Flow over traveling and rotating cylinders using a hybrid Eulerian–Lagrangian solver

Pasolari, R.; Pan, J.; Ferreira, Carlos; van Zuijlen, A.H.

DOI
10.1016/j.compfluid.2024.106327

Publication date
2024

Document Version
Final published version

Published in
Computers & Fluids

Computers and Fluids 279 (2024) 106327

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Flow over traveling and rotating cylinders using a hybrid Eulerian–Lagrangian solver

R. Pasolari [*], J. Pan, C.J. Ferreira, A. van Zuijlen

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

ARTICLE INFO

Keywords:
Hybrid solver
Eulerian–Lagrangian solver
Coupling
Dynamic mesh
OpenFOAM
Magnus effect

ABSTRACT

Hybrid Eulerian–Lagrangian solvers have gained increasing attention in the field of external aerodynamics, particularly when dealing with strong body–vortex interactions. This approach effectively combines the strengths of the Eulerian component, which accurately resolves boundary layer phenomena, and the Lagrangian component, which efficiently evolves the wake downstream. This study builds on our team's previous work by enhancing the capabilities of a two-dimensional hybrid Eulerian–Lagrangian solver. We aim to upgrade our solver which was initially designed for static cases, to now also simulate cases involving moving objects. To ensure the reliability and applicability of a new solver, it is essential to validate its performance in complex cases. Here, the solver is validated across the case of a traveling cylinder and the case of a rotating cylinder in two different rotational speeds at low Reynolds numbers. In the realm of Eulerian solvers, such as OpenFOAM (utilized for the Eulerian component of this hybrid approach), traditional techniques include the use of morphing meshes, overset meshes, and Arbitrary Mesh Interfaces (AMI) to model body motion. The proposed methodology involves extending the Eulerian mesh up to a short distance from the solid boundary and moving it entirely as a solid entity. Then the Lagrangian solver is responsible for calculating the updated boundary conditions, thereby completing the hybrid solver's functionality. This approach is very similar to the overset mesh technique. However, unlike the traditional method where an Eulerian mesh moves on top of a static one, our method involves the motion of an Eulerian mesh over a Lagrangian grid. We compared the results from our hybrid solver with those from a purely Eulerian solver, specifically OpenFOAM. The comparison demonstrates that our solver can replicate OpenFOAM's results with high accuracy. Another interesting point highlighted in this study is the presence of high-frequency oscillations in the body forces in hybrid solvers that incorporate the redistribution of Lagrangian particles and do not utilize surface elements such as vortex panels, specifically when dealing with dynamic mesh simulations. When the Eulerian mesh travels on top of the Lagrangian grid of particles, the positions of the particles with respect to the Eulerian mesh continuously change. This results in a constant shift of particles near the solid body, where the highest vorticity is observed. Particles that are close to the solid boundary at one time step may find themselves inside the boundary at the next time step, leading to their removal. This pattern continuously changes during the simulation, causing fluctuations in the boundary conditions of the Eulerian solver and manifesting as oscillations in the forces acting on the body. It is shown that this issue can be alleviated either by increasing the spatial resolution of the Lagrangian solver or by synchronizing the movement of the Lagrangian grid with the motion of the Eulerian mesh. The results of the study make the solver trustworthy and pave the way for more demanding external aerodynamic simulations.

1. Introduction

Hybrid Eulerian–Lagrangian solvers have gained significant attention in the field of external aerodynamics. These solvers offer a combination of advantages from both Eulerian and Lagrangian solvers while mitigating many of their inherent limitations. In particular, hybrid solvers employ a Eulerian solver to accurately capture near-wall phenomena such as boundary layers, separation, and aerodynamic forces. Meanwhile, the Lagrangian solver takes charge of the wake evolution downstream. This approach ensures that the region close to the body is adequately resolved while minimizing numerical diffusion and significantly reducing computational cost.

Several hybrid solvers have been introduced to date. In our previous work [1], OpenFOAM [2] was coupled with a Vortex Particle Method (VPM), using the Domain Decomposition Technique in the

[*] Corresponding author.
E-mail address: r.pasolari@tudelft.nl (R. Pasolari).

https://doi.org/10.1016/j.compfluid.2024.106327
Received 25 March 2024; Received in revised form 30 April 2024; Accepted 27 May 2024
Available online 8 June 2024

form presented by Daeninck [3]. Similarly, Palha et al. [4] combined FEniCS with the same VPM. Another approach was presented by Stock and Gharakhani [5], who coupled a high-order spectral finite difference method with an open-source VPM. Furthermore, Billuart et al. [6] developed a weak coupling approach between a body-fitted velocity–pressure solver and a Vortex Particle-Mesh method in two dimensions. Finally, Papadakis and Voutsinas [7] integrated a compressible Eulerian solver (MaPFlow) with a Lagrangian solver using a Vortex Particle-Mesh technique.

Dealing with moving, rigid objects poses one of the most challenging and crucial aspects of Computational Fluid Dynamics (CFD). In CFD simulations, aerodynamic objects can undergo forced motion (prescribed motion) or move freely, like the case of Vortex Induced Vibrations (VIV). The primary focus here is on forced motions and, more specifically, translation and rotation. Simulations involving moving objects require high-fidelity methods to accurately capture all the phenomena involved. In the case of Eulerian solvers like OpenFOAM, morphing or overset meshes are commonly employed to address translation and oscillation, while AMI is used for rotating objects. A comparison of morphing and overset techniques in OpenFOAM was conducted by Alletto [8] for the case of forced and VIV inline oscillations of a cylinder. An application of the AMI technique can be found in [9], where a Savonius rotor is simulated. However, Eulerian solvers face significant challenges: they demand extensive computational resources, particularly for simulations involving moving objects, and they introduce artificial diffusion into the flow in areas where the mesh lacks high resolution.

Lagrangian methods have also been employed for moving body simulations. Alvarez and Ning [10] used VPM for modeling multirotor aerodynamic interactions, while the propeller was modeled as a rotary lifting surface. Karimi-Zindashti and Kurç [11] used a deterministic vortex method along with vortex panels to simulate the flow around rotating cylinders. However, Lagrangian solvers struggle to accurately resolve regions near solid boundaries, often requiring additional solvers for support.

The methods mentioned previously fall short in delivering both cost-effective and accurate results for moving body simulations simultaneously. Hybrid solvers, which merge the strengths of both Eulerian and Lagrangian approaches, are introduced as a promising alternative. In dynamic mesh simulations, hybrid solvers offer the advantage of enabling the entire Eulerian mesh to move as a solid body while the Lagrangian particles handle the computation of updated boundary conditions, as Papadakis et al. [12] achieved for their compressible hybrid solver. This feature becomes particularly valuable when dealing with multiple bodies, as each case's mesh can move independently, with the linkage between cases achieved solely through the Lagrangian particles. For example, in the case of a wind turbine, each blade can be represented as an independent Eulerian case with its own mesh, which can move independently as a solid body.

This paper extends our previous work presented in [1] by including the dynamic mesh capabilities of the solver. For every new solver developed, it is necessary to validate it thoroughly on different well-examined scenarios before applying it to demanding real-life problems. This study aims to establish the accuracy of the hybrid Eulerian–Lagrangian solver in simulations including non-stationary Eulerian domains. Furthermore, this paper addresses a challenge encountered in the development of solvers that utilize the Vortex Redistribution Method (VRM), similar to the solver described here, specifically those that do not use surface elements such as vortex panels. During dynamic mesh simulations, the Eulerian mesh moves while the particles remain redistributed at fixed points on an underlying grid. Consequently, the relative position of the Eulerian mesh with respect to the vortex particles is in constant flux, leading to shifts in the distribution of particles near the solid boundary. These particles are crucial as they carry the highest vorticity of the flow, and their shifting can induce artificial high-frequency oscillations in the boundary conditions, which may in turn cause oscillations in the body forces. It is demonstrated that these oscillations can be mitigated by either increasing the resolution of the vortex particles or by synchronizing the movement of the Lagrangian grid with that of the Eulerian mesh.

The paper is organized as follows: Section 2 provides a brief overview of the main components of the hybrid solver, along with the coupling strategy employed. Section 3 starts the discussion of the validation of the solver. Section 3.1 provides the results for the traveling cylinder case, which serves as a first validation of the hybrid code before proceeding to the more demanding motions. Section 3.2 discusses the results of the rotational case, which is very interesting to examine since interest phenomena appear. Pure Eulerian simulations in OpenFOAM are also presented here and used to validate the hybrid code. Moreover, the obtained results will be compared with the corresponding results from the bibliography. Finally, Section 4 provides the paper's conclusions and a discussion on the potential capabilities of the solver.

2. Hybrid solver

In this paper, the hybrid solver is a coupling of a Lagrangian Vortex Particle Method solver, and an Eulerian Finite Volume Method implemented in the OpenFOAM software. Here, a brief reference to the solver is done. The reader can find more information about the specific solver in [1,13] and about other similar hybrid solvers in [4,6,12,14,15].

The hybrid solver is based on the Domain Decomposition Technique introduced by Cottet [16] and later improved by Daeninck [3]. The way that the computational domain is decomposed can be seen in Fig. 1. This figure shows that the Eulerian mesh extends up to the numerical boundary, covering the region close to the solid walls. This ensures that the boundary layer effects are resolved sufficiently. The Lagrangian solver covers the entire computational domain, with its primary goal being the wake's evolution downstream. The nature of the solver poses a barrier to resolving the boundary layer effects, so the Lagrangian solution is corrected in a region close to the solid body, bounded by the interpolation boundary, using the Eulerian solution.

2.1. Lagrangian solver

The Lagrangian solvers have shown great potential in the field of external aerodynamics, especially in cases where strong body–vortex interactions are present. In the present paper, the Lagrangian solver chosen is the Vortex Particle Method (VPM). A complete analysis of the method can be found in [17], while Mimeau and Mortazavi [18] presents a detailed review, including all the advances that have been made in the field.

In VPM, the Navier–Stokes (N–S) equations are written in terms of velocity ($\boldsymbol{u}$) and vorticity (ω). In the two-dimensional space, and for the case of an incompressible flow, the set of equations that are solved can be summarized as:

$$\frac{D\omega}{Dt} = \nu\nabla^2\omega \qquad \text{N-S equations in 2D} \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{incompressibility constraint} \tag{1b}$$

$$\nabla \times \boldsymbol{u} = \omega \qquad \text{velocity-vorticity relation} \tag{1c}$$

$$\omega(\boldsymbol{x}, t) = \omega_0(\boldsymbol{x}) \qquad \text{initial vorticity} \tag{1d}$$

while their boundary conditions are:

$$\lim_{|\boldsymbol{x}|\to\infty} \boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{U_{inf}} \qquad \text{velocity at infinity} \tag{2a}$$

$$\lim_{|\boldsymbol{x}|\to\infty} \omega(\boldsymbol{x}, t) = 0 \qquad \text{vorticity at infinity} \tag{2b}$$

The original formulation of the vortex particles gives a singularity point at its center since the induced velocity is infinite. To avoid

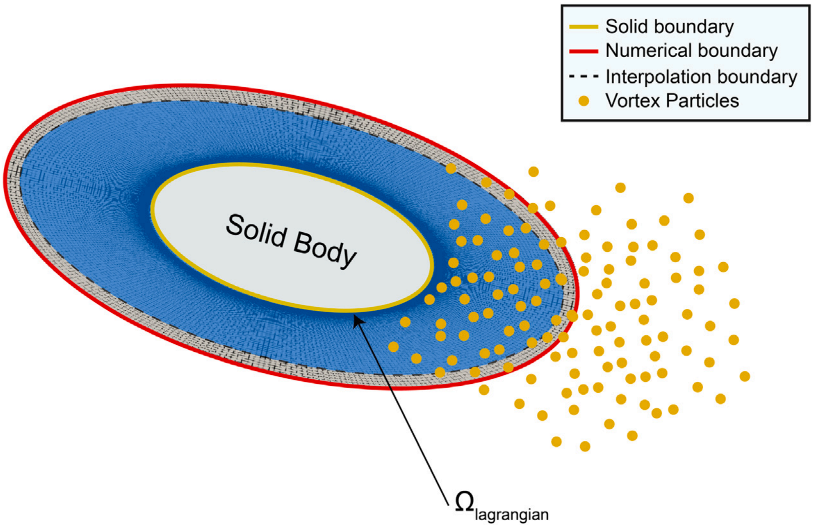

Fig. 1. Decomposition of the computational domain. The Lagrangian solver covers the entire computational domain (denoted as $\Omega_{lagrangian}$), whereas the Eulerian mesh extends up to a short distance (numerical boundary) away from the solid boundary. The under-resolved solution of the Lagrangian is corrected using the Eulerian solution in a region bounded by the interpolation boundary.

this singularity, mollified particles with finite cores are used instead. The velocity and the vorticity fields can be written as summations of the contributions of all the particles (in the velocity field, the free-stream velocity $\boldsymbol{U}_{inf}$ is added) because both are linear solutions of a Poisson equation. Hence, the induced velocity and vorticity fields can be written as:

$$\boldsymbol{u}_p(\boldsymbol{x}) = -\frac{1}{2\pi}\sum_p \frac{g_\sigma(\left|\boldsymbol{x}-\boldsymbol{x}_p\right|)}{\left|\boldsymbol{x}-\boldsymbol{x}_p\right|^2}(\boldsymbol{x}-\boldsymbol{x}_p)\times e_z\Gamma_p + \boldsymbol{U}_{\text{inf}} \tag{3a}$$

$$\omega_p(\boldsymbol{x}) = \sum_p \zeta_\sigma(\left|\boldsymbol{x}-\boldsymbol{x}_p\right|)\Gamma_p \tag{3b}$$

where the smoothing functions are:

$$g_\sigma(r) = \frac{1}{2\pi\sigma^2}\left(1 - e^{-\dfrac{r^2}{2\sigma^2}}\right) \quad \text{and} \quad \zeta_\sigma(r) = \frac{1}{2\pi\sigma^2}e^{-\dfrac{r^2}{2\sigma^2}} \tag{4}$$

The vortex particles are evolved in two distinct steps, as proposed by Chorin [19]. First, the particles are advected using a 4th order Runge–Kutta integration scheme and then are diffused using the Vortex Redistribution Method (VRM) [20]. The VRM involves redistributing particles across a predefined grid, which is beneficial for two main reasons. Firstly, it addresses the issue of particle deformation due to flow strains, which can lead to regions either devoid of particles or with high concentrations, causing inaccuracies in the flow. Secondly, the Vortex Particle Method (VPM) provides a mechanism to model the diffusion process effectively.

2.2. Coupling strategy

The two solvers are coupled in two steps. The first step of the coupling is the evaluation of the boundary conditions for the Eulerian solver at the numerical boundary (see Fig. 1). In OpenFOAM, we need to specify velocity and pressure boundary conditions, and so the Lagrangian particles are used here to calculate the velocity and the pressure gradient on the boundary faces. The velocity components are calculated using the expression of the induced velocity (Eq. (3)), while the pressure gradient is retrieved using the unsteady Bernoulli equation (Eq. (5)).

$$\nabla\bar{p} = -(\frac{\partial\boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nu\nabla^2\boldsymbol{u}), \qquad \bar{p} = p/\rho \tag{5}$$

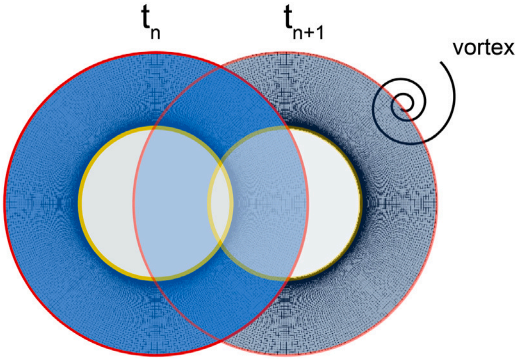

Fig. 2. A vortex that inserts in the Eulerian subdomain when the Eulerian mesh moves towards it. This is just a graphical representation for clarifying purposes since the body cannot cover this distance in one time step.

The second step of the coupling involves correcting the Lagrangian solution within the interpolation region (blue region in Fig. 1), using the more accurate Eulerian solution. For this correction, the technique proposed by Billuart et al. [6] is employed. Specifically, instead of interpolating the vorticity field from the Eulerian solver to the Lagrangian, the velocity field is interpolated at the edges of a square around each Lagrangian particle. Subsequently, the strength of each particle is determined by integrating the velocity around these edges. As Billuart et al. [6] explain, this method significantly enhances the accuracy of circulation conservation. It also accommodates the motion of solid bodies, as velocity values can be assigned to edges that are within the body.

In OpenFOAM, two main techniques are utilized for conducting simulations involving moving bodies, namely morphing mesh and overset mesh, while AMI is very often used for rotating cases. In the case of morphing mesh, the mesh's topology undergoes alterations by displacing the patch associated with the moving object while preserving the connectivity of the internal cells. Conversely, in the overset mesh approach, a stationary background mesh is employed, and for each moving object, an additional mobile mesh is superimposed on top of the former. This additional mesh moves about the background mesh, and their interaction is achieved through the interpolation of variables between them. The approach that we employ here is quite similar to

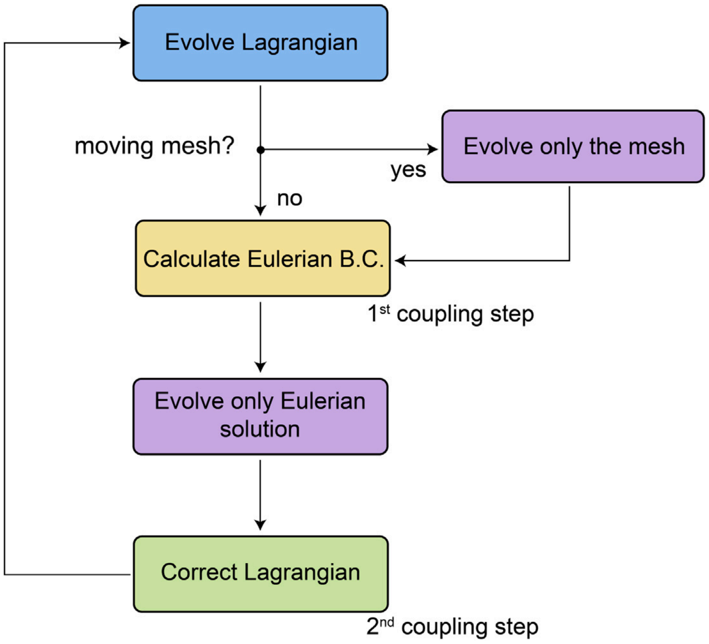

Fig. 3. The flowchart of the hybrid solver, including the mesh update, assumes that both solvers are at time t_n, and their solutions are correct. The first step is to evolve the Lagrangian solver to t_{n+1}. Then, if the body is moving, the mesh is moved; otherwise, the calculation of the Eulerian boundary conditions takes place directly. Subsequently, the Eulerian solution is evolved to t_{n+1}, and finally, the Lagrangian solver is corrected inside the interpolation region.

the overset mesh approach, but instead of a static Eulerian mesh, we employ a background Lagrangian grid.

In this scenario, a question arises regarding how the Eulerian solver recognizes fluid structures, such as vortices, outside its boundary. For example, consider a scenario where a vortex exists beyond the Eulerian domain's boundaries, and the mesh is moving towards it, as depicted in Fig. 2.

The Lagrangian solver effectively addresses this issue by updating the boundary conditions at the new coordinates after the movement. Consequently, updating the position of the Eulerian mesh before calculating the boundary conditions is essential. As a result, the evolution of the Eulerian solver can be divided into two distinct steps. Initially, the mesh undergoes motion and updates the coordinates of cells, faces, and vertices. Subsequently, in a separate step, all operations related to correcting the fields due to the movement and the solution's evolution are performed. Compared to the flowchart presented in [1] where static cases were examined, the updated flowchart is illustrated in Fig. 3.

In this project, OpenFOAM v9 [21] is employed as the Eulerian component of the hybrid solver. Specifically, the solver in this context is pimpleFOAM, an incompressible, transient solver capable of accommodating dynamic mesh simulations. pimpleFOAM uses the PIMPLE algorithm [22] for correcting the velocity and pressure fields to enforce the continuity equation. Modifications have been done to the original pimpleFOAM solver in order to achieve the coupling between the two solver. A detailed discussion on these changes can be found in [13].

3. Validation

The solver is validated through the traveling cylinder case at $Re = 100$ and the rotating cylinder at two different rotating speeds at $Re = 200$. A combination of these two motions can describe every motion a rigid body can undergo, so it is vital to validate both to have a complete analysis of the two-dimensional dynamic case.

The exact configuration is employed for the different cases, as depicted in Fig. 4. The simulated cylinder has a radius R_c, and the Eulerian domain extends up to a distance of R_E from the center of the cylinder (the red circle represents the numerical boundary). The Lagrangian solver covers the entire computational domain and is corrected within the interpolation region Ω_{int} (illustrated in blue). Notably, a small layer of cells is excluded from the correction process, denoted as d_{Bdry}. The Eulerian mesh is exclusively comprised of hexahedra, as detailed in the snippet. Mesh density increases near the cylinder's surface to capture near-wall phenomena effectively, with the stretching ratio being 1.05 and the height of the first cell 2.7 mm (for the base mesh presented below).

Table 1
Global simulation parameters for the validation cases.

Parameter	Symbol	Value	Dimension
Cylinder's radius	R_C	0.5	m
Eulerian's domain radius	R_E	1.0	m
Simulation time	t_{sim}	100	s
Gaussian kernel width spreading	k	2	–
Overlap ratio	λ	1	–
Interpolation domain offset from Eulerian boundary	d_{bdry}	0.09	m

Many parameters between the different cases are similar, and they can be summarized in Table 1. The parameters that differentiate among the cases are mentioned in the corresponding subsections.

3.1. Traveling cylinder at Re = 100

The first case to validate the solver is the traveling cylinder. The solver has already been validated into the case of a stationary cylinder with a free-stream velocity in [1], but in that case, for $Re = 550$. Here, the validation case presents the opposite situation. The free-stream velocity is set to zero while the cylinder travels with a constant velocity U_{mesh} as shown in Fig. 5 operating at $Re = 100$. In order to have a fair comparison between the static and the dynamic case, the static case at $Re = 100$ is also simulated here. The initial simulations are executed using a base Eulerian mesh, which consists of 11.040 hexahedra, and is depicted in Fig. 5.

A preliminary comparison of the aerodynamic coefficients from the two simulations (static and dynamic) is illustrated in Fig. 6. While the

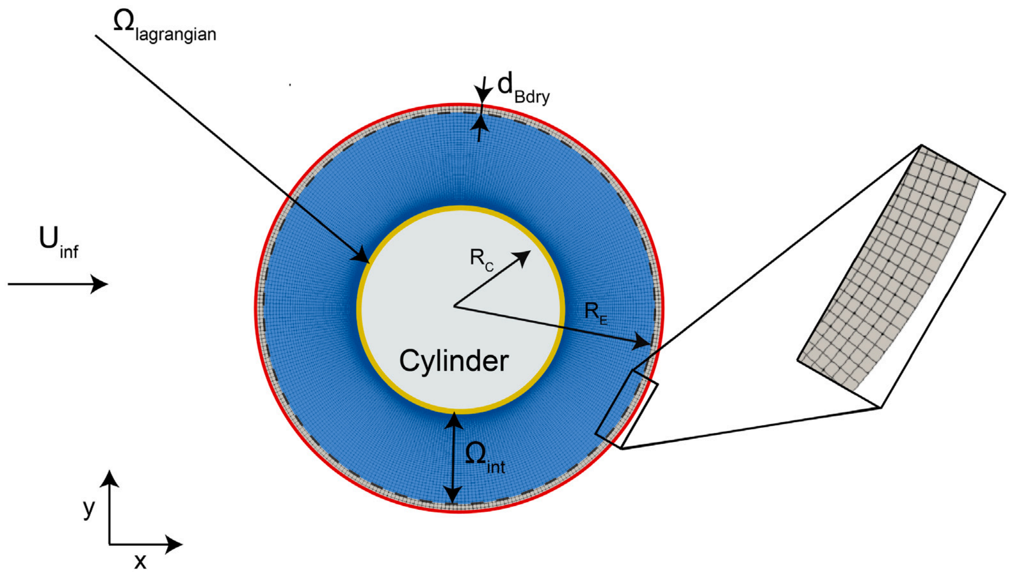

Fig. 4. The cylinder configuration used for the validation cases, with a snippet depicting the cells close to the numerical boundary.

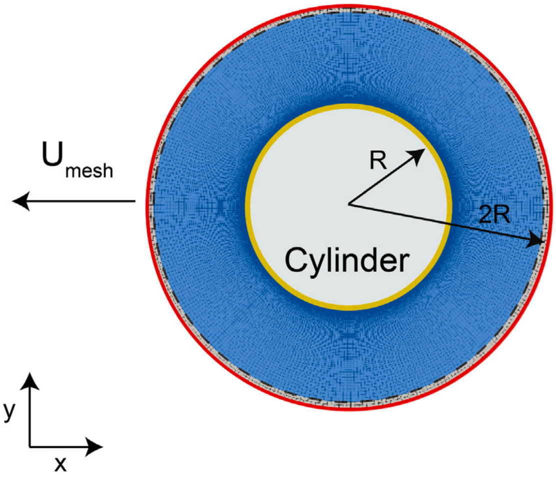

Fig. 5. The traveling cylinder case configuration. The cylinders moves in the $-x$ direction with a constant velocity U_{mesh}.

Table 2
Test cases of the traveling cylinder at $Re = 100$, to demonstrate the alleviation of the high-frequency oscillations on the body forces.

Case state	Particle spacing (h)	Eulerian mesh	Synchronization
Static	0.030 m	Base	–
Static	0.015 m	Base	–
Dynamic	0.030 m	Base	False
Dynamic	0.030 m	Base	True
Dynamic	0.015 m	Base	False
Dynamic	0.015 m	Base	True

mean values of the drag and lift coefficients appear similar, the dynamic simulation exhibits noticeable high-frequency oscillations, particularly in the drag coefficient time history, with significant amplitude. Upon detailed analysis, it was determined that these oscillations are caused by the movement of the Eulerian mesh over the Lagrangian particles, which are consistently redistributed at the same grid points in global coordinates. This interaction results in a continuous shift of particles near the solid body, where the highest vorticity is recorded, relative to the Eulerian mesh, as shown in Fig. 7. Consequently, particles close to the solid body at one time step may end up inside the body at the next, necessitating their removal. This alteration in the distribution of the particles carrying the highest vorticity affects the boundary conditions of the Eulerian solver, thereby inducing oscillations in the forces acting on the body.

However, this study demonstrates that these oscillations can be mitigated in two ways. First, by increasing the resolution of the Lagrangian solver, we achieve a denser distribution of particles around the solid body and reduce the relative distances as the mesh moves. Second, by moving the redistribution points of the Lagrangian particles in such a way that this movement is synchronized with the Eulerian mesh motion. To validate these approaches, we conducted the simulations summarized in Table 2.

Fig. 8 displays the first 30 s of the drag coefficient for the six different cases outlined in Table 2. On the left, the cases with a spacing of $h = 0.03$ m are shown, and on the right, those for $h = 0.015$ m. Initially, the left subfigure demonstrates that synchronizing the particles with the mesh movement significantly reduces the amplitude of the oscillations. Further reductions are observed when the particle spacing is decreased, as evidenced by the comparison between the left blue line and the right one. Moreover, if we also synchronize the particles with the mesh movement in addition to reducing the spacing, the oscillations become very minimal, closely aligning with the static case's results.

It should be noted that residual oscillations remain due to the continuous interpolation of the particles at different grid points during the synchronization. This is evidenced by running a simulation where the mesh movement per time step matches the particle spacing, ensuring that the particle distribution relative to the solid body remains constant. For this test case, we increased the timestep to 0.02 s for both static and dynamic scenarios to ensure a fair comparison. The outcomes of this analysis are depicted in Fig. 9.

Now that the issue has been addressed, we can proceed to the validation of the results for the hybrid dynamic solver. It is important to note that the remaining simulations for the traveling cylinder case involve the synchronization of the Lagrangian grid with the Eulerian mesh. The results are initially compared with those obtained from the static case, where the cylinder is stationary and a free-stream velocity is applied. Subsequently, the dynamic case results are compared with those from a simulation performed purely in OpenFOAM (static simulation), and finally with bibliographical numerical results from Pingjian and Wenping [23].

Additionally, to perform a convergence test for the Eulerian mesh used in the hybrid simulations, two different meshes, named base

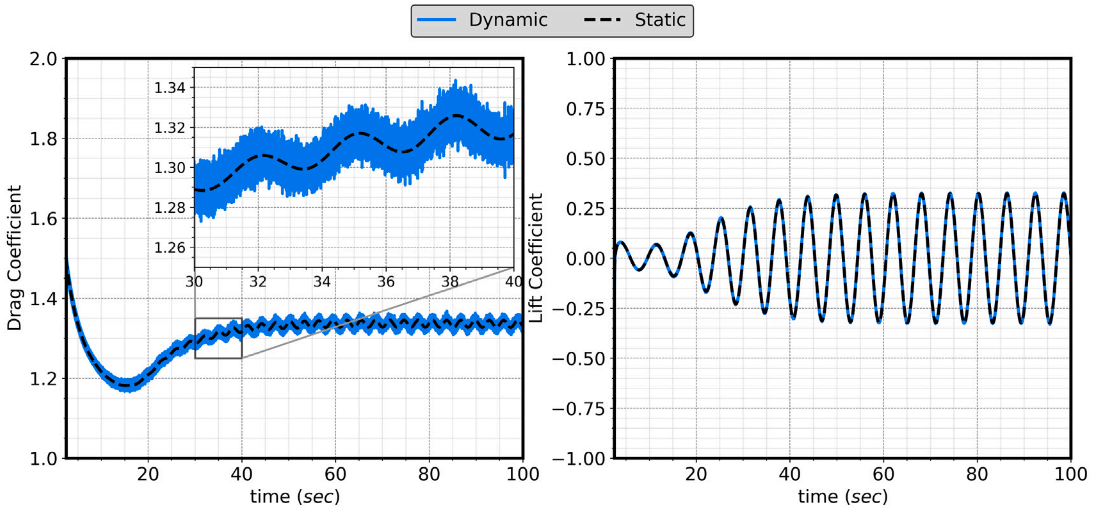

Fig. 6. Lift and drag coefficients over time for the case of the static and traveling cylinder at $Re = 100$, with the base Eulerian mesh and particle spacing $h = 0.03$ m.

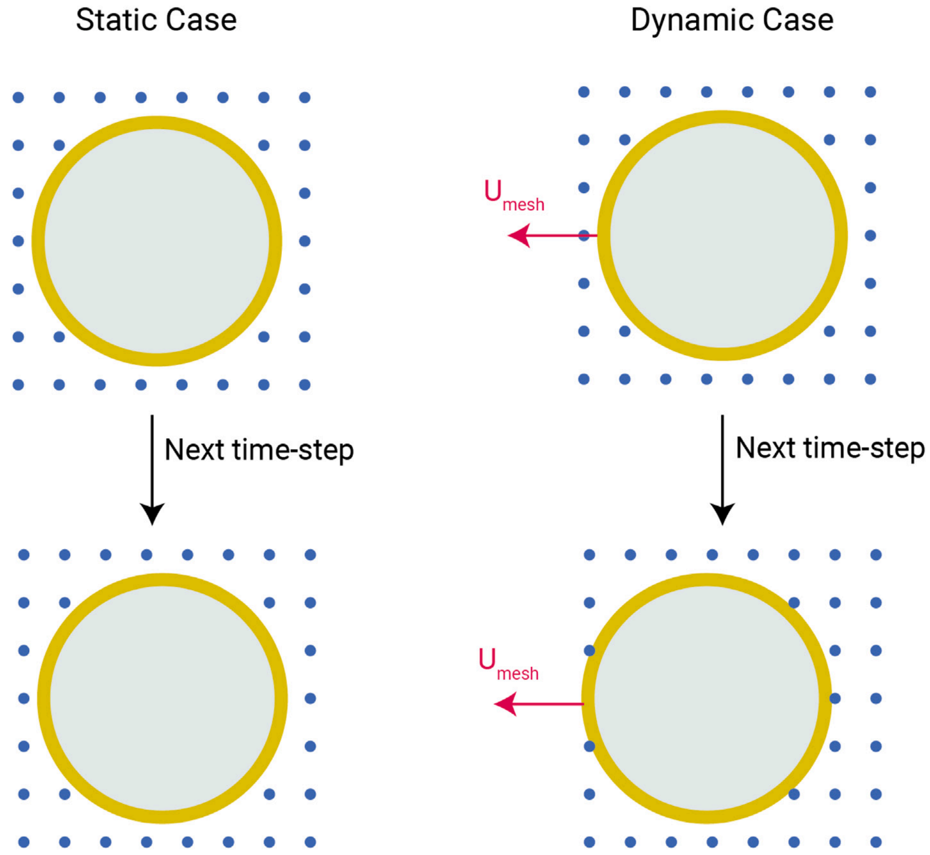

Fig. 7. Configuration of the Lagrangian particles close to the solid boundary over a time step. In the static case, the configuration remains the same; for the dynamic case, the distribution of the particles around the solid boundary changes, leading to high-frequency oscillations in the calculation of the Eulerian boundary conditions and, subsequently, on the body forces.

and refined mesh, are employed. These are summarized in Table 3, alongside the particle spacing parameter and the time step used in the simulations. Table 4 presents the results for the drag coefficient C_d, the lift coefficient C_l, and the Strouhal number. Fig. 10 shows the drag and lift coefficients over time for the present method, the static hybrid case, and the OpenFOAM case.

Finally, Fig. 11 shows the vorticity field in two different instances as the cylinder travels in time. The left image is at the time $t = 20$ s and the right at $t = 100$ s where the center of the cylinder is at

Table 3
Simulation parameters for the case of the traveling cylinder at $Re = 100$.

Case	Eulerian mesh	Particle spacing (h)	Time step
Hybrid static	Base (11040)	0.03 m	0.005 s
Hybrid static	Refined (27600)	0.015 m	0.0025 s
Hybrid dynamic	Base (11040)	0.03 m	0.005 s
Hybrid dynamic	Refined (27600)	0.015 m	0.0025 s

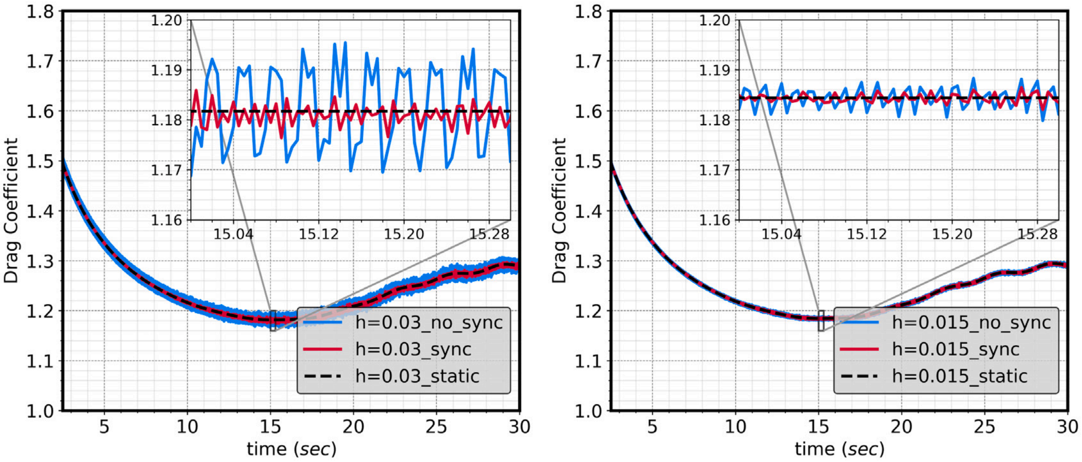

Fig. 8. Drag coefficient over time for different test cases of the traveling cylinder at Re = 100, that demonstrate the alleviation of the high-frequency oscillations on the body forces.

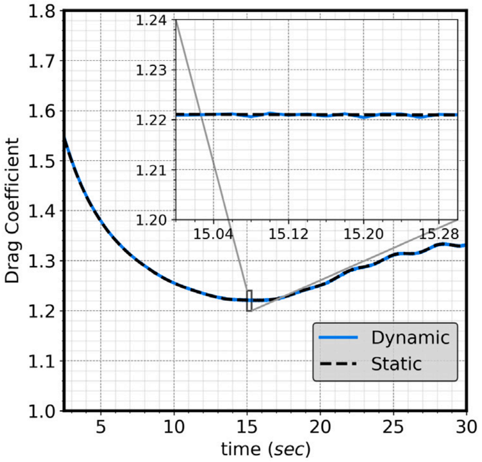

Fig. 9. Comparison of the drag coefficient between a static and a dynamic case where the mesh displacement is equal to the particles' spacing.

Table 4
Results for the case of the traveling cylinder at $Re = 100$.

Method	C_d	C_l	Strouhal
Pingjian and Wenping [23]	1.340 ± 0.008	±0.3130	0.165
OpenFOAM	1.339 ± 0.008	±0.3090	0.164
Hybrid static (base)	1.337 ± 0.009	±0.3225	0.165
Hybrid static (refined)	1.337 ± 0.009	±0.3223	0.165
Hybrid dynamic (base)	1.337 ± 0.017	±0.3264	0.165
Hybrid dynamic (refined)	1.337 ± 0.009	±0.3223	0.165

$x_c = -20$ m and $x_c = -100$ m, respectively. The numerical boundary is represented as a black dashed circle. The vorticity field is smooth, without any inaccuracies to be observed when the vortices generated on the cylinder's surface exit the Eulerian domain.

The above results demonstrate a strong agreement between the hybrid dynamic solver, the static one, and OpenFOAM. Specifically, the hybrid dynamic solver converges to the same values as the hybrid static solver for the lift and drag coefficients, as well as the Strouhal number. When compared with the pure OpenFOAM results, the errors on the lift and drag coefficients are less than 4.0% and 0.2% respectively. Meanwhile, when compared with the Ref. [23], the corresponding errors are 2.8% and 0.22%, respectively. Finally, the hybrid solver predicts the same Strouhal number as the reference, equal to 0.165, while the pure OpenFOAM value is slightly lower, at 0.164.

3.2. Rotating cylinder at Re = 200

The second validation case is the flow around a rotating cylinder at $Re = 200$. In this case, while the cylinder is spinning through the moving air, it experiences a lift force, known as the Magnus effect. There is a dimensionless parameter that characterizes the flow and relates the tangential velocity on the surface of the cylinder with the free-stream velocity, and it is expressed as:

$$\alpha = \frac{\Omega_{rot} R}{U_{inf}} \tag{6}$$

where Ω_{rot} is the rotating speed, R_c is the radius of the cylinder and U_{inf} is the free-stream velocity. This parameter is crucial for the rotating cylinder case since it characterizes the flow. Mittal and Kumar [24] presented the results for the case of $Re = 200$ for value of α from 0.0 to 5.0. It is essential that they showed that a von Kármán vortex street is developed in the flow up to $\alpha = 1.91$. Up to that point, there is a deflection of the wake, but the periodic vortex street is present. The vortex street is not present for values larger than 1.91, and a steady solution is reached. A second region of instability is observed again at $\alpha = 4.4$, while for $\alpha \geq 4.8$, multiple solutions emerge, and the flow becomes more complicated. Karimi-Zindashti and Kurç [11] examined the case of rotating circular cylinders at $0.0 \leq \alpha \leq 5.5$ and rotating square cylinders at $0.0 \leq \alpha \leq 5.0$ using a deterministic vortex method. Here, the hybrid solver will be tested in two different cases, specifically $\alpha = 0.5$ and $\alpha = 2.5$, to have a case before and after the steady solution is reached. The results are compared with those in [24], as well as with pure OpenFOAM simulations. For the OpenFOAM simulations, AMI is used. The case configuration can be seen in Fig. 12. This figure illustrates the rotational speed as Ω_{mesh}.

3.2.1. Case $\alpha = 0.5$

For the case of $\alpha = 0.5$, a von Kármán vortex street is expected, as Mittal and Kumar [24] state. Table 5 shows the results for the minimum and maximum lift coefficient, and for the mean drag coefficient, for the present method, as well as the results obtained in [24], and the results obtained by OpenFOAM simulations. For the case of $\alpha = 0.5$, the base mesh of Table 3 is used.

Fig. 13 shows the drag and lift coefficients for the hybrid case and the OpenFOAM simulations. Moreover, Fig. 14 shows the vorticity field obtained by the hybrid solver simulations for two different time instances: one at the beginning of the simulation (t = 10 s) and one when a periodicity in the wake is reached (t = 60 s). As was expected, for this case, a von Kármán vortex street is present in the wake of the flow and deflected compared to the case of a non-rotating cylinder.

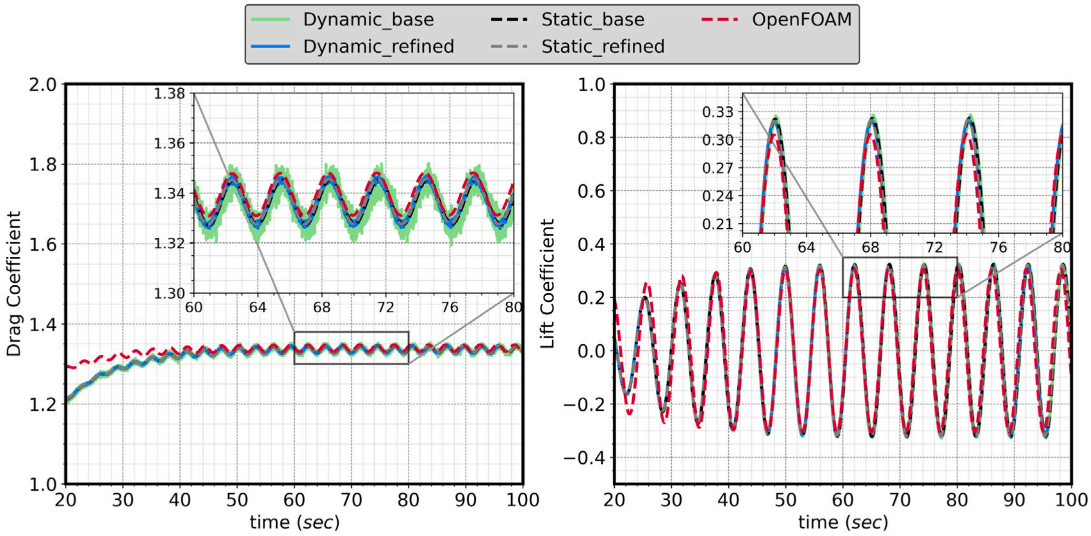

Fig. 10. Lift and drag coefficient over time for the case of the traveling cylinder at $Re = 100$.

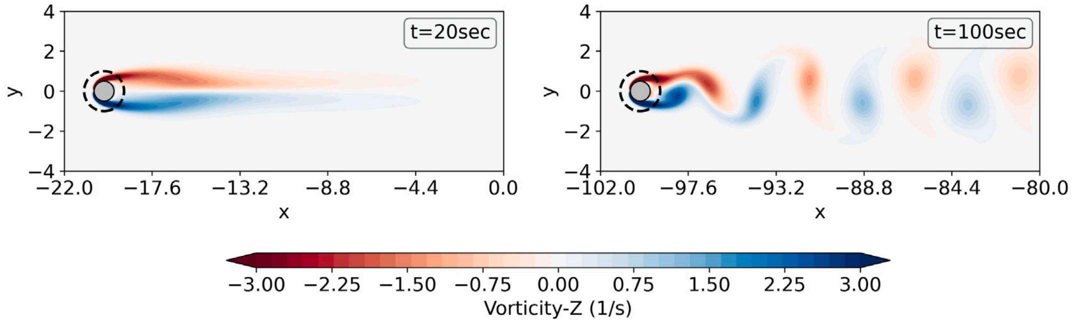

Fig. 11. Vorticity field for the case of the traveling cylinder at $Re = 100$. The numerical boundary is represented with a black dashed circle line.

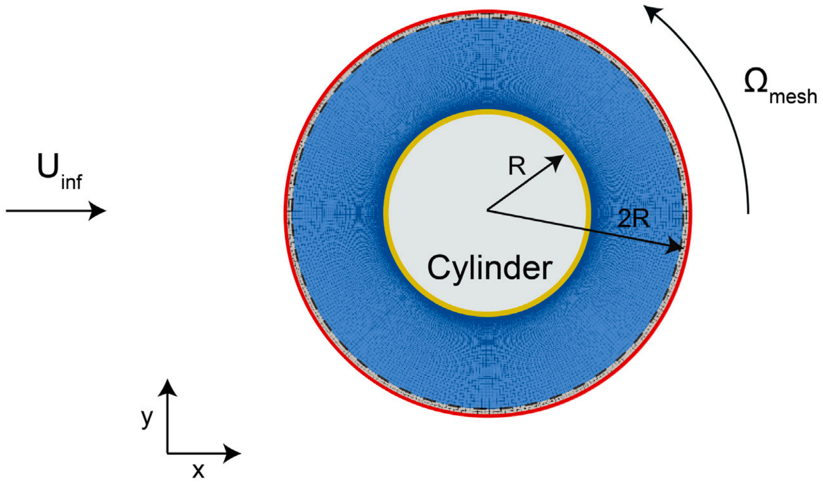

Fig. 12. The rotating cylinder case configuration. The cylinder rotates counterclockwise with a rotational speed Ω_{mesh}, while a free-stream velocity U_{inf} acts in the $+x$ direction.

Table 5
Results for the case of the rotating cylinder at $Re = 200$ and $\alpha = 0.5$.

Case	$C_{l,min}$	$C_{l,max}$	$C_{d,mean}$
Mittal and Kumar [24]	−1.910	−0.487	1.255
OpenFOAM	−1.805	−0.577	1.283
Hybrid	−1.853	−0.520	1.276

3.2.2. Case $\alpha = 2.5$

For the case of $\alpha = 2.5$, the von Kármán vortex street is not expected in the flow, as Mittal and Kumar [24] state. The wake should be steady and deflected. In this simulation, the refined mesh of Table 3 is used, while a time step convergence test is conducted, with the time step to vary from 0.004 s to 0.00025 s. Table 6 shows the results for the steady lift coefficient developed on the rotating body for the present method,

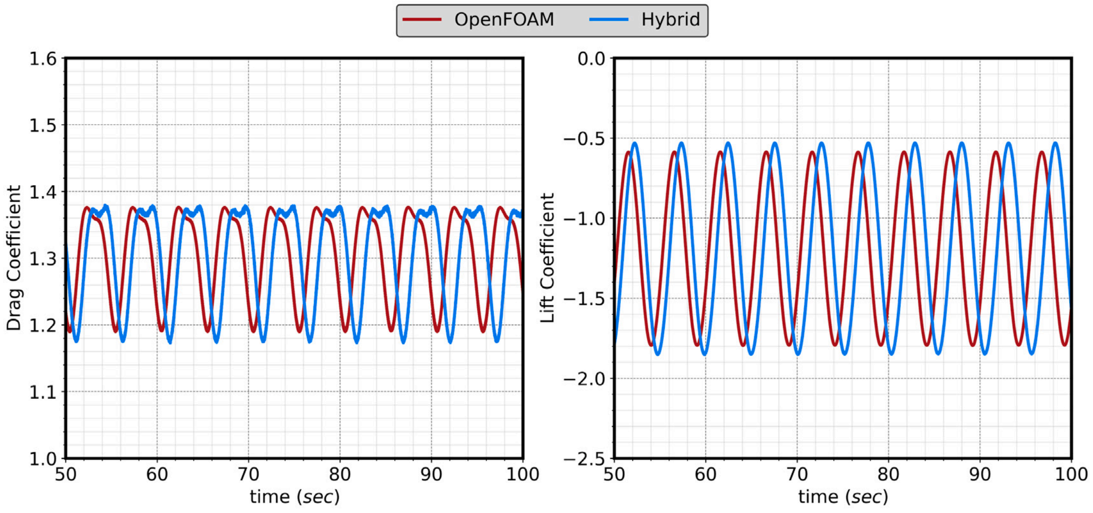

Fig. 13. Lift and drag coefficients over time for the case of the rotating cylinder at $Re = 200$ and $\alpha = 0.5$.

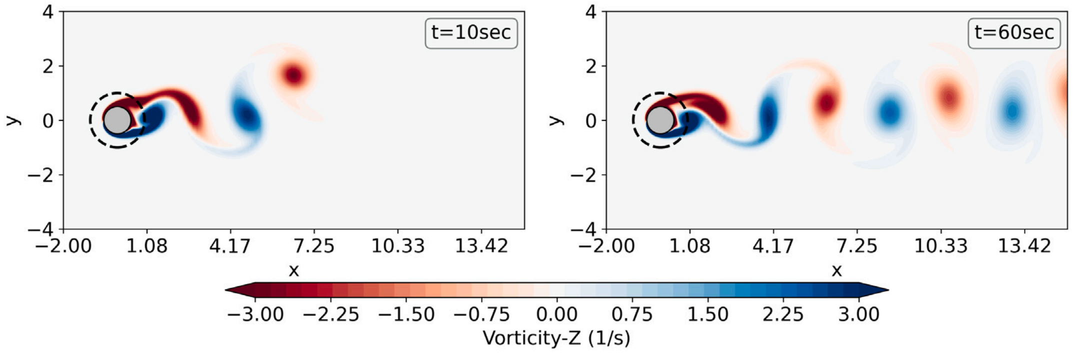

Fig. 14. Vorticity field for the case of the rotating cylinder at $Re = 200$ and $\alpha = 0.5$. The numerical boundary is represented with a black dashed circle line.

as well as the results obtained in [11,24], and the results obtained by OpenFOAM simulations.

Fig. 15 shows the lift coefficient and a time step convergence test for the lift coefficient. It can be seen that the hybrid solver has good agreement with the reference results from Mittal and Kumar [24], with the converged value deviating from the reference value by only 1.2%. Finally, Fig. 16 shows the vorticity field in two different time instances: one at the beginning of the simulation and one when the steady wake has been reached.

It should be noted that in this case, the solver does not experience the issue of high-frequency oscillations in the body forces. This is because the motion of the Eulerian mesh is purely rotational around its center, ensuring that the distribution of the Lagrangian particles around the solid boundary remains constant throughout the simulation.

4. Discussion & future aim

In this study, the 2D incompressible hybrid Eulerian–Lagrangian solver introduced in [1] has been extended to dynamic mesh applications. In the hybrid configuration, the Eulerian solver is responsible for providing a high-fidelity solution in the proximity of the solid object, while the Lagrangian solver is responsible for evolving the wake downstream, being much less diffusive than conventional CFD [17]. Instead of the most common methods used in OpenFOAM for simulating dynamic meshes, such as morphing meshes, overset meshes, and AMI, a different strategy is employed. The entire Eulerian mesh moves as a solid region in respect to the prescribed motion, with the Lagrangian particles being responsible for providing boundary conditions to the Eulerian solver.

Table 6
Results for the case of the rotating cylinder at $Re = 200$ and $\alpha = 2.5$.

Case	C_l
Mittal and Kumar [24]	−7.680
Karimi-Zindashti and Kurç [11]	−7.016
OpenFOAM	−7.410
Hybrid ($\Delta_t = 0.004$ s)	−6.915
Hybrid ($\Delta_t = 0.001$ s)	−7.532
Hybrid ($\Delta_t = 0.00025$ s)	−7.583

The solver has been validated through cases involving a traveling and a rotating cylinder. While the hybrid code had already been validated in cases of flow around a cylinder at low Reynolds numbers [1], this time, no free-stream velocity was applied; instead, the cylinder is traveling through the air. This seemingly simple scenario provided valuable insights into a numerical phenomenon that arises in hybrid solvers using a redistribution method when simulating moving bodies. Specifically, in hybrid solvers that do not use any surface elements to capture the wall vorticity in the Lagrangian solver, such as in the present study and the work by Billuart et al. [6], a constant shift in the distribution of particles near the solid boundary occurs as the mesh moves through space (see Fig. 7). The Eulerian mesh, along with the solid body, moves, while the particles are always redistributed at specific points on a Lagrangian grid, altering their distribution continuously. These particles, carrying the highest vorticity of the flow, significantly impact the boundary conditions. The shift in these particles changes the way vorticity is transferred from the Eulerian

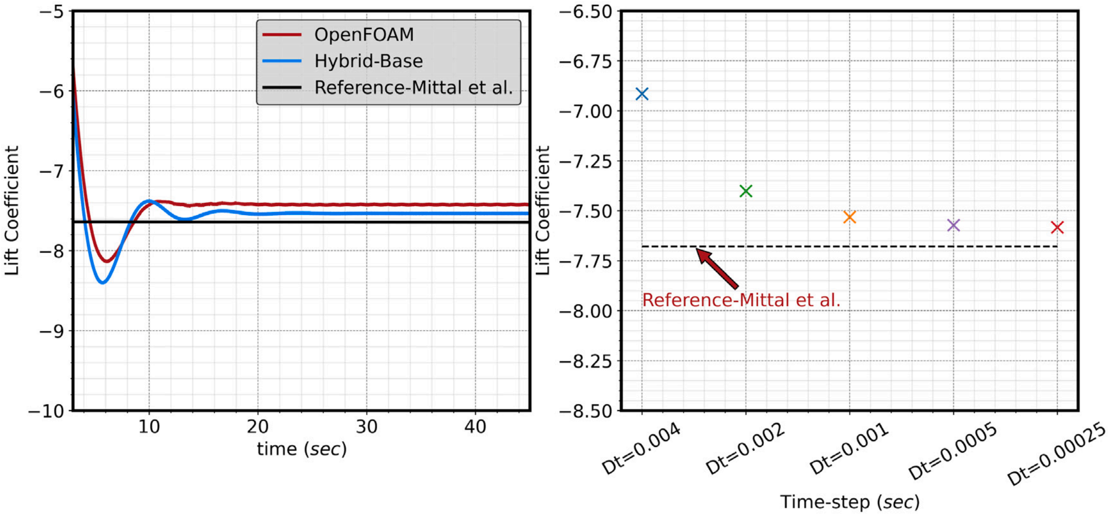

Fig. 15. On the left is the lift coefficient over time for the hybrid case, the OpenFOAM case, and the Ref. [24]. The time step convergence test for the lift coefficient is on the right compared to the Ref. [24].

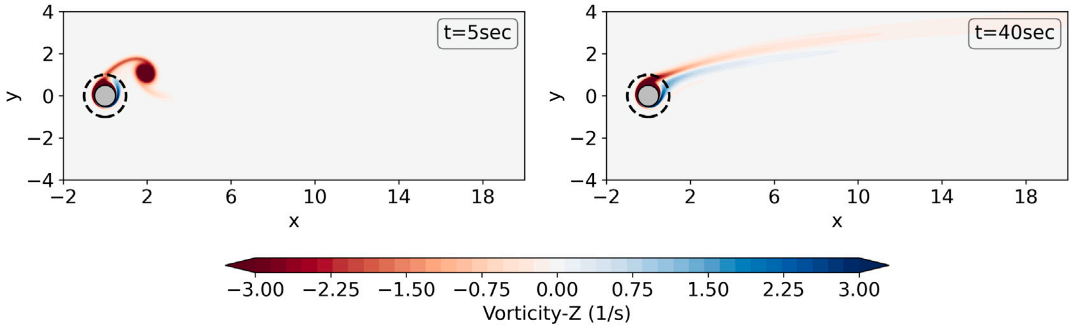

Fig. 16. Vorticity field for the case of the rotating cylinder at $Re = 200$ and $\alpha = 2.5$. The numerical boundary is represented with a black dashed circle line.

solver to the Lagrangian solver, consequently affecting the Eulerian boundary conditions. This results in high-frequency oscillations in the boundary conditions, and thus in the forces acting on the body. Although it was shown that these oscillations do not alter the mean value of the drag coefficient, they significantly affect the amplitude of the oscillations. Nevertheless, it was demonstrated that these oscillations can be significantly reduced either by increasing the resolution of the particles or by moving the particle grid to synchronize its movement with that of the Eulerian mesh.

For the traveling cylinder case, the comparison of the dynamic capabilities of the solver with the static solver, OpenFOAM, and a reference simulation demonstrated that the hybrid dynamic solver is capable of reproducing the exact results as the static solver. When compared to OpenFOAM and the reference, all discrepancies in the results are less than 4.0%.

For the rotating cylinder case, the solver has been validated in two different scenarios, with $\alpha = 0.5$ and $\alpha = 2.5$. In the first case a deflected von Kármán vortex street is present, exactly as stated by Mittal and Kumar [24]. The aerodynamic coefficients show good agreement with both the OpenFOAM results and the reference results [24]. In the case where $\alpha = 2.5$, the von Kármán vortex street is not present, as stated again by Mittal and Kumar [24] and Karimi-Zindashti and Kurç [11]. The wake is steady and deflected by the symmetry axis. Due to the Magnus effect, the cylinder experiences a lift force (negative in the presented results), which aligns well with the results presented by the references and OpenFOAM. This case was much more sensitive to the time step selection, necessitating the use of a convergence test. It should be noted that in this case, no high-frequency oscillations were present, since the Eulerian mesh rotates around its center point, which coincides with the center of the cylinder. Consequently, the distribution of the particles around it remains constant throughout the simulation.

It can be concluded that the solver is capable of simulating moving bodies without difficulty and reproducing results from Eulerian solvers. This capability paves the way for hybrid solvers in the simulation of external aerodynamics. The hybrid solver offers an elegant approach to conducting multi-body simulations, as each body can be represented and solved as an independent case, while particles interconnect them [12]. The solver also needs to be tested in Fluid–Structure Interaction (FSI) problems, where the cylinders can move freely due to the fluid forces acting on them. Later, soft bodies can be incorporated to provide a more comprehensive study of FSI.

CRediT authorship contribution statement

R. Pasolari: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **J. Pan:** Writing – review & editing, Visualization, Validation, Software, Investigation, Data curation. **C.J. Ferreira:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Conceptualization. **A. van Zuijlen:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors acknowledge the use of computational resources of DelftBlue supercomputer [25], provided by Delft High Performance Computing Centre (https://www.tudelft.nl/dhpc).

References

[1] Pasolari R, Ferreira C, van Zuijlen A. Coupling of OpenFOAM with a Lagrangian vortex particle method for external aerodynamic simulations. Phys Fluids 2023;35(10):107115. http://dx.doi.org/10.1063/5.0165878.

[2] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620–31. http://dx.doi.org/10.1063/1.168744.

[3] Daeninck Goéric. Developments in hybrid approaches: vortex method with known separation location; vortex method with near-wall Eulerian solver; RANS-LES coupling [Ph.D. thesis], Université Catholique de Louvain; 2006.

[4] Palha Artur, Manickathan Lento, Ferreira Carlos Simao, van Bussel Gerard. A hybrid Eulerian-Lagrangian flow solver. 2015, arXiv:1505.03368.

[5] Stock Mark J, Gharakhani Adrin. A hybrid high-order vorticity-based Eulerian and Lagrangian vortex particle method, the 2-D case. In: Fluids engineering division summer meeting, Volume 1: Aerospace Engineering Division Joint Track; Computational Fluid Dynamics, 2021, http://dx.doi.org/10.1115/FEDSM2021-65637.

[6] Billuart P, Duponcheel M, Winckelmans G, Chatelain P. 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 2023;473. http://dx.doi.org/10.1016/j.jcp.2022.111726.

[7] Papadakis George, Voutsinas Spyros G. A strongly coupled Eulerian Lagrangian method verified in 2D external compressible flows. Comput & Fluids 2019;195:104325. http://dx.doi.org/10.1016/j.compfluid.2019.104325, URL https://www.sciencedirect.com/science/article/pii/S0045793019302853.

[8] Alletto Michael. Comparison of overset mesh with morphing mesh: Flow over a forced oscillating and freely oscillating 2D cylinder. OpenFOAM® J 2022;2(January):13–30. http://dx.doi.org/10.51560/ofj.v2.47.

[9] Pan Jingna, Ferreira Carlos, van Zuijlen Alexander. Estimation of power performances and flow characteristics for a savonius rotor by vortex particle method. Wind Energy 2023;26(1):76–97. http://dx.doi.org/10.1002/we.2788, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2788.

[10] Alvarez Eduardo J, Ning Andrew. High-fidelity modeling of multirotor aerodynamic interactions for aircraft design. AIAA J 2020;58(10):4385–400. http://dx.doi.org/10.2514/1.J059178, Publisher: American Institute of Aeronautics and Astronautics _eprint.

[11] Karimi-Zindashti Golnesa, Kurç Özgür. Flow past rotating cylinders using deterministic vortex method. Ocean Eng 2024;291:116342. http://dx.doi.org/10.1016/j.oceaneng.2023.116342, URL https://www.sciencedirect.com/science/article/pii/S0029801823027269.

[12] Papadakis George, Riziotis Vasilis A, Voutsinas Spyros G. A hybrid Lagrangian–Eulerian flow solver applied to elastically mounted cylinders in tandem arrangement. J Fluids Struct 2022;113:103686. http://dx.doi.org/10.1016/j.jfluidstructs.2022.103686, URL https://www.sciencedirect.com/science/article/pii/S0889974622001086.

[13] Pasolari Rention, Ferreira Carlos Simão, van Zuijlen Alexander, Baptista Carlos Fernando. Dynamic mesh simulations in OpenFOAM: A hybrid Eulerian–Lagrangian approach. Fluids 2024;9(2). http://dx.doi.org/10.3390/fluids9020051, URL https://www.mdpi.com/2311-5521/9/2/51.

[14] Papadakis Giorgos, Voutsinas Spyros G. In view of accelerating CFD simulations through coupling with vortex particle approximations. J Phys Conf Ser 2014;524(1). http://dx.doi.org/10.1088/1742-6596/524/1/012126.

[15] Stock Mark J, Gharakhani Adrin. Open-source accelerated vortex particle methods for unsteady flow simulation. Am Soc Mech Eng Fluids Eng Div (Publication) FEDSM 2020;3. http://dx.doi.org/10.1115/FEDSM2020-20486.

[16] Cottet Georges-Henri. A particle-grid superposition method for the Navier-Stokes equations. J Comput Phys 1990;89(2):301–18. http://dx.doi.org/10.1016/0021-9991(90)90146-R, URL https://www.sciencedirect.com/science/article/pii/002199919090146R.

[17] Cottet Georges-Henri, Koumoutsakos Petros. Vortex methods - Theory and practice. Cambridge University Press; 2000.

[18] Mimeau Chloé, Mortazavi Iraj. A review of vortex methods and their applications: From creation to recent advances. Fluids 2021;6. http://dx.doi.org/10.3390/fluids6020068.

[19] Chorin Alexandre Joel. Numerical study of slightly viscous flow. J Fluid Mech-ICS 1973;57:785–96, Part 4.

[20] Tutty OR. A Simple Redistribution Vortex Method (with Accurate Body Forces). 2010, p. 1–27, arXiv:1009.0166.

[21] OpenFOAM Foundation. OpenFOAM version 9. 2020, https://openfoam.org/version/9/. [Accessed 22 February 2024].

[22] OpenFOAM Foundation. pimpleLoop class reference. 2024, https://cpp.openfoam.org/v9/classFoam_1_1pimpleLoop.html. [Accessed 22 February 2024].

[23] Pingjian Ming, Wenping Zhang. Numerical simulation of low Reynolds number fluid-structure interaction with immersed boundary method. Chin J Aeronaut 2009;22(5):480–5. http://dx.doi.org/10.1016/S1000-9361(08)60129-6, URL https://www.sciencedirect.com/science/article/pii/S1000936108601296.

[24] Mittal Sanjay, Kumar Bhaskar. Flow past a rotating cylinder. J Fluid Mech 2003;476:303–34. http://dx.doi.org/10.1017/S0022112002002938.

[25] Delft High Performance Computing Centre (DHPC). DelftBlue supercomputer (phase 1). 2022, https://www.tudelft.nl/dhpc/ark:/44463/DelftPhase1.