier-Stokes equations," J. Comput. Phys. **89**(2), 301–318 (1990).
[2]G. Daeninck, "Developments in hybrid approaches: Vortex method with known separation location; vortex method with near-wall Eulerian solver; RANS-LES coupling," Ph.D. thesis (Catholique de Louvain, 2006).
[3]M. J. Stock, A. Gharakhani, and C. P. Stone, "Modeling rotor wakes with a hybrid OVERFLOW-vortex method on a GPU cluster," AIAA Paper No. 2010-4553, 2010.
[4]A. Palha *et al.*, "A hybrid Eulerian-Lagrangian flow solver," arXiv: 1505.03368 (2015).
[5]G. Papadakis and S. G. Voutsinas, "In view of accelerating CFD simulations through coupling with vortex particle approximations," J. Phys.: Conf. Ser. **524**, 012126 (2014).
[6]G. Papadakis and S. G. Voutsinas, "A strongly coupled Eulerian Lagrangian method verified in 2D external compressible flows," Comput. Fluids **195**, 104325 (2019).
[7]P. Billuart *et al.*, "A weak coupling between a near-wall Eulerian solver and a vortex particle-mesh method for the efficient simulation of 2D external flows," J. Comput. Phys. **473**, 111726 (2023).
[8]R. Pasolari, C. Ferreira, and A. van Zuijlen, "Coupling of OpenFOAM with a Lagrangian vortex particle method for external aerodynamic simulations," Phys. Fluids **35**(10), 107115 (2023).
[9]M. J. Stock and A. Gharakhani, "A hybrid high-order vorticity-based Eulerian and Lagrangian vortex particle method, the 2-D case," in *Fluids Engineering Division Summer Meeting* (ASME, 2021).
[10]G. Papadakis, V. A. Riziotis, and S. G. Voutsinas, "A hybrid Lagrangian–Eulerian flow solver applied to elastically mounted cylinders in tandem arrangement," J. Fluids Struct. **113**, 103686 (2022).
[11]H. G. Weller *et al.*, "A tensorial approach to computational continuum mechanics using object oriented techniques," Comput. Phys. **12**(6), 620–631 (1998).
[12]R. Pasolari *et al.*, "Dynamic mesh simulations in OpenFOAM: A hybrid Eulerian–Lagrangian approach," Fluids **9**(2), 51 (2024).
[13]A. Malhotra, A. Gupta, and P. Kumar, "Study of static stall characteristics of a NACA 0012 aerofoil using turbulence modeling," *Innovative Design and Development Practices in Aerospace and Automotive Engineering*, edited by R. P. Bajpai and U. Chandrasekhar (Springer Nature, Singapore, 2017), pp. 369–378.
[14]D. F. Kurtulus, "On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at Re = 1000," Int. J. Micro Air Veh. **7**(3), 301–326 (2015).
[15]G. Di Ilio *et al.*, "Fluid flow around NACA 0012 airfoil at low-Reynolds numbers with hybrid lattice Boltzmann method," Comput. Fluids **166**, 200–208 (2018).
[16]V. D. Nguyen *et al.*, "Low order modeling of dynamic stall using vortex particle method and dynamic mode decomposition," Int. J. Micro Air Veh. **15**, 1 (2023).
[17]C. Zhu, T. Wang, and W. Zhong, "Combined effect of rotational augmentation and dynamic stall on a horizontal axis wind turbine," Energies **12**(8), 1434 (2019).
[18]A.-J. Buchner *et al.*, "Dynamic stall in vertical axis wind turbines: Comparing experiments and computations," J. Wind Eng. Ind. Aerodyn. **146**, 163–171 (2015).
[19]K. Mulleners, K. Kindler, and M. Raffel, "Dynamic stall on a fully equipped helicopter model," Aerosp. Sci. Technol. **19**(1), 72–76 (2012).
[20]J. M. Brandon, "Dynamic stall effects and applications to high performance aircraft," in *Special Course on Aircraft Dynamics at High Angles of Attack: Experiments and Modelling* (Hampton, Virginia, AGARD-R-776, 1991).
[21]S. Wang *et al.*, "Numerical investigations on dynamic stall of low Reynolds number flow around oscillating airfoils," Comput. Fluids **39**(9), 1529–1541 (2010).
[22]M. H. Akbari and S. J. Price, "Simulation of dynamic stall for a NACA 0012 airfoil using a vortex method," J. Fluids Struct. **17**(6), 855–874 (2003).
[23]D. F. Kurtulus, "Unsteady aerodynamics of a pitching NACA 0012 airfoil at low Reynolds number," Int. J. Micro Air Veh. **11**, 1 (2019).
[24]OpenFOAM Foundation, see https://openfoam.org/ for "OpenFOAM-The OpenFOAM Foundation" accessed 22 February 2024.
[25]C. Greenshields and H. Weller, *Notes on Computational Fluid Dynamics: General Principles* (CFD Direct Ltd, Reading, UK, 2022).
[26]F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction With OpenFOAM® and Matlab* (Springer, 2015).
[27]See https://cpp.openfoam.org/v9/classFoam_1_1pimpleLoop.html for "OpenFOAM Foundation. pimpleLoop Class Reference;" accessed 22 February 2024.
[28]See https://openfoam.org/version/9/ for "OpenFOAM Foundation. OpenFOAM Version 9 (2020);" accessed 22 February 2024.
[29]A. Goude and S. Engblom, "Adaptive fast multipole methods on the GPU," J. Supercomput. **63**(3), 897–918 (2013).
[30]A. J. Chorin, "Numerical study of slightly viscous flow," J. Fluid Mech. **57**(4), 785–796 (1973).
[31]G.-H. Cottet and P. Koumoutsakos, *Vortex Methods—Theory and Practice* (Cambridge University Press, 2000).
[32]C. Mimeau and I. Mortazavi, "A review of vortex methods and their applications: From creation to recent advances," Fluids **6**(2), 68 (2021).
[33]Y. Liu *et al.*, "Numerical bifurcation analysis of static stall of airfoil and dynamic stall under unsteady perturbation," Commun. Nonlinear Sci. Numer. Simul. **17**(8), 3427–3434 (2012).
[34]C. Geuzaine and o R. Jean-Franc, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," Numer. Meth. Eng. **79**(11), 1309–1331 (2009).
[35]J. Pan, C. Ferreira, and A. van Zuijlen, "A numerical study on the blade–vortex interaction of a two-dimensional Darrieus–Savonius combined vertical axis wind turbine," Phys. Fluids **35**(12), 125152 (2023).
[36]R. Pasolari *et al.*, "Flow over traveling and rotating cylinders using a hybrid Eulerian–Lagrangian solver," Comput. Fluids **279**, 106327 (2024).
[37]TU Delft, see https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1 for "DelftBlue Supercomputer (Phase 1), Delft High Performance Computing Centre (DHPC)" (2022).