Engineering with Computers
https://doi.org/10.1007/s00366-024-02043-6

ORIGINAL ARTICLE

3d fluid–structure interaction simulation with an Arbitrary–Lagrangian–Eulerian approach with applications to flying objects

Daniele Di Cristofaro[1] · **Attilio Frangi**[1] · **Massimiliano Cremonesi**[1]

Received: 4 March 2024 / Accepted: 25 July 2024
© The Author(s) 2024

Abstract
Air-structure interaction is a key aspect to account for during the design of Micro Air Vehicles. In this context, modelisation and numerical simulations represent a powerful tool to analyse aerodynamic performances. This work proposes an advanced fluid–structure interaction numerical technique for the simulation of dragonfly wings, considered one of the most interesting model due to their complex flapping kinematic. The fluid subproblem, described by incompressible Navier–Stokes equations, is solved in a Finite Element Arbitrary-Lagrangian-Eulerian framework, while the solid subproblem is addressed using structural Finite Element, such as membranes and beams. Moreover, a novel remeshing algorithm based on connectivity manipulation and refinement procedure has been implemented to reduce element distortion in fluid mesh, thus increasing the accuracy of the fluid solution. Firstly, the deformation of a single hindwing has been studied. Secondly, the dragonfly model is enriched by incorporating the forewing and a simplified thorax geometry. Preliminary results highlight the complex dynamic of the fluid around the body as well as the efficiency of the proposed mesh generation algorithm.

Keywords Fluid–Structure Interaction · Arbitrary-Lagrangian–Eulerian · Structural Finite Elements · Remeshing Technique · Flying Objects

1 Introduction

Flying objects have always fascinated and intrigued humans, symbolising both impressive technological achievements and the marvels of nature. From bird flight to the advanced engineering of modern aircraft, the field of aerial entities encompasses a wide variety of shapes, sizes, and functions.

Within this context, Micro Air Vehicles (MAVs) represent the state-of-the-art in aerospace engineering technology and robotics [1, 2]. These compact aerial platforms have progressively gained significant interest thanks to their small dimension, lightweight design, and manoeuvrability [3–5]. Notably, recent MAVs designs draw inspiration from various insects including Diptera [6], Hymenoptera [7], and Odonata [8], as well as avian species such as doves [9, 10]. These cutting-edge bio-inspired MAVs aim to mimic the high aerodynamic performance of their wings and achieve their optimal weight balance. The complete comprehension of the structural functioning of wings under dynamic wing loading poses an ongoing challenge to optimise the design of MAVs [11].

In the present investigation, we focus on the three-dimensional structural function of dragonfly wings, because their venation pattern and corrugation have remained surprisingly similar throughout their evolution, indicating that this conserved trait plays a significant role in flight [12]. Additionally, dragonflies possess two pairs of wings that can be independently controlled, which enable them to perform

Communicated by Kris van der Zee.

Attilio Frangi and Massimiliano Cremonesi have contributed equally to this work.

✉ Daniele Di Cristofaro
daniele.dicristofaro@polimi.it

Attilio Frangi
attilio.frangi@polimi.it

Massimiliano Cremonesi
massimiliano.cremonesi@polimi.it

[1] Department of Civil and Environmental Engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy

Published online: 11 September 2024

impressive manoeuvres and maintain greater stability throughout all stages of gliding [13].

Furthermore, previous works have discovered intriguing findings regarding the crucial role of insect venation patterns in their capacity to bear loads [14]. Remarkably, the corrugated nature of insect wings has proven to be advantageous in terms of both stiffness and aerodynamic prowess at low Reynolds numbers providing strong structural integrity [15]. In this regard, researchers worldwide attempt to reproduce insect flight with mechanical systems, and several prototypes capable of hovering, gliding, and control authority have been developed [16, 17]. In particular, the choice of the wings geometry requires an in-depth study of efficient and aerodynamically performing design [18].

During this design phase, analyses of the structure and reliability of the devices are necessary [19]. In this context, numerical simulation allows for quantitatively studying both the structural response and aerodynamic performance beforehand, thereby reducing the number of prototypes to be manufactured [20]. From a mathematical perspective, the physical phenomenon can be modeled as a Fluid-Structure Interaction (FSI) problem, since the deformation of the structure is strongly influenced by the force distribution over the entire wing surface [21].

In the current literature, there exist two categories of numerical techniques to treat FSI problems: monolithic approaches and partitioned ones. The monolithic approach treats the entire problem as one unit and solves the fluid and the solid equations together. This method offers the advantage of keeping accuracy and stability through strong coupling between the equations [22]. Nevertheless, it requires the development of specialised software and does not permit the reuse of existing modules for the fluid and structural domains. Additionally, it is worth noting that solving a large system of equations from two distinct physical domains can present numerical challenges, thereby limiting its practicality in real-world engineering problems [23]. In the latter approach, the fluid and structural equations are solved separately but connected through interface conditions. This can be achieved using a strong coupling, which involves subiterations [24], or weak coupling, which imposes weakly kinematic and dynamic constraints [25, 26]. This approach offers the advantage of reusing existing codes for different components of the problem, making it more flexible and solving smaller equation systems. However, a disadvantage of loosely coupled partitioned schemes is that interface conditions are imposed asynchronously, which could potentially lead to decreased accuracy and numerical instabilities [27]. In this setting, we have chosen to adopt a partitioned scheme and to apply strong coupling conditions at the interface, additionally prescribing mesh conformity among the fluid and solid meshes. Finally, the algorithm's modularity enables us to utilise the commercial software Abaqus [28] for the solid and coupling subproblems and focus our effort on developing advanced features for the fluid subproblem.

To simulate MAV aerodynamics in a FSI framework, several numerical strategies have been employed. In [29], a comprehensive analysis of wing flexibility has been performed with the finite volume method, emphasising their mechanical behavior and stability in forward flight. More recently, the Immersed Boundary Method (IBM) [30] has been adopted to optimise MAVs design and improve their reliability under harsh atmospheric conditions [31–33]. Finally, Finite Element Methods (FEM) are adopted during the design process, aiming to enhance comprehension of flapping wing's aerodynamics and the formulation of aerodynamic models encompassing aspects such as flight stability, drag and lift forces, and eddies generation [34, 35].

To guarantee the accuracy of predicting aerodynamic interaction of the wings of dragonflies during large amplitude flapping, we utilise an Arbitrary-Lagrangian–Eulerian (ALE) solver for incompressible Navier–Stokes equations [36]. However, a limitation of this approach is the occurrence of mesh distortion, which appears when the fluid domain is significantly deformed by the solid solution [37]. In this context, numerous remeshing and remapping procedures are employed to ensure that the computational grid remains accurate and well-suited to represent the evolving physical phenomena throughout the simulation [38].

Adaptive Mesh Refinement (AMR) is a widely used technique in computational fluid dynamics that dynamically adjusts mesh resolution based on local flow features or solution gradients. This approach enables simulations to focus computational resources where they are most needed, achieving higher accuracy with fewer computational elements in regions of low interest. AMR algorithms typically employ criteria such as solution gradients, error estimators, or flow features to guide mesh refinement. An example of AMR implementation is described in [39], where they introduced a hierarchical approach to refine meshes in regions of interest. On the other hand, coarsening involves reducing mesh resolution in regions where detailed solutions are not necessary, thus limiting computational cost without sacrificing accuracy [40]. Techniques, such as grid coarsening or element collapse, are commonly used in simulations where overall trends are more critical than local details. Local Mesh Modification refers to targeted adjustments made to specific regions of the mesh to accommodate geometric changes or refine resolution in critical areas [41]. This approach is particularly useful in simulations involving complex geometries or adaptive boundary conditions.

With reference to the in-house fluid solver in [42], we introduce a novel global remeshing algorithm for the quadratic velocity solution, based on mesh connectivity manipulation such as element collapse and expansion [43], edge-swap [44], and vertices re-connection [45] to drastically

reduce projection error. Moreover, this procedure is enriched with a local refinement and derefinement option, to avoid loss of accuracy in the FSI solution [46, 47].

The paper is organised as follows. We introduce the fluid in-house solver along with the innovative re-connection algorithm in Sect. 2. Sections 3 and 4 provide detailed explanations of the solid solver and the adopted coupling scheme, respectively, with a particular focus on the splitting surface procedure. Subsequently, in Sect. 5 we conduct three numerical benchmarks to validate all the newly implemented features. Finally, we present the dragonfly wings modelisation and discuss numerical results in Sect. 6.

2 Fluid solver

2.1 Continuum equation

Let us define a final time $T > 0$ and a generic bounded, Lipschitz, and time-dependent fluid domain $\Omega_f^t = \Omega_f(t) \subset \mathbb{R}^3$, for every $t \in [0, T]$. Throughout the following discussion, we denote as $\partial\Omega_f^t = \partial\Omega_f(t)$ the fluid domain boundaries, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ as the fluid velocity field, and $p = p(\mathbf{x}, t)$ as the fluid pressure field, where $\mathbf{x} \in \Omega_f^t$ indicates the position vector in the current configuration. The boundaries can be divided into three pairwise disjoint subsets, as follows:

$$\partial\Omega_f^t = \Gamma_{f,d}^t \cup \Gamma_{f,n}^t \cup \Gamma_{fsi}^t, \quad t \in [0, T], \tag{1}$$

where $\Gamma_{f,d}^t = \Gamma_{f,d}(t)$ is Dirichlet boundary, $\Gamma_{f,n}^t = \Gamma_{f,n}(t)$ is Neumann boundary, and $\Gamma_{fsi}^t = \Gamma_{fsi}(t)$ indicates fluid–structure interface.

In the present work, the fluid domain is modelled as an incompressible Newtonian medium using an ALE approach [48]. The continuum equation, for every $\mathbf{x} \in \Omega_f^t$, $t \in (0, T]$ reads:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} = \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\sigma}_f(p, \mathbf{u}) + \mathbf{f}_e, \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

where $\boldsymbol{\sigma}_f(p, \mathbf{u}) = -p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})$ is the Cauchy stress tensor, $\mathbf{I}$ is the identity tensor, ∇ is the spatial gradient operator with respect to current coordinates, ρ_f is the fluid density, $\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$ is the domain velocity field, $\mathbf{f}_e = \mathbf{f}_e(\mathbf{x}, t)$ is a generic volume force per unit mass, $\nu_f = \mu/\rho_f$ is the kinematic viscosity, and μ is the dynamic viscosity. Moreover, the physical properties ρ_f and μ are assumed to be uniform across the domain and independent of time. Equation (2) enforces momentum conservation, while Eq. (3) imposes incompressibility constraint.

In addition, these equations are endowed with the following mixed boundary conditions, for every $t \in (0, T]$:

$$\mathbf{u} = \mathbf{v}, \quad \mathbf{x} \in \Gamma_{fsi}^t, \tag{4}$$

$$\mathbf{u} = \mathbf{g}, \quad \mathbf{x} \in \Gamma_{f,d}^t, \tag{5}$$

$$\boldsymbol{\sigma}_f(p, \mathbf{u}) \cdot \mathbf{n}_f = \mathbf{h}, \quad \mathbf{x} \in \Gamma_{f,n}^t, \tag{6}$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is the solid velocity at the fluid–structure surface, $\mathbf{n}_f = \mathbf{n}_f(\mathbf{x}, t)$ is the outward unit normal vector associated to $\partial\Omega_f^t$, $\mathbf{g} = \mathbf{g}(\mathbf{x}, t)$ is a prescribed velocity, and $\mathbf{h} = \mathbf{h}(\mathbf{x}, t)$ is a prescribed stress distribution.

Finally, to ensure the model's well-posedness, we provide an initial condition for the velocity field, as follows:

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \qquad \mathbf{x} \in \Omega_f^0, \tag{7}$$

where Ω_f^0 is the initial domain configuration and $\mathbf{u}_0 = \mathbf{u}_0(\mathbf{x})$ is the initial velocity profile.

2.2 Space discretisation

The fluid domain is discretised, at each time instance $t \in [0, T]$, with a mesh $\mathcal{T}_H^t = \mathcal{T}_H(t)$ composed of linear tetrahedral elements $K_j^t = K_j(t)$, $j = 1, ..., E^t$, where $E^t = E(t)$ is the element number as a function of time and $N^t = N(t)$ is the node number as a function of time.

In this context, we choose Ladyzhenskaya-Babuška-Brezzi (LBB) stable $\mathbb{P}^2 - \mathbb{P}^1$ spaces [49]. These spaces ensure stability in the numerical representation of the discretised fluid velocity $\mathbf{u}_H = \mathbf{u}_H(t)$ which exhibits piecewise quadratic behavior, and the discretised pressure field $p_H = p_H(t)$, which displays piecewise linear behavior on the mesh. In addition, the mesh velocity field $\mathbf{w}_H = \mathbf{w}_H(t)$ is defined to be linear across the mesh, thereby imposing a constraint of straight edges on the tetrahedral elements.

Finally, the formulation is enhanced by incorporating a Streamline Upwind Petrov-Galerkin (SUPG) stabilisation to effectively manage high Reynolds numbers [50], where the optimal stabilisation parameter is defined elementwise following [51].

2.3 Time discretisation

Let us consider the subsequent subdivision of the time interval $[0, T]$:

$$0 = t^0 < t^1 < ... < t^n < ... < t^M = T, \tag{8}$$

where $t^{n+1} - t^n = \Delta t^n$, $n = 0, 1, ..., M - 1$. Therefore, we can define the fluid velocity field $\mathbf{u}_H^n = \mathbf{u}_H(t^n)$, the mesh velocity

field $\mathbf{w}_H^n = \mathbf{w}_H(t^n)$, the pressure field $p_H^n = p_H(t^n)$, and the fluid mesh $\mathcal{T}_H^n = \mathcal{T}_H(t^n)$, at a fixed time t^n.

In this work, we rely on a Chorin-Temam incremental fractional step method [52, 53], wherein we incorporate a semi-implicit approach to address the advection term. We introduce an intermediate velocity field $\widetilde{\mathbf{u}}$ and the pressure variation $\delta p = p_H^{n+1} - p_H^n$. The fractional step method consists of solving sequentially the following equations:

$$\mathbf{K}_u^{n+1}(\boldsymbol{\phi}_H^n)\widetilde{\mathbf{U}} = \mathbf{f}_1^{n+1}(p_H^n) + \mathbf{f}_2^n(\mathbf{u}_H^n), \tag{9}$$

$$\mathbf{K}_p^{n+1}\delta\mathbf{P} = \mathbf{f}_3^{n+1}(\widetilde{\mathbf{u}}), \tag{10}$$

$$\mathbf{M}_u^{n+1}\mathbf{U}^{n+1} = \mathbf{M}_u^{n+1}\widetilde{\mathbf{U}} - \Delta t^n \mathbf{G}^{n+1}\delta\mathbf{P}, \tag{11}$$

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \delta\mathbf{P}, \tag{12}$$

where $\mathbf{K}_u^{n+1}$ is the velocity stiffness matrix, $\boldsymbol{\phi}_H^n = \mathbf{u}_H^n - \mathbf{w}_H^{n+1}$ is the advection velocity, $\widetilde{\mathbf{U}}$ is the intermediate velocity vector at time t^{n+1}, $\mathbf{f}_1^{n+1}$ is the velocity right-hand side computed on $\mathcal{T}_H^{n+1}$, $\mathbf{f}_2^n$ is the velocity right-hand side computed on $\mathcal{T}_H^n$, $\mathbf{K}_p^{n+1}$ is the pressure stiffness matrix, $\mathbf{f}_3^{n+1}$ is the pressure right-hand side computed on $\mathcal{T}_H^{n+1}$, $\mathbf{M}_u^{n+1}$ is the velocity mass matrix, $\mathbf{G}^{n+1}$ is the gradient matrix, $\delta\mathbf{P}$ is the pressure variation vector, $\mathbf{U}^n$ indicates the velocity vector at time t^n, and $\mathbf{P}^n$ is the pressure vector at time t^n. In particular, the right-hand sides have the following expressions [54]:

$$\mathbf{f}_1^{n+1} = -\frac{1}{\rho_f}\mathop{\mathbf{A}}_{j=1}^{E^{n+1}}\int_{K_j^{n+1}} \nabla p_H^n \cdot \mathbf{v}\,\mathrm{d}\Omega, \tag{13}$$

$$\mathbf{f}_2^n = \frac{1}{\Delta t^n}\mathop{\mathbf{A}}_{j=1}^{E^n}\int_{K_j^n} \mathbf{u}_H^n \cdot \mathbf{v}\,\mathrm{d}\Omega, \tag{14}$$

$$\mathbf{f}_3^{n+1} = -\frac{\rho_f}{\Delta t^n}\mathop{\mathbf{A}}_{j=1}^{E^{n+1}}\int_{K_j^{n+1}} (\nabla \cdot \widetilde{\mathbf{u}})s\,\mathrm{d}\Omega, \tag{15}$$

where $(\mathbf{v}, s)$ are the local test functions for velocity and pressure field, $E^n = E(t^n)$ is the total number of elements $K_j^n = K_j(t^n)$, $j = 1, ..., E^n$ at time t^n, and $\mathbf{A}$ indicates the assembly operator. The treatment of boundary conditions in the context of a fractional step scheme is crucial. In this work, we rely on the classical split boundary condition presented e.g. in [55].

To handle the non-linearity of the advection term and to reduce the error generated by fractional scheme, we enrich the fluid solver by incorporating a subiteration procedure [56]. This technique involves repeating the predictor-corrector scheme several times, where the advection velocity is updated with the previous fluid velocity solution, as follows:

$$\boldsymbol{\phi}_H^{n,b+1} = \mathbf{u}_H^{n,b} - \mathbf{w}_H^{n+1}, \quad b \geq 0, \tag{16}$$

In particular, the fluid solver iterates until either the number of iterations reaches a maximum value b_m or until convergence is achieved by checking the subsequent relative criterion, for $b \geq 0$:

$$||\mathbf{u}_H^{n,b+1} - \mathbf{u}_H^{n,b}||_{H^1} < \varepsilon_u ||\mathbf{u}_H^{n,b}||_{H^1}, \tag{17}$$

$$||p_H^{n,b+1} - p_H^{n,b}||_{L^2} < \varepsilon_p ||p_H^{n,b}||_{L^2}, \tag{18}$$

where ε_u and ε_p are user-defined tolerances. Lastly, we apply a constant time interpolation between two consecutive time steps, namely:

$$\mathbf{u}_H^{n+1,0} = \mathbf{u}_H^{n,B}, \quad p_H^{n+1,0} = p_H^{n,B}, \tag{19}$$

where $(\mathbf{u}_H^{n,B}, p_H^{n,B})$, $B \in \{1, ..., b_m\}$ denotes the converged solution at the previous time step.

2.4 Domain displacement

To describe the motion of fluids we have chosen an ALE approach, where the grid follows a motion rule that is neither fixed nor the fluid one. We define the motion $\mathcal{A}$ of the frame as follows:

$$\Omega_f^0 \times (0, T] \ \ni \ (\mathbf{r}, t) \rightarrow \mathbf{x} = \mathcal{A}(\mathbf{r}, t) \in \Omega_f^t, \tag{20}$$

where $\mathbf{r}$ are the coordinates in the reference configuration. Moreover, we assume that $\mathcal{A}$ is invertible $\forall t \in (0, T]$, and we indicate its inverse as $\mathcal{A}^{-1}$. This theoretical framework allows us to accurately define $\Omega_f^t = \mathcal{A}(\Omega_f^0, t)$ and $\partial\Omega_f^t = \mathcal{A}(\partial\Omega_f^0, t)$, for every time $t \in (0, T]$.

To govern grid displacement, the Navier–Stokes equations have to be complemented by an additional equation for the mesh velocity $\mathbf{w}$, endowed with the following Dirichlet boundary conditions:

$$\mathbf{w} = \mathbf{v}, \qquad \mathbf{x} \in \Gamma_{fsi}^t, \tag{21}$$

$$\mathbf{w} = \mathbf{0}, \qquad \mathbf{x} \in \partial\Omega_f^t \setminus \Gamma_{fsi}^t. \tag{22}$$

In this work, we propose a method consisting of solving, for every time step t^n, a linear elliptic problem with the anisotropic behavior of the discretised mesh velocity field $\mathbf{w}_H^n$. Anisotropy is prescribed as a function of the domain tessellation and introduced at the element's level using fictitious springs connecting nodes and faces [57]. More precisely, for

every node $\mathbf{x}_i$ of the mesh, two sets of equations are imposed. The first one states that elastic forces generated by the adjacent edges and with a magnitude proportional to the inverse of their lengths must balance, namely:

$$\sum_{j=1}^{G_i} \mathbf{f}_{ij} = 0, \quad \mathbf{f}_{ij} = k_{ij}\mathbf{e}_{ij}, \quad \forall i = 1, ..., N^n, \tag{23}$$

where G_i is the number of neighborhood of node i, $N^n = N(t^n)$ is the number of node in the mesh at time t^n, $\mathbf{f}_{ij}$ is the force that node i exerts on the node j, $k_{ij} = 1/|\mathbf{e}_{ij}|$ is the anisotropic elastic coefficient, and $\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the vector connecting the two nodes.

Regarding the second set of equations, for every node the following relation is imposed:

$$\sum_{p=1}^{F_i} \mathbf{t}_{ip} = 0, \quad \mathbf{t}_{ip} = k_{ip}\mathbf{e}_{ip}, \quad \forall i = 1, ..., N^n, \tag{24}$$

where $\mathbf{t}_{ip}$ is the force that node $\mathbf{x}_i$ exerts on the virtual node $\mathbf{x}_p$ that is the projection of $\mathbf{x}_i$ on the opposite face of the tetrahedron, F_i denotes the number of faces in the range of the polyhedral ball that encloses the i-th node, and B_i is the set of faces that encircle $\mathbf{x}_i$ and are one edge away from it. The resultant is a linear system, for every time t^n, of the form:

$$\mathbf{K}_w^{n+1}\delta\mathbf{X}^{n+1} = \mathbf{f}_w^{n+1}, \tag{25}$$

where $\mathbf{K}_w^{n+1}$ is the stiffness matrix obtained from (23), (24), $\delta\mathbf{X}^{n+1}$ is the vector of mesh displacement, and $\mathbf{f}_w^{n+1}$ is the right and side that takes into account the boundary conditions in Eq. (21), (22). Once the mesh displacement $\delta\mathbf{x}^{n+1}$ is known, the nodes' positions and mesh velocity $\mathbf{w}_H^{n+1}$ are updated in an explicit fashion, as follows:

$$\mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \delta\mathbf{x}^{n+1}, \quad \mathbf{w}_H^{n+1} = \frac{\delta\mathbf{x}^{n+1}}{\Delta t^n}, \tag{26}$$

for every nodes $\mathbf{x}_j$, $j = 1, ..., N^n$.

It is worth to notice that the proposed technique effectively disentangles the mesh displacement from the fluid velocity [58], thus allowing us to modify the mesh without directly impacting the fluid solution.

2.5 Re-connection algorithm

During FEM simulation, it could happen that the mesh $\mathcal{T}_H^n$ is not suitable to describe Ω_f^t because of flipped elements that make the linear system ill-posed or bad-shaped elements that reduce the accuracy of the numerical solution. To address these issues, a new mesh $\mathcal{T}_{H+1}^n$ is generated through a re-connection procedure.

To properly investigate any mesh manipulation, a quality mesh indicator to estimate quantitative both global and local mesh distortion has been defined in Sect. 2.5.1. Moreover, the employed mesh generator is introduced in Sect. 2.5.2, while a suitable remapping scheme for the physical variable has been described in Sect. 2.5.3. Finally, an alternative approach involving refinement and derefinement is presented in Sect. 2.5.4, with the purpose of improving the mesh quality just after the connectivity change.

2.5.1 Quality mesh indicator

To identify the quality of tetrahedra, several mesh indicators have been proposed, such as eccentricity, aspect ratio, outer radius, and in-radius [59, 60]. In this work, we choose as local quality mesh indicator the function Q, defined as:

$$Q(K^n) = \xi \frac{V(K^n)}{(\sum_{i=1}^{6} |\boldsymbol{a}_i|^2)^{3/2}}, \quad \forall K^n \in \mathcal{T}_H^n, \tag{27}$$

where $\boldsymbol{a}_i$, $i = 1, .., 6$ are the tetrahedron edges, V is the tetrahedron volume, and $\xi = 72\sqrt{3}$ has been chosen in such a way that $Q(K^n) \in [0, 1]$. The maximum value $Q(K^n) = 1$ is achieved only if K^n is a regular tetrahedron, meaning that its edges have the same length [61]. This measure demonstrates a high sensitivity in determining whether an element can be deemed well-shaped and, furthermore, is directly proportional to the elemental jacobian, thus facilitating the identification of inverted elements, characterised by a non-admissible negative Q value. Accordingly, we can define a global mesh indicator as follows:

$$Q_{\text{MIN}}(\mathcal{T}_H^n) = \min_n Q(K^n). \tag{28}$$

In this setting, we can state the following global re-connection criterion: if $Q_{\text{MIN}}(\mathcal{T}_H^{n+1}) \leq Q^*$, then re-connection is performed, namely:

$$\mathcal{T}_H^n \to \mathcal{T}_{H+1}^n, \tag{29}$$

where $Q^* \in [0, 1]$ is a user-defined threshold. As a consequence, $H \in \mathbb{N}$ serves as a counter of the total number of re-connections that have taken place.

2.5.2 Mesh generation

Let Π_H^1 be the set of nodes located at the tetrahedron's vertices, Π_H^2 the set of edge nodes, and $\Pi_H = \Pi_H^1 \cup \Pi_H^2$ the entire nodes associated to $\mathcal{T}_H^n$, with H fixed. The new mesh is generated through a Constrained Delaunay Tetrahedrisation [62] (CDT), namely:

$$(\Pi_H^1, \mathcal{C}_H^n) \to (\Pi_{H+1}^1, \mathcal{T}_{H+1}^n), \tag{30}$$

where $\mathcal{C}_H^n$ is the mesh associated with fluid boundaries $\partial\Omega_f^n$ at a fixed time t^n. The result is a new linear tessellation in

which $\Pi^1_{H+1} \subseteq \Pi^1_H$, and $\mathcal{C}^n_{H+1} = \mathcal{C}^n_H$. Furthermore, Π^2_{H+1} is generated as middle points of the new mesh's edges.

It is worth noticing that the insertion of new vertices is strictly forbidden, but the deletion of existing ones is allowed whenever is required by the mesh generator, thus reducing the operations in the remapping phase.

2.5.3 Remapping phase

Once a change of mesh is performed, the previous solution has to be properly mapped on the new tetrahedralisation. More precisely, given pressure field p^n_H and velocity field $\mathbf{u}^n_H$ on $\mathcal{T}^n_H$, we need to compute p^n_{H+1} and $\mathbf{u}^n_{H+1}$ on $\mathcal{T}^n_{H+1}$ such that linear momentum and mass are conserved. In this context, several non-conservative choices have been proposed [63, 64], as well as a repair algorithm [65]. Considering that the Delaunay tesselation used to generate the new mesh keeps the linear nodes in the same positions ($\Pi^1_{H+1} \subseteq \Pi^1_H$), the pressure field does not need a remapping, namely:

$$p^n_{H+1}(\mathbf{x}_k) = p^n_H(\mathbf{x}_k), \qquad \forall \mathbf{x}_k \in \Pi^1_{H+1}. \tag{31}$$

The same consideration can be repeated for the nodal fluid velocity values at the vertexes of the tetrahedra. The remapping of nodal velocities is only required in Π^2_{H+1} (nodes in the middle of each edge). We decide to apply a remapping method based on Lagrangian interpolation, namely, for every $\mathbf{x}_k \in \Pi^2_{H+1}$:

$$\mathbf{u}^n_{H+1}(\mathbf{x}_k) = \mathbf{u}^n_H(\mathbf{x}_k), \tag{32}$$

if $\mathbf{x}_k \in \Pi^2_H$, otherwise, there exists a unique element $K \in \mathcal{T}^n_H$ to which $\mathbf{x}_k$ belongs. In this second case, the quadratic fluid velocity function $\bar{\mathbf{u}}$ is obtained by interpolating on K, for $i = 1, ..., 3$:

$$\bar{u}_i(\mathbf{x}) = \sum_{j=1}^{d} N_{ij}(\mathbf{x}) \hat{U}^n_{ij}, \quad \mathbf{x} \in K, \tag{33}$$

where $d = 10$, N_{ij} is the matrix of the local shape functions, and $\hat{U}^n_{ij}$ are the old values of the velocity vector restricted on the element K, at time t^n. The interpolated new value is computed straightforward as:

$$\mathbf{u}^n_{H+1}(\mathbf{x}_k) = \bar{\mathbf{u}}(\mathbf{x}_k). \tag{34}$$

The approach described above has the drawback of being non-conservative and very diffusive. In order to overcome this problem, we perform a Pressure-Implicit with Splitting of Operators (PISO) subiteration [66], employing an iterative predictor-corrector scheme with Eqs. (10), (11). The convergence is reached according to:

$$||p^{n,f+1}_{H+1} - p^{n,f}_{H+1}||_{L^2} \leq \varepsilon_R, \quad 0 \leq f \leq f_m, \tag{35}$$

where ε_R is a user-defined absolute tolerance, f_m is the user-defined maximum number of PISO subiterations, and $p^{n,0}_{H+1}$ is obtained using Eq. (31). In this way, we respect the numerical incompressibility constraint and the linear momentum conservation is partially reconstructed.

2.5.4 Refinement option

One of the possible outcomes of the re-connection algorithm is that $\mathcal{T}^n_H = \mathcal{T}^n_{H+1}$, meaning that mesh distortion is not solved. This could happen, for instance, when a large deformation or large translation occurs.

In this scenario, an alternative approach is proposed, based on a local refinement and derefinement, which modify the generator set Π^1_H, namely:

$$\Pi^1_{H+1} = (\Pi^1_H \cup \Pi_a) \setminus \Pi_r, \tag{36}$$

where Π_a is the set of new generator nodes, and Π_r contains generators to remove from the original set.

Let us define l_j the length of the j-th edge in the mesh, parameterised as a convex combination of its endpoints $\mathbf{x}_{j_1}$, $\mathbf{x}_{j_2}$, as follows:

$$k\mathbf{x}_{j_2} + (1-k)\mathbf{x}_{j_1}, \quad k \in [0, 1]. \tag{37}$$

Moreover, we introduce two user-defined parameters $\alpha_1 < 1$, $\alpha_2 > -1$ and we denote the maximum element diameter as L. In this framework, Π_a is filled employing the subsequent criterion:

$$\frac{1}{2}(\mathbf{x}_{j_1} + \mathbf{x}_{j_2}) = \mathbf{x}_{j_m} \in \Pi_a, \tag{38}$$

if the length of the j-th edge exceeds a fixed threshold $l_1 = (1 + \alpha_2)L$ and one and only one endpoint belongs to $\mathcal{C}^n_H$, otherwise the entire edge lies on the fluid boundary, hence requiring a modification of the constrained surface, which is not allowed by the solver. Additionally, this set is further filtered in such a way that the distance between every element of Π_a is not below a minimum length $l_2 = (1 - \alpha_1)L$, as illustrated in Algorithm 1.

It is important to recall that every newly added node is in the middle of an edge and belongs to Π^2_H, consequently, fluid velocity at the previous time step is known and a mapping procedure is not necessary.

Regarding Π_r, the procedure is similar. For every edge which length is lower than l_2, three mutually disjoint scenarios could happen: if none of the two endpoints belongs to $\mathcal{C}^n_H$, we can freely choose which one to remove, if one endpoint belongs to $\mathcal{C}^n_H$, we can remove the other, while, if both endpoints belong to $\mathcal{C}^n_H$, the considered edge lies on the fluid boundary and no refinement procedure is applied. This algorithm allows us to maintain good global mesh regularity without making the mapping phase more computationally

demanding, as the solution of pressure and velocity is known at the middle point of every edge. The entire algorithm is summarised in Algorithm 2.

Algorithm 1 Filtering procedure

Require: Π_a, l_2
for $\mathbf{x} \in \Pi_a$ **do**
  for $\mathbf{y} \in \Pi_a \backslash \{\mathbf{x}\}$ **do**
    if $|\mathbf{x} - \mathbf{y}| < l_2$ **then**
      Remove $\mathbf{y}$ from Π_a
    end if
  end for
end for

Algorithm 2 Re-connection algorithm

Require: $Q^* \in [0,1]$, refine $\in \{0,1\}$, M, $n = 1$
Impose initial conditions
while $n \leq M$ **do**
  Solve FSI problem at time t^n
  Mesh displacement $\mathcal{T}_H^n \rightarrow \mathcal{T}_H^{n+1}$
  Compute Q_{MIN}
  if $Q_{MIN} < Q^*$ **then**
    $\mathcal{T}_H^{n+1} \rightarrow \mathcal{T}_H^n$
    if refine **then**
      $\mathcal{T}_H^n \rightarrow \mathcal{T}_{H+1}^n$ (2.5.4)
    else
      $\mathcal{T}_H^n \rightarrow \mathcal{T}_{H+1}^n$ (2.5.2)
    end if
    Remapping phase (2.5.3)
  end if
  $\mathcal{T}_{H+1}^n \rightarrow \mathcal{T}_{H+1}^{n+1}$
  $t = t + \Delta t^n$
  $n = n + 1$
end while

3 Solid solver

As previously mentioned, the solid subdomain will be investigated using the commercial software Abaqus/Standard. The notable advantages lie in its highly efficient and comprehensive library of structural elements, including shells, membranes, and beams, as well as its extensive collection of pre-built constitutive laws [28]. As a consequence, we will solely present the general formulation of the initial boundary value problem for dynamic structural analysis, in order to introduce the variables involved in the FSI problem. We will henceforth denote the solid velocity, solid displacement, and solid acceleration as $\mathbf{d} = \mathbf{d}(\mathbf{x}, t)$, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$, and $\mathbf{a} = \mathbf{a}(\mathbf{x}, t)$, respectively, and the solid domain as $\Omega_s^t = \Omega_s(t) \subset \mathbb{R}^3$, for every $t \in [0, T]$.

3.1 Three dimensional solid domain

The strong formulation of the equilibrium conditions for the structural domain leads to momentum conservation, for every $\mathbf{x} \in \Omega_s^t, t \in (0, T]$:

$$\rho_s \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \nabla \cdot \boldsymbol{\sigma}_s + \rho_s \mathbf{b}, \tag{39}$$

where ρ_s is the solid density, $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_s(\mathbf{x}, t)$ is the Cauchy stress tensor, and $\mathbf{b} = \mathbf{b}(\mathbf{x}, t)$ is a generic force for unit volume. Equation (39) is complemented with proper boundary conditions and initial conditions, namely:

$$\mathbf{d} = \mathbf{j}, \quad \mathbf{x} \in \Gamma_{s,d}^t, t \in (0, T], \tag{40}$$

$$\boldsymbol{\sigma}_s \cdot \mathbf{n}_s = \mathbf{q}, \quad \mathbf{x} \in \Gamma_{s,n}^t, t \in (0, T], \tag{41}$$

$$\boldsymbol{\sigma}_s \cdot \mathbf{n}_s = \mathbf{f}_{fsi}, \quad \mathbf{x} \in \Gamma_{fsi}^t, t \in (0, T], \tag{42}$$

$$\mathbf{v} = \mathbf{v}_0, \quad \mathbf{x} \in \Omega_s^0, t = 0, \tag{43}$$

$$\mathbf{d} = \mathbf{d}_0, \quad \mathbf{x} \in \Omega_s^0, t = 0, \tag{44}$$

where $\mathbf{j} = \mathbf{j}(\mathbf{x}, t)$ is a prescribed displacement, $\mathbf{n}_s = \mathbf{n}_s(\mathbf{x}, t)$ is the solid outer normal vector, $\mathbf{q} = \mathbf{q}(\mathbf{x}, t)$ is a prescribed force, $\mathbf{v}_0$ is a given initial velocity field, $\mathbf{d}_0$ is a given initial displacement field, and $\Gamma_{s,d}^t = \Gamma_{s,d}(t)$, $\Gamma_{s,n}^t = \Gamma_{s,n}(t)$ represent the Dirichlet and Neumann boundaries, respectively.

Following a standard procedure, Eq. (39) is formulated in a weak form and both space and time discretisations are performed. Regarding time integration, for addressing transient dynamics the second order α-Hilbert-Huges-Taylor method [67] is employed, namely:

$$\mathbf{a}^{n+1} = \mathbf{M}^{-1}[\mathbf{c}(\mathbf{d}^n, \mathbf{d}^{n+1}) + \mathbf{F}^{n+1}], \tag{45}$$

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^n \mathbf{v}^n + (\Delta t^n)^2 s_1(\mathbf{a}^n, \mathbf{a}^{n+1}), \tag{46}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t^n s_2(\mathbf{a}^n, \mathbf{a}^{n+1}), \tag{47}$$

where we define:

$$\mathbf{c}(\mathbf{d}^n, \mathbf{d}^{n+1}) = \alpha \mathbf{K}\mathbf{d}^n - (1 + \alpha)\mathbf{K}\mathbf{d}^{n+1}, \tag{48}$$

$$s_1(\mathbf{a}^n, \mathbf{a}^{n+1}) = \left(\frac{1}{2} - \beta\right)\mathbf{a}^n + \beta \mathbf{a}^{n+1}, \tag{49}$$

$$s_2(\mathbf{a}^n, \mathbf{a}^{n+1}) = (1 - \gamma)\mathbf{a}^n + \gamma \mathbf{a}^{n+1}, \tag{50}$$

with $\mathbf{d}^n = \mathbf{d}(t^n)$, $\mathbf{v}^n = \mathbf{v}(t^n)$, $\mathbf{a}^n = \mathbf{a}(t^n)$, $\mathbf{K}$ stiffness matrix, $\mathbf{M}$ mass matrix, $\mathbf{F}$ external forces vector, $\beta = (1-\alpha)^2/4$, $\gamma = 1/2 - \alpha$, and $\alpha \in [-1/3, 0]$.

3.2 Structural elements

In some specific situations, the solid domain can be conveniently modeled by two-dimensional structural elements like shells or membranes, or mono-dimensional structural elements such as beams.

In particular, the shell elements are based on a 2D parametrisation space $\mathcal{P}_1$ that represents the middle surfaces of the solid, namely $\mathcal{P}_1 \subset \mathbb{R}^2$. Depending on the specific assumptions made about the structure's dimensions, different shell theories can be formulated. In this work, thin structures, in which one dimension, the thickness h, is significantly smaller than the others, are considered. These elements can be adequately described by classical (Kirchhoff) shell theory [68]. In particular, 3-node triangular elements with piecewise linear displacement are adopted (`S3` in Abaqus notation). They provide accurate results in most loading situations. However, because of their constant bending and membrane strain approximations, high mesh refinement may be required to capture pure bending deformations or solutions to problems involving high strain gradients.

In addition to the translation degrees of freedom (DOFs), the shell element also includes three additional rotational DOFs for each node, resulting in a total of six DOFs per node. The translation problem has already been addressed in Sect. 3.1.

In this formulation, when we overlook the bending stiffness and shear stiffness related to the shell formulation, we obtain a membrane element. As a consequence, membrane elements do not have rotational DOFs. In this work, 3-node membrane elements (`M3D3`) are adopted for discretising dragonfly wings.

Finally, the beam element derive from a 1D parametrisation space $\mathcal{P}_2$ that represents the axis of the solid, namely $\mathcal{P}_2 \subset \mathbb{R}$. In this context, the classical Timoshenko assumption is the simplest approach to beam theory [69]. The model takes into account shear deformation and rotational bending effects, which makes it suitable for describing the behaviour of thick beams. Here, linear beam elements (`B31`) are selected to discretise the venation pattern, that creates joints between distinct membranes, as it will be presented in Sect. 6. Moreover, we assume that the transverse shear behavior of Timoshenko beams is linear elastic with a fixed modulus and, thus, independent of the response of the beam section to axial stretch and bending. Furthermore, Timoshenko beams can be subjected to large axial strains, while strains are assumed to be small due to torsion [28].

4 Coupling scheme

In the fluid–structure interaction setting, for the well-posedness of the multiphysic problem, two interface conditions have to be satisfied. The kinematic condition states that the velocity on the fluid–structure interface matches the fluid and the solid solution, namely:

$$\mathbf{u} = \mathbf{v}, \quad \mathbf{x} \in \Gamma_{fsi}, \quad t \in [0, T], \tag{51}$$

while the dynamic condition establishes the stress continuity across the fluid–structure region, as follows:

$$(\boldsymbol{\sigma}_s - \boldsymbol{\sigma}_f) \cdot \mathbf{n} = \mathbf{0}, \quad \mathbf{x} \in \Gamma_{fsi}, \quad t \in [0, T], \tag{52}$$

where $\mathbf{n} = \mathbf{n}_f = -\mathbf{n}_s$.

Within this work, we employ a partitioned approach and strongly impose Eqs. (51)–(52). Moreover, we generate a conforming mesh between solid and fluid subdomains, and the coupling condition is numerically imposed at every timestep Δt^n, which is shared between both subproblems.

To manage the coupling conditions, we rely on the SIMULIA Co-Simulation Services (CSS), a software component for run-time coupling of simulation tools, including Dassault Systèmes simulation solver and third-party simulation software for multiphysic and multi-scale simulations [28].

4.1 Immersed surface

In this subsection, we deal with the particular case where $\Gamma^n_{fsi} = \mathcal{S}^n$ is a two-dimensional surface immersed in the fluid domain. This circumstance could arise when we examine a thin structure that can be represented using shell elements.

Let us denote with $\partial\mathcal{S}^n$ the one-dimensional surface's boundaries associated with $\mathcal{S}^n$. Moving forward, we will distinguish between two instances: if $\partial\mathcal{S}^n \subset \Omega^n_f$, then the surface is fully immersed within the fluid domain, while, if $\partial\mathcal{S}^n \subset \partial\Omega^n_f$, then the surface does not have free edges, but they belong to the fluid Dirichlet boundary. The mixed scenario, i.e. when $\partial\mathcal{S}^n = \partial\mathcal{S}^n_1 \cup \partial\mathcal{S}^n_2$, with $\mathcal{S}^n_1 \subset \Omega^n_f$ and $\mathcal{S}^n_2 \subset \partial\Omega^n_f$, follows as combination of the above mentioned ones. Moreover, we limit our investigation to the situation where $\partial\mathcal{S}^n_\beta$, $\beta = 1, 2$, does not change over time, namely neither free edges can be generated nor eliminated.

In the present approach, the immersed face is split into two identical copies [70], for instance, $\mathcal{S}^n_+$ and $\mathcal{S}^n_-$, ensuring they possess opposing orientations, specifically $\mathbf{n}^+_f = -\mathbf{n}^-_f$, for every point $\mathbf{x} \in \mathcal{S}^n$, as depicted in Fig. 1.

In this setting, it is possible to duplicate the DOFs on the surface and compute two fluid velocities, mesh velocity and pressure values for each physical node. Since on the

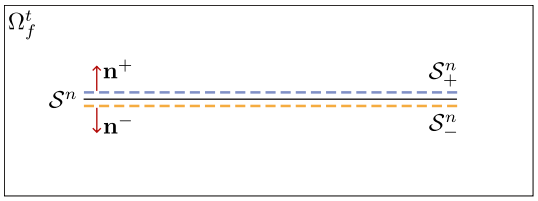

Fig. 1 2D example of split fluid–structure boundary $\mathcal{S}^n$ (black line) immersed in the fluid domain Ω_f^t, at fixed time t^n. The upper boundary is indicated with $\mathcal{S}_+^n$ (purple dashed line) and associated with normal unit vector $\mathbf{n}^+$, while the lower boundary is named $\mathcal{S}_-^n$ (orange dashed line) and it has unit normal vector $\mathbf{n}^-$

fluid–structure boundary no-slip conditions are prescribed, velocity and mesh velocity are kept piecewise continuous:

$$\mathbf{u}_H^{n,+} = \mathbf{u}_H^{n,-} = \mathbf{u}_H^n, \quad \mathbf{w}_H^{n,+} = \mathbf{w}_H^{n,-} = \mathbf{w}_H^n, \tag{53}$$

while the pressure field is allowed to experience a jump discontinuity across the immersed surface, namely:

$$p_H^{n,+} - p_H^{n,-} \neq 0, \tag{54}$$

for every $\mathbf{x} \in \mathcal{S}_H^n$.

Taking into account the discrete level, the splitting process necessitates the modification of the mesh connectivity by incorporating newly divided nodes. Additionally, for each $K^n \in \mathcal{T}_H^n$ such that $K^n \cap \mathcal{S}_H^n \neq \emptyset$, the appropriate DOF has to be assigned. This connectivity correction can be achieved by relying on geometric properties.

Let us firstly consider the case $\partial\mathcal{S}^n \subset \Omega_f^n$ and define the subset of tetrahedra $\mathcal{W}_H^n \subset \mathcal{T}_H^n$:

$$\mathcal{W}_H^n := \{K^n \in \mathcal{T}_H^n \ : \ K^n \cap \mathcal{S}_H^n \neq \emptyset\}. \tag{55}$$

As a consequence, $\mathcal{W}_H^n$ contains all the elements that share a geometrical entity $K^n \cap \mathcal{S}_H^n = M^n$ with the split surface, either it is a point, an edge or an entire triangle. In conclusion, the connectivity of $K^n \in \mathcal{W}_H^n$ is modified to represent $p_H^{n,+}$ if the following orientation condition holds:

$$(\mathbf{x}_B - \mathbf{x}_k) \cdot \mathbf{n}_E^+ > 0, \tag{56}$$

where $\mathbf{x}_B$ represents the centroid of the tetrahedron, $\mathbf{x}_k \in M^n$, and $\mathbf{n}_E^+$ is the normal vector associated to one of the linear triangle, e.g. E, in the fluid–structure interface mesh $\mathcal{S}_H^n$, in which M^n lies.

The second case where $\partial\mathcal{S}^n \subset \partial\Omega_f^n$ is treated equivalently, but for the surface's boundary. The subset $\mathcal{B}_H^n \subset \mathcal{C}_H^n \backslash \mathcal{S}_H^n$ of the triangular elements that share a geometrical variety with $\partial\mathcal{S}_H^n$, identified similar to Eq. (55) as:

$$\mathcal{B}_H^n := \{T^n \in \mathcal{C}_H^n \setminus \mathcal{S}_H^n \ : \ T^n \cap \partial\mathcal{S}_H^n \neq \emptyset\}. \tag{57}$$

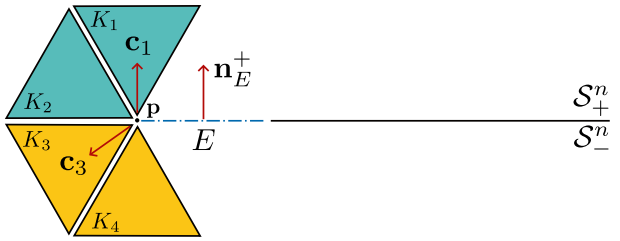

Fig. 2 2D example of fluid–structure boundary $\mathcal{S}^n$ split into $\mathcal{S}_+^n$ and $\mathcal{S}_-^n$ (see Fig. 1), with focus on the treatment of fluid elements that have geometric entities in common with $\partial\mathcal{S}^n$. The depicted fluid elements K_j, $j = 1, ..., 4$, share a vertex $\mathbf{p} = M^n$ with the boundary element E (dash-dot line), whose positive unit normal is $\mathbf{n}_E^+$. Since $\mathbf{c}_1 \cdot \mathbf{n}_E^+ > 0$, we assign to the element the degree of freedom of $\mathcal{S}_+^n$ (light blue). Reversely, $\mathbf{c}_3 \cdot \mathbf{n}_E^+ < 0$, meaning that we assign the degree of freedom of $\mathcal{S}_-^n$ (orange yellow)

If a tetrahedron belonging to $\mathcal{W}_H^n$ has also a not empty intersection with triangles in $\mathcal{B}_H^n$, we need to properly modify the connectivity of the boundary mesh.

It is worth noticing that the procedure described in this subsection, illustrated with a 2D example in Fig. 2 can be performed once with the initial mesh unless a remeshing occurs.

4.2 Gauss–Seidel scheme

The fluid and solid subproblems are coupled exploiting the Gauss-Seidel numerical scheme.

Gauss-Seidel method consists of a staggered and sequential scheme, where fluid and solid are solved iteratively until convergence on the interface region is achieved. More precisely, fixing a time step t^n, we first solve the leading subproblem (the solid one in our case) and then transfer the physical quantity to the slave subproblem (in our scenario, the fluid solver).

From the solid to the fluid solver, we pass the Dirichlet boundary condition on Γ_{fsi}^n, namely:

$$\mathbf{u}_H^n = \mathbf{v}^n, \quad \forall \mathbf{x} \in \mathcal{S}_H^n, \tag{58}$$

while from the fluid to the solid we impose a force distribution, computing the following integrated quantity:

$$\mathbf{f}_{fsi}^n = \int_{\mathcal{S}_H^n} \boldsymbol{\sigma}_f(p_H^n, \mathbf{u}_H^n) \cdot \mathbf{n} \, \mathrm{d}\Gamma. \tag{59}$$

The convergence criteria could be imposed either in the fluid–structure velocity or on the fluid–structure stresses or on both simultaneously, given a predefined relative tolerance for all the quantities to check. In particular, we define the velocity tolerance as ε_v, the stress tolerance as ε_f, and the maximum number of FSI iterations as q_m. We employed a

L^2 norm and the normalisation is performed with respect to the norm of the first subiteration.

Finally, in the case in which shell elements are involved in the solid simulation, from solid to fluid we transfer not only integrated nodal forces $\mathbf{f}^n_{fsi}$, but also concentrated momentum $\mathbf{C}^n_{fsi}$, namely:

$$\mathbf{f}^n_{fsi} = \mathbf{f}^{n,+}_{fsi} + \mathbf{f}^{n,-}_{fsi}, \tag{60}$$

$$\mathbf{C}^n_{fsi} = \mathbf{C}^{n,+}_{fsi} + \mathbf{C}^{n,-}_{fsi}, \tag{61}$$

where we define:

$$\mathbf{f}^{n,+}_{fsi} = \int_{\mathcal{S}^n_H} \boldsymbol{\sigma}_f(p^{n,+}_H, \mathbf{u}^n_H) \cdot \mathbf{n}^+ \, \mathrm{d}\Gamma, \tag{62}$$

$$\mathbf{f}^{n,-}_{fsi} = \int_{\mathcal{S}^n_H} \boldsymbol{\sigma}_f(p^{n,-}_H, \mathbf{u}^n_H) \cdot \mathbf{n}^- \, \mathrm{d}\Gamma, \tag{63}$$

$$\mathbf{C}^{n,+}_{fsi} = \int_{\mathcal{S}^n_H} \frac{h}{2}\mathbf{n}^+ \times \boldsymbol{\sigma}_f(p^{n,+}_H, \mathbf{u}^n_H) \cdot \mathbf{n}^+ \, \mathrm{d}\Gamma, \tag{64}$$

$$\mathbf{C}^{n,-}_{fsi} = \int_{\mathcal{S}^n_H} \frac{h}{2}\mathbf{n}^- \times \boldsymbol{\sigma}_f(p^{n,-}_H, \mathbf{u}^n_H) \cdot \mathbf{n}^- \, \mathrm{d}\Gamma. \tag{65}$$

4.3 Acceleration algorithm

Among the issues of the standard Gauss-Seidel scheme [24] we recall that convergence is not guaranteed and depends on the chosen time step [71]. A possibility of improving the performance of the Gauss–Seidel algorithm is the use of acceleration techniques, e.g. quasi-Newtonian methods [72, 73].

Given the fluid velocity $\mathbf{u}^n_{q-1}$ and the solid velocity $\mathbf{u}^n_q$ on the fluid–structure surface at fixed subiteration q, the relaxed Gauss-Seidel reads:

$$\mathbf{u}^n_q = \omega \mathbf{v}^n_q + (1-\omega)\mathbf{u}^n_{q-1}, \quad 1 \le q \le q_m, \tag{66}$$

where ω is the constant relaxation parameter and $\mathbf{u}^n_0$ is interpolated from the previous time. This method guarantees the convergence $\forall \omega \in (0, \omega_{opt}]$, $\omega_{opt} < 1$ [74], but the number of subiterations depends on the choice of ω, where $\omega = \omega_{opt}$ achieves the minimum [75]. Similarly, with the Aitken acceleration [27], we can speed up convergence by defining $\omega = \omega_q$ for every q:

$$\omega_q = -\omega_{q-1} \frac{(\mathbf{r}^{q-1})^T(\mathbf{r}^q - \mathbf{r}^{q-1})}{(\mathbf{r}^q - \mathbf{r}^{q-1})^T(\mathbf{r}^q - \mathbf{r}^{q-1})}, \tag{67}$$

where $\mathbf{r}^q = \mathbf{u}^n_q - \mathbf{u}^n_{q-1}$ is the residual at iteration q and ω_0 is user-defined.

5 Validation

The proposed approach is validated with three numerical benchmarks. Firstly, in Sect. 5.1 we propose a problem in which the structural shell coincides with a fluid boundary face. Then, in Sect. 5.2 we consider an immersed shell and validate the splitting surface algorithm. Finally, we extensively discuss the re-connection algorithm in Sect. 5.3.

5.1 Cavity flow with flexible bottom

To test the effectiveness of the proposed FSI framework in the presence of thin shells, the first benchmark is a 3D extension of the well-known driven cavity flow problem [76] including a flexible bottom. Given that the thin structure is situated on the fluid boundary, only the upper surface of the shell comes into contact with the fluid and serves as the designated reference surface.

The initial configuration of the fluid domain is shown in Fig. 3 and consists of a unit cube $\Omega^0_f = [0, l]^3$, with $l = 1.00$ m and a movable upper wall Γ_{up}. In this variant of the cavity flow problem, which has been explored in various contributions [77–79], the hard bottom of the cavity is replaced by a flexible thin shell Γ_{fsi}. Additionally, the originally constant top velocity is replaced by an oscillatory velocity, including the corner edges to avoid singularities. Moreover, inflow and outflow are permitted near the top face allowing the volume to change over time, while the initial configuration of the fluid is at rest. Boundary conditions are summarised as follows, for every $t \in (0, T]$:

$$\mathbf{u} = [f_1(t), 0, 0], \quad \mathbf{x} \in \Gamma_{up}, \tag{68}$$

$$\mathbf{u} = [f_2(t, z), 0, 0], \quad \mathbf{x} \in \Gamma_{in}, \tag{69}$$

$$p = 0, \quad \mathbf{x} \in \Gamma_{out}, \tag{70}$$

$$\mathbf{u} \cdot \mathbf{n}_f = 0, \quad \mathbf{x} \in \Gamma_{free}, \tag{71}$$

where $f_1(t) = 1 - \cos(2\pi t/5)$, $f_2(t, z) = (z - z_0)/(l - z_0) f_1(t)$, $z_0 = 0.875$ m, and $T = 50.0$ s. Finally, fluid density and kinematic viscosity are chosen as $\rho_f = 1.0$ kg/m^3 and $\nu_f = 0.01$ m^2/s, respectively.

The solid domain is defined as a square plate $\Omega^0_s = [0, l]^2 \times [0, -h]$ with a constant thickness $h = 0.2$ mm and with homogeneous Dirichlet boundary condition on all four edges. Furthermore, the solid material is selected as

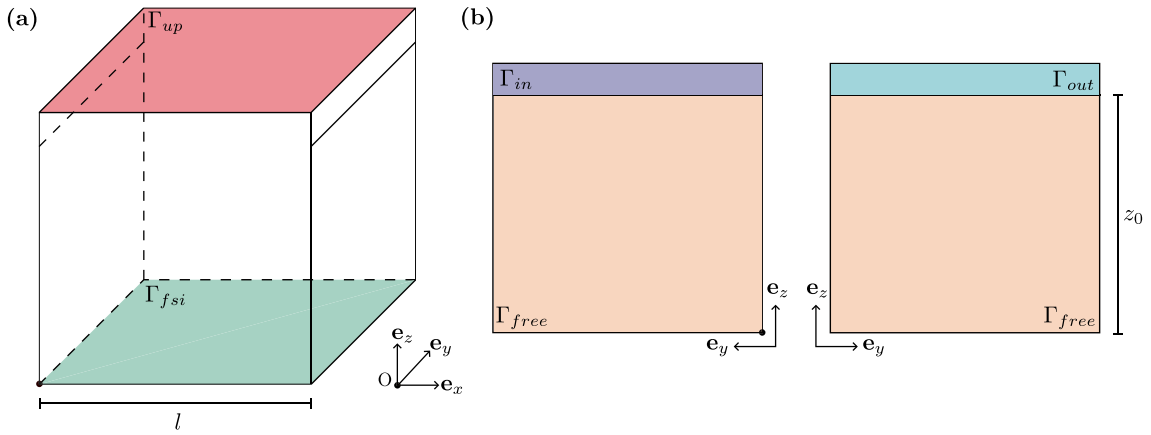

Fig. 3 Cavity with flexible bottom: three-dimensional representation of fluid domain Ω_f (**a**), and cartesian views of (**b**). Γ_{sup} is the upper face (red), Γ_{in} is the linear inlet portion (purple), Γ_{out} is the outflow boundary (light blue), and Γ_{free} is the boundary portion where zero normal velocity is imposed, as listed in Eq. (68). Moreover, $l = 1.00$ m is the length of the cube edge and $z_0 = 0.875$ m

linear elastic, with density $\rho_S = 500.0\ \text{kg/m}^3$, Poisson's ratio $\nu = 0.0$, and Young modulus $E = 250.0$ Pa.

To conduct numerical simulation, a constant timestep $\Delta t = 0.1$ s is chosen for both subproblems. The Aitken accelerated Gauss-Seidel coupling scheme is employed, with $q_m = 50$, and $\varepsilon_v = 10^{-4}$. In addition, we apply fluid subiteration, with $b_m = 4$, $\varepsilon_u = 10^{-4}$, and $\varepsilon_p = 10^{-3}$. The fluid mesh consists of 102,238 tetrahedral elements and 710,123 degrees of freedom, while the solid mesh is composed of 676 thin triangular shell elements, resulting in a total of 1,718 degrees of freedom. Across all the time frames, the average number of coupling iterations is $\bar{q} = 11$, with a maximum of $q_{\max} = 17$ and minimum of $q_{\min} = 7$.

Figure 4 displays the vertical displacement of the middle point of the bottom flexible surface, situated at coordinates $\mathbf{A} = (0.5, 0.5, 0.0)$ m in the initial configuration, while Table 1 provides a summary of the key features for comparison with existing literature results. Notably, all three solutions indicate an identical oscillatory period of $T = 5.12$ s. However, our solution exhibits amplitude and mean vertical displacement that closely aligns with the original reference, with an error of 0.7% and 2.3%, respectively. Lastly, the maximum vertical displacement of $d_{z,max} = 26.90$ cm occurs periodically at $t = 24.4\ \text{s} + kT$, $k \geq 0$, at coordinates $(x, y) = (41.55,\ 52.35)$ cm, as highlighted in Fig. 5.

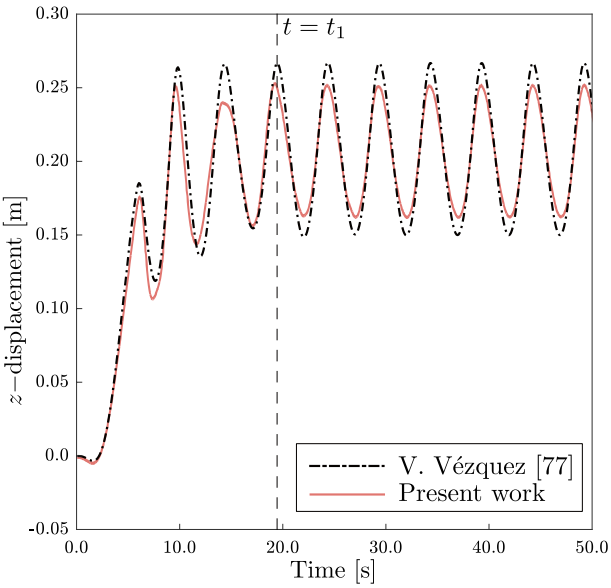

Fig. 4 Cavity with flexible bottom: vertical displacement of the middle point (**A**) of the flexible surfaces over time. The dark dash-dot line represents the reference solution in [77], and the pink line is the solution obtained with the proposed solver, where re-connection is not triggered. After a transitory time span of $t_1 = 19.2$ s, the trend matches the sinusoidal regime of the reference solution, with the same period $T = 5.12$ s

5.2 L-shaped obstacle

To validate the effectiveness of the split surface algorithm, a shell immersed in a fluid is here considered [80, 81].

Figure 6 depicts both fluid and solid geometries. It consists of a channel with a square section $\Omega_f^t = \Omega_f = [0, 5\,L] \times [0, 5\,L] \times [0, 33\,L]$, fixed in time. A thin L-shaped structure with a uniform thickness $h = 0.01$L is immersed in the channel. For simplicity, we set L = 1.00 m. The simulation is divided into two distinct steps. Initially, the structure is assumed to be rigid and fixed until the time $t_1 = 20.0$ s when the inflow is fully developed into the channel. At this point, the structure is allowed to move and deform until the final time $T = 40.0$ s. During all the analysis the structure is clamped at its base. Boundary conditions are summarised as follows, for every $t \in [0, T]$:

Table 1 Cavity with flexible bottom: quantitative comparisons between the proposed solution and literature. We can show that all simulations achieve the same oscillatory period T. Regarding the mean and amplitude values, the present work is in agreement with the literature results, with 0.7% and 2.3% percentage errors with respect to the original reference [79], and 3.7% and 11.7% percentage errors with respect to Fig. 4

Solution	Mean [cm]	Amplitude [cm]	T [s]
Present work	20.85	5.86	5.12
Valdez [77]	20.10	6.8	5.12
Mok [79]	21.0	6.0	5.12

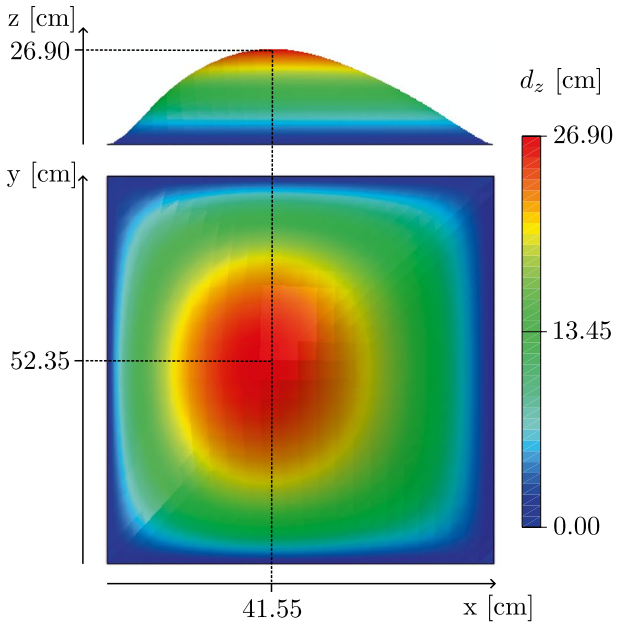

Fig. 5 Cavity with flexible bottom: solid z–displacement at time $t = 24.4$ s. It shows that the maximum value is not achieved in the middle, but slightly shifted towards the inflow boundary at $(x, y, z) = (41.55, 53.35, 26.90)$ cm

$$\mathbf{u} = \mathbf{u}_\infty, \quad \mathbf{x} \in \Gamma_{in}, \tag{72}$$

$$p = 0, \quad \mathbf{x} \in \Gamma_{out}, \tag{73}$$

$$\mathbf{u} \cdot \mathbf{n}_f = 0, \quad \mathbf{x} \in \Gamma_{lat}, \tag{74}$$

$$\mathbf{u} = \mathbf{0}, \quad \mathbf{x} \in \Gamma_d, \tag{75}$$

where $\mathbf{u}_\infty$ is a constant inlet velocity. Furthermore, we employ a linear elastic constitutive law for the solid, with $\rho_S = 1200.0\,\text{kg/m}^3$, $\nu = 0.32$, and $E = 3.5\,\text{GPa}$. For the fluid, we adopt a constant Reynolds number $Re = 200$ and a density of $\rho_f = 1.0\,\text{kg/m}^3$. Three distinct analyses have been

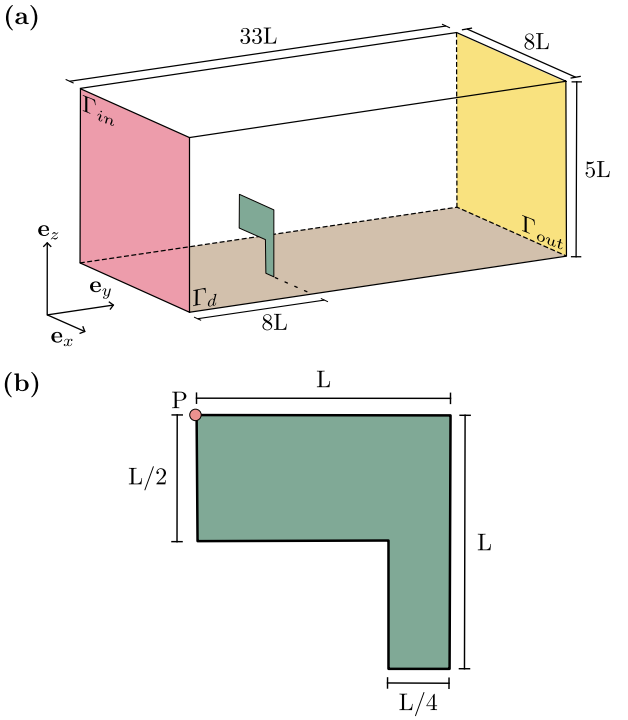

Fig. 6 L-shaped obstacle: three-dimensional representation of fluid geometry Ω_f (**a**), and solid geometry Ω_s (**b**). Γ_{in} is the inflow boundary (pink), Γ_d is the free slip boundary (brown), and Γ_{out} is the outflow boundary (yellow). The obstacle is a thin L-shaped structure with thickness $h = 0.01$L fixed on the bottom of the channel

conducted, encompassing variations in inlet velocity and the corresponding fluid viscosity, as outlined in Table 2.

The numerical simulations are performed with a constant timestep of $\Delta t = 0.01$ s in Case 1 and Case 2, while $\Delta t = 0.0025$ s in Case 3. For the problem at hand, we apply the Aitken acceleration scheme with parameter $q_m = 10$ and tolerances $\varepsilon_v = 10^{-4}$ and $\varepsilon_f = 10^{-3}$. Additionally, fluid subiterations are activated, with a limit of $b_m = 3$ and tolerances of $\varepsilon_u = 10^{-4}$ and $\varepsilon_p = 10^{-3}$. The fluid mesh consists of 95,000 tetrahedral elements, resulting in a total of 600,123 degrees of freedom, while the solid mesh is composed of 560 thin triangular shell elements, which yields a total of 952 degrees of freedom.

Figure 7 displays the pressure and velocity fields at the final time, emphasising the presence of a split surface that creates a pressure discontinuity across the obstacle, while Fig. 8 shows the streamlines and the vortex around the deformable structure. The displacement history of point P,

Table 2 L-shaped obstacle: kinematic viscosity ν_f and inlet velocity $\mathbf{u}_\infty$ for the three different cases, with $Re = 200$ fixed

Case	$\mathbf{u}_\infty [m/s]$	$\nu_f\,[m^2/s]$
1	5.48	0.0247
2	10.00	0.05
3	14.07	0.07035

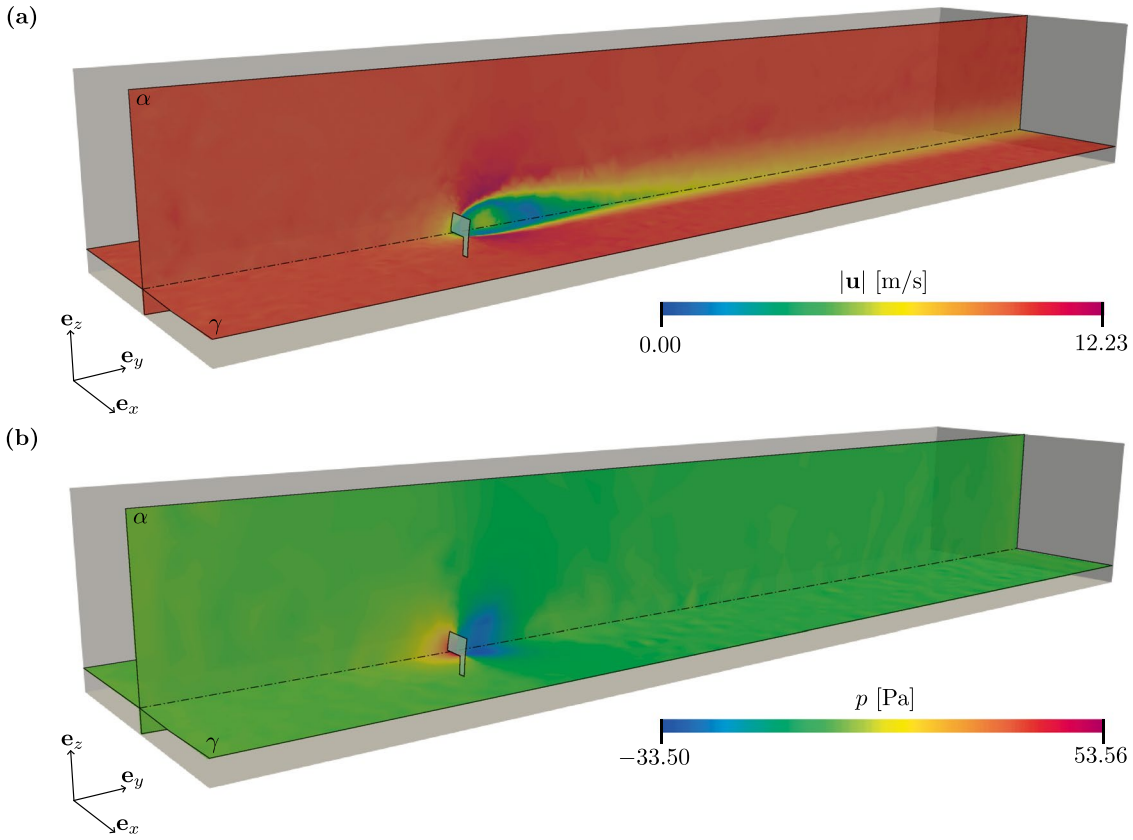

Fig. 7 L-shaped obstacle: fluid velocity magnitude (**a**) and normalised pressure (**b**) at time $t = T$ on two perpendicular planes $\alpha : x = 0$ m, and $\gamma : z = 0.6$ m for Case 2. It is worth to point out that a pressure discontinuity is achieved across the bi-dimensional obstacle

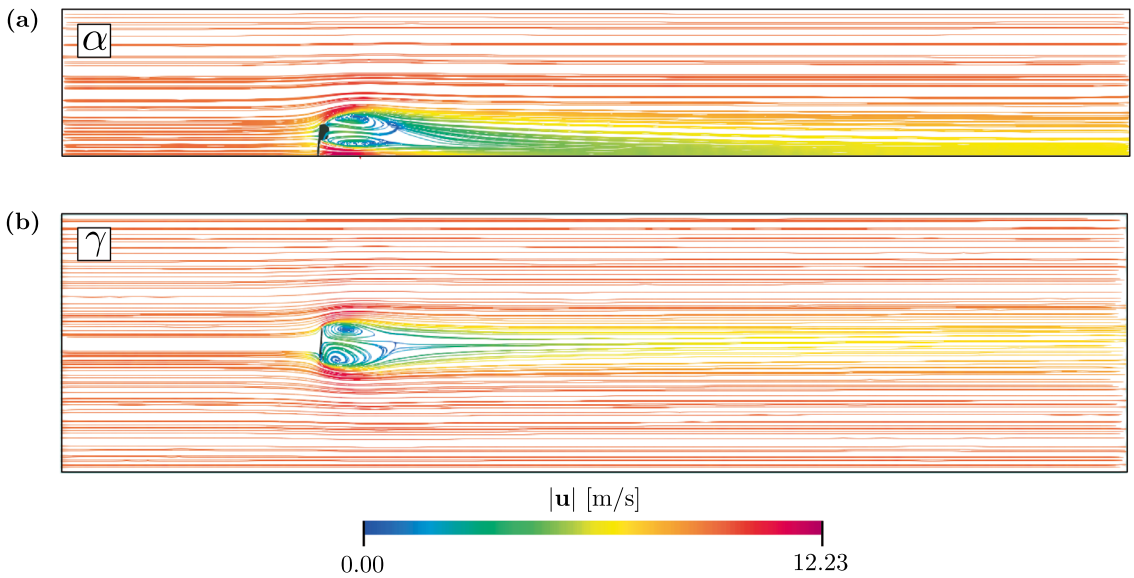

Fig. 8 L-shaped obstacle: streamlines representation at the final time $t = T = 40.00$ s on two perpendicular planes $\alpha : x = 0$ m (**a**), and $\gamma : z = 0.6$ m (**b**) for Case 2. The colour bar represents the fluid velocity magnitude over the streamlines

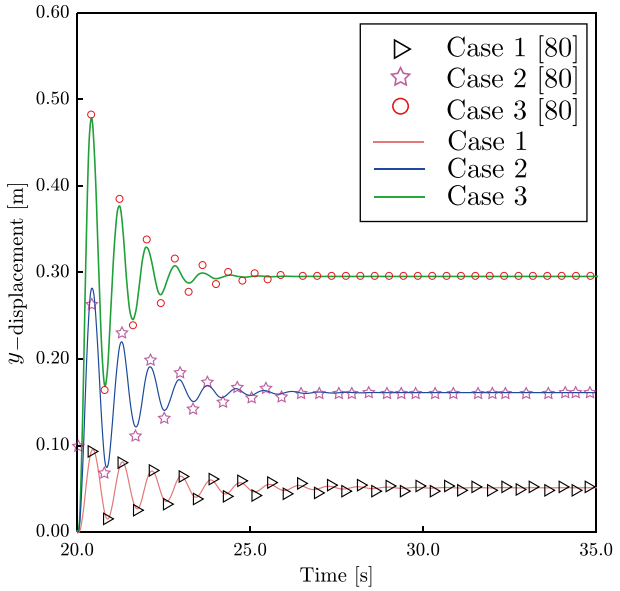

Fig. 9 L-shaped obstacle: displacement of point P in Case 1 (pink line), Case 2 (blue line), and Case 3 (green line) compared to the reference solutions [80]

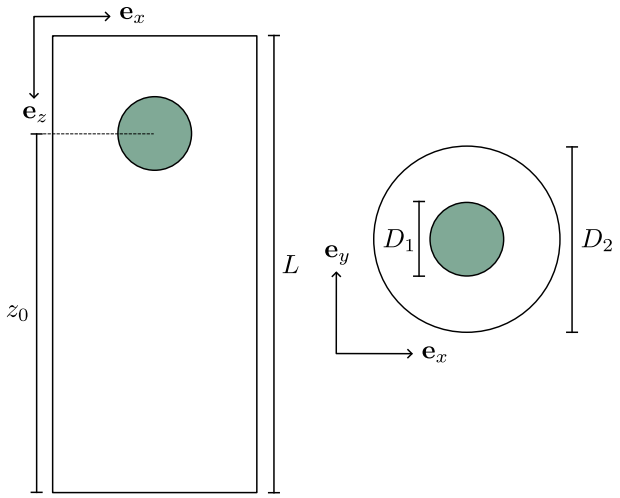

Fig. 10 Sphere falling in a channel: geometry of fluid domain Ω_f. It consists of a cylindrical channel of length $L = 60.00$ m and diameter $D_2 = 20.00$ m with a spherical obstacle fully immersed. The sphere diameter is $D_1 = 8.00$ m and its centre is located at $z_0 = 50.00$ m from the upper base

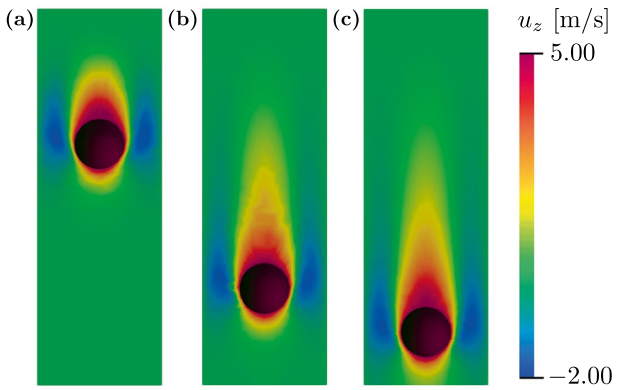

Fig. 11 Sphere falling in a channel: z-component of fluid velocity. The leftmost figure represents the fluid solution without performing re-connection (**a**), the middle figure shows the fluid solution when the re-connection is activated (**b**), while in the last figure re-connection plus refinement is toggled (**c**). The final time is selected for every case: $t_1 = 2.0$ s, $t_2 = 6.6$ s, and $t_3 = T = 8.0$ s, respectively

defined in Fig. 6, is illustrated in Fig. 9. The level of agreement with the findings in [80] can be considered satisfactory. It is worth stressing that these particular cases deal with small amplitude oscillations in the steady state regime, and the information transfer operator proves to be sensitive enough to capture these self-excited oscillations.

5.3 Sphere falling in a channel

To show the effectiveness of the re-connection algorithm, an ad-hoc benchmark has been developed. In this specific problem, the solid is considered rigid with a prescribed velocity and an one-way coupling is exploited.

Let us consider a cylindrical channel of radius $D_2 = 40.00$ m and height $L = 60.00$ m and a spherical obstacle of radius $D_1 = 8.00$ m and centred on the channel axis, $z_0 = 50.00$ m from the upper face (see Fig. 10). Moreover, the following boundary conditions are imposed, for $t \in [0, T]$:

$$\mathbf{u} = -v_z \boldsymbol{e}_z, \qquad \mathbf{x} \in \Gamma_s, \tag{76}$$

$$p = 0, \qquad \mathbf{x} \in \Gamma_d, \tag{77}$$

where Γ_s is the sphere surface, Γ_d is the bottom base of the cylinder, $v_z = 5.0$ m/s, and $T = 8.0$ s. Homogeneous Dirichlet boundary conditions are imposed on the remaining surfaces. In addition, we choose fluid density $\rho_f = 1.0$ kg/m^3, dynamic viscosity $\mu = 1.0$ m/s^2, $b_m = 10$, $\varepsilon_U = 10^{-4}$, $\varepsilon_p = 10^{-3}$, and $f_m = 50$.

In Figs. 11 and 12, the results of three distinct simulations are presented. In the first simulation on the left, the re-connection algorithm is not used; the analysis terminates at time $t_1 = 2.0$ s due to the detection of a flipped tetrahedron. In the middle simulation, standard re-connection is used, with a re-connection parameter of $Q^* = 0.05$. However, at time $t_2 = 6.6$ s, the algorithm is not able to generate a good quality mesh and the procedure stops. This issue can only be resolved by manually modifying the generator node or reducing the time step, as the distorted element count surpasses the predefined threshold. Lastly, in the rightmost

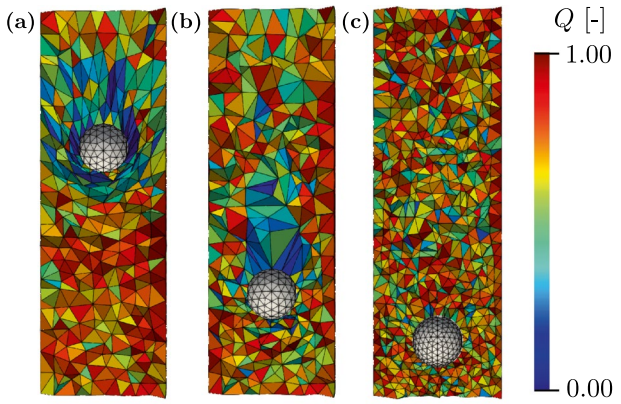

Fig. 12 Sphere falling in a channel: quality mesh Q on a xz-plane for the centre of the sphere. The leftmost figure represents the fluid solution without performing re-connection (**a**), the middle figure shows the fluid solution when the re-connection is activated (**b**), while in the last figure re-connection plus refinement is toggled (**c**). The final time is selected for every cases: $t_1 = 2.0$ s, $t_2 = 6.6$ s, and $t_3 = T = 8.0$ s, respectively

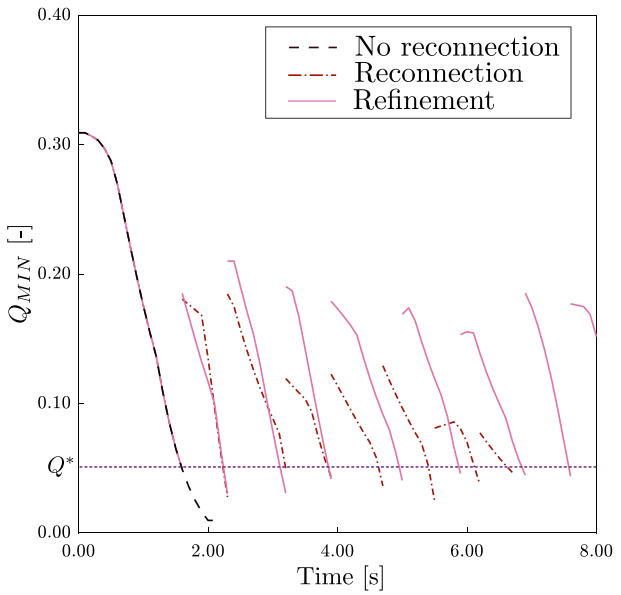

Fig. 13 Global quality mesh Q_{MIN} over time for no re-connection case (dashed black), re-connection case (dashed dot red), and refinement case (continuum pink). The horizontal line $Q^* = 0.05$ (dot purple) represents the threshold for re-connection. While the no re-connection case fails after $t_1 = 2.0$ s due to the flipped element, the re-connection case keeps the simulation running and after every re-connection instance the global indicator is above the threshold. However, because of the fact that the sphere undergoes large translation, a frequent change of mesh is required. Finally, the refined case prescribes a more accurate solution and a smaller number of re-connections that better regularise the mesh

simulation, we activate the refinement process with an assigned value of $\boldsymbol{\alpha} = [0.5, 1.0]$. Remarkably, no complications arise during this simulation. Additionally, upon observing the distribution of elements, it becomes evident that the refinement procedure leads to a more uniform mesh.

In Fig. 13, the global quality mesh indicator is depicted over time for each case. It is noteworthy to mention several observations. Firstly, the implementation of the refinement option consistently ensures a superior mesh quality with each re-connection. Secondly, it significantly reduces the frequency of re-connections required throughout the simulation duration. Lastly, we closely monitored the number of elements and nodes, ultimately deducing that the re-connection algorithm augmented the number of nodes and elements by approximately 10%.

6 Application to the flight of dragonflies

Dragonflies possess two pairs of wings, with the foremost pair referred to as the forewings and the rear pair known as the hindwings. These wings operate together to generate lift and smoothly manage the dragonfly's flight, greatly enhancing its remarkable manoeuvrability. The wings are connected to the thorax by a hinge joint, which grants them a significant level of mobility. This hinge joint allows the wings to move upwards and downwards, as well as to rotate, enabling precise control in flight [82]. Moreover, dragonfly wings frequently possess nano-structures that render them hydrophobic, thereby preventing waterlogging and enabling the dragonfly to sustain its agility even in harsh conditions [83].

In this investigation, in Sect. 6.2 we first focus on a single wing and conduct preliminary simulations to gauge the magnitude of the aerodynamic forces acting upon the wing structure. Next, in Sect. 6.3, we introduce the full model, dealing with both wings and their mutual interactions.

6.1 Wings model: structure and material

The complex and elaborate design of the hindwing is inspired by various sources in the literature [14, 84] and is depicted in Fig. 14a, where also the overall dimensions $l_{x,1} = 34.2$ mm and $l_{y,1} = 11.7$ mm are reported. The wing has a constant thickness of $h_1 = 3.6$ μm and it is modeled as a flexible membrane made of linear elastic material. The material properties are: density $\rho_s = 1200$ kg/m^3, Young modulus $E_1 = 3.75$ GPa, and Poisson's ratio $\nu_1 = 0.49$.

Furthermore, Fig. 14b emphasizes the prominent veins that contribute to the structural integrity of the wing. These veins are represented as cylindrical hollow beams in the model, also made of a linear elastic material with the same density as the wing membrane, Young modulus $E_2 = 6.00$ GPa, and Poisson's ratio $\nu_2 = 0.49$. In addition, a subdivision is introduced between primary and secondary veins having different outer radius and wall thickness,

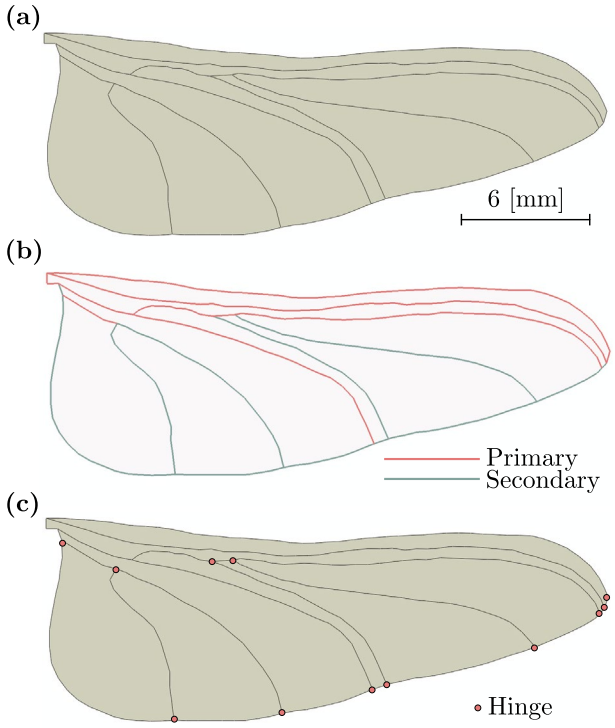

Fig. 14 Hindwing geometry: the initial configuration (**a**) has a characteristic lengths of $l_{x,1} = 34.2$ mm, $l_{y,1} = 11.7$ mm. Moreover, two levels of vein are defined in the model (**b**) based on their thicknesses. Finally, to allow relative movement between membranes, hinge constraints are imposed at the main intersection of veins **(c)**

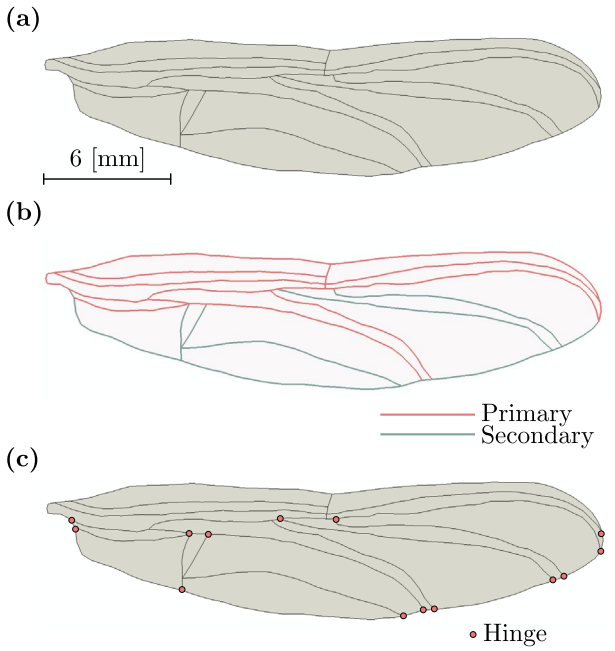

Fig. 15 Forewing geometry: the initial configuration (**a**) has a characteristic lengths of $l_{x,2} = 27.1$ mm, $l_{y,2} = 6.4$ mm. Moreover, two levels of vein are defined in the model (**b**) based on their thicknesses. Finally, to allow relative movement between membranes, hinge constraints are imposed at the main intersection of veins (**c**)

specifically $(r_1, w_1) = (0.1, 0.075)$ mm for the primary veins and $(r_2, w_2) = (0.045, 0.033)$ mm for the secondary ones. Finally, to allow relative movements between membranes, joints are inserted at the intersection between two vein branches, as shown in Fig. 14c. Moreover, to handle large deformation, geometric non linearity is enabled.

Next, we introduce the forewing design, as illustrated in Fig. 15. Compared to the hindwing, it presents a larger longitudinal span and is narrower, with

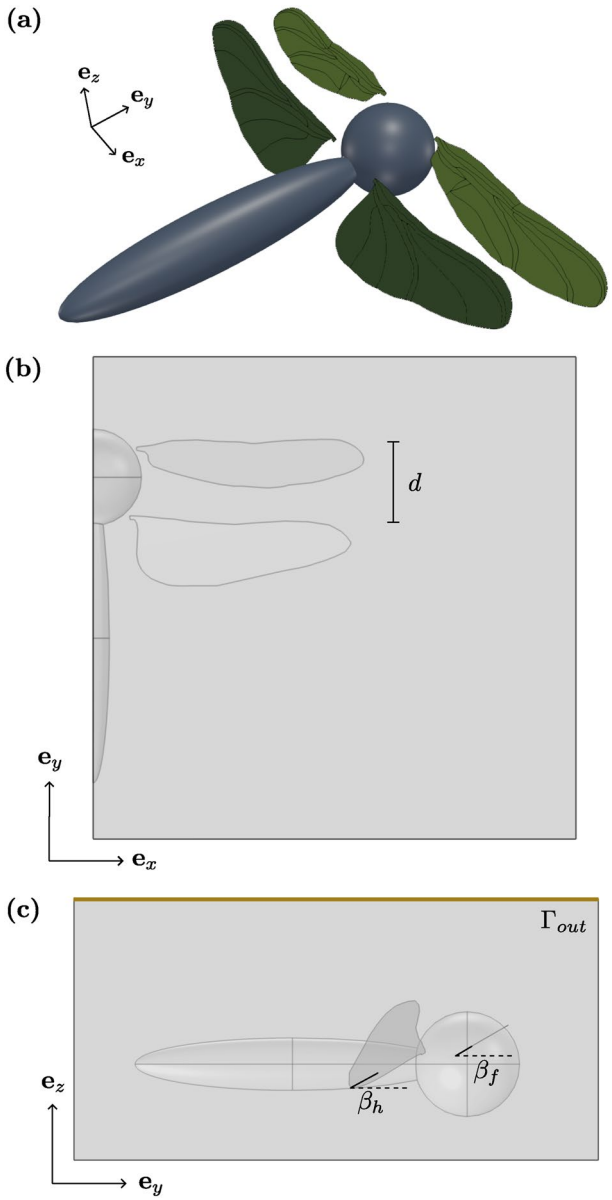

Fig. 16 Dragonfly: 3D visualisation of wings with a simplified sketch of thorax (**a**), modeled as the combination of a sphere of radius $r = 6.0$ mm and an ellipsoid of axes $(a, b, c) = (3.0, 2.0, 18.0)$ mm. Moreover, xy view (**b**) and yz view (**c**) are reported, highlighting the definition of angle of attach for hindwing ($\beta_h = 30°$) and forewing ($\beta_f = 45°$), as well as the distance between the two distinct wings $d = 5.00$ mm. Finally, Γ_{out} represents the outflow boundary (brown)

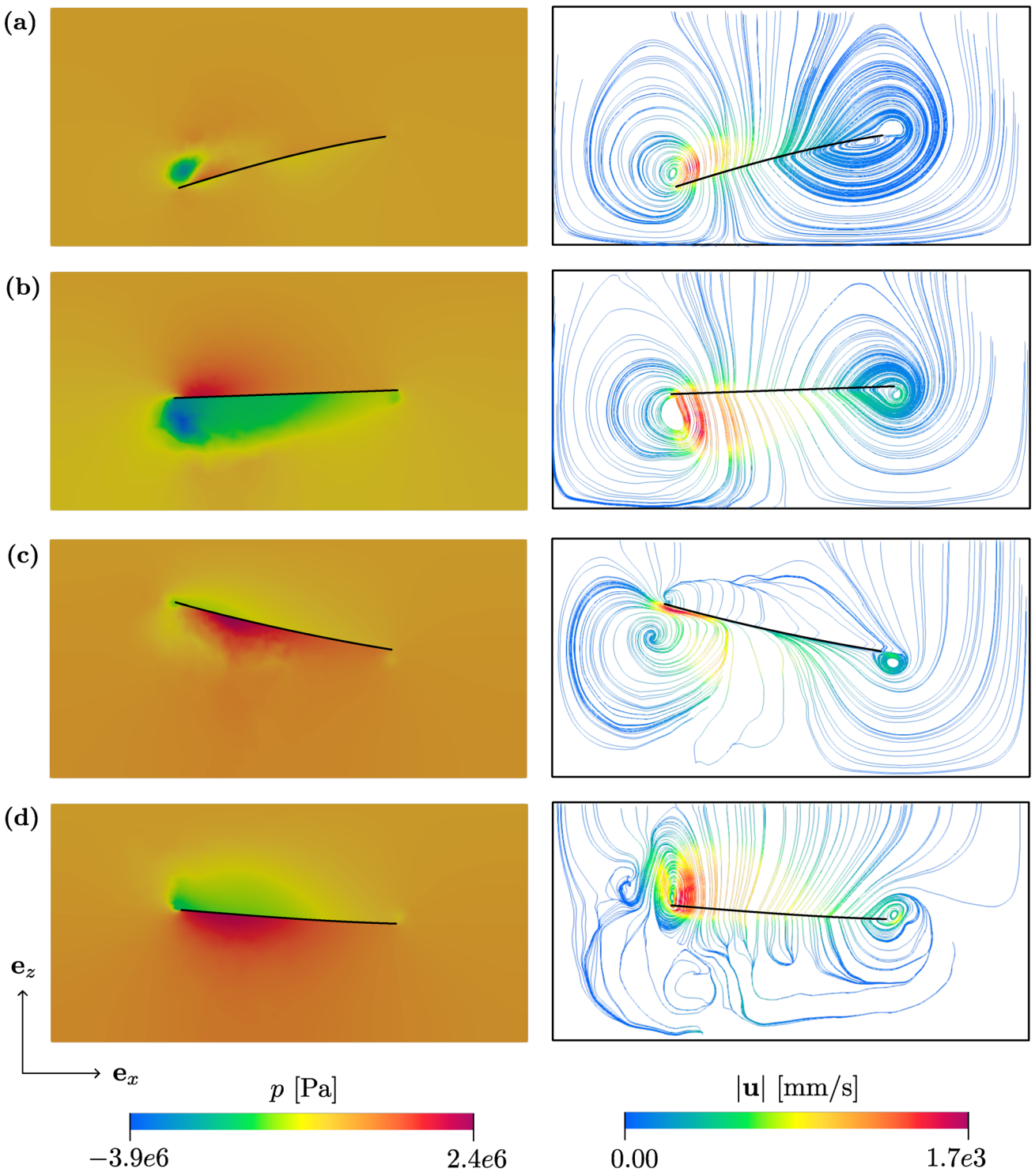

Fig. 17 Hindwing: normalised pressure and streamlines in a section generated by the $\mathbf{e}_x - \mathbf{e}_z$ plane passing through the centre of the fluid domain. The plots refer to four instants along the period: $T/4$ (**a**), $T/2$ (**b**), $3T/4$ (**c**), and T (**d**)

$l_{x,2} = 27.1$ mm and $l_{y,2} = 6.4$ mm. The forewing is thicker, with $h_2 = 7.2$ μm and the parameters of the two levels of veins are $(r_1, w_1) = (0.15, 0.11)$ mm for the primary veins and $(r_2, w_2) = (0.08, 0.06)$ mm for the secondary veins. The forewing geometry is depicted in Fig. 15a, while the vein pattern is highlighted in Fig. 15b and the hinge locations are illustrated in Fig. 15c. Finally, a three-dimensional representation of the full dragonfly is provided in Fig. 16a,

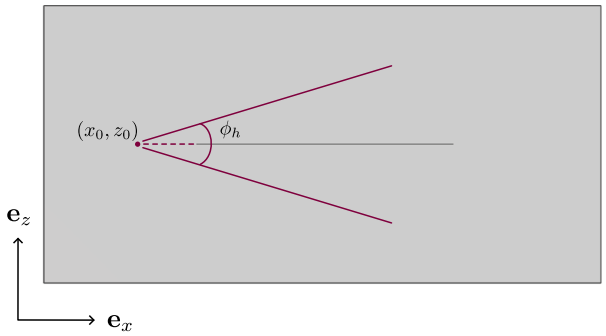

Fig. 18 Hindwing: the axis of rotation is parallel to $\mathbf{e}_y$ and located at $(x_0, z_0) = (-10.0, 0.0)$ mm. The oscillation has a frequency of $f = 32.3$ Hz and maximum amplitude of $\phi_h = 30°$

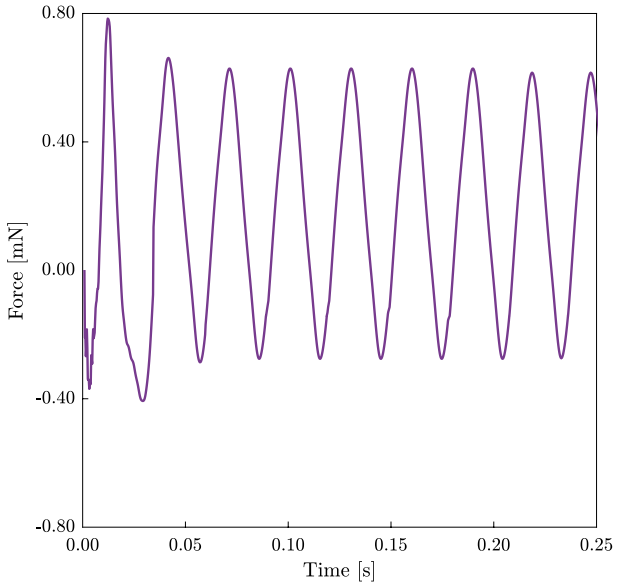

Fig. 19 Hindwing: lift force exerted on the wing's surface over time for eight periods

where the body is modeled as the combination of a rigid sphere of radius $r = 6.0$ mm and a rigid ellipsoid of axes $(a, b, c) = (3.0, 2.0, 18.0)$ mm. The distance between the two wings is about $d = 5.00$ mm. In this investigation, a simplified kinematic is adopted. For each wing, let us consider a local (x', y', z') reference frame centred at the attachment point between the wing and the body and with local axes parallel to the global ones. The wing oscillates around a fixed axis lying in the local y', z' plane and having an angle β with respect to y', called *angle of attack*. The selected angles are $\beta_h = 30°$ and $\beta_f = 45°$, respectively for the hindwings and the forewings. The cartesian views in Figs. 16b and c help clarify the adopted choices. Only half of the dragonfly is simulated assuming symmetry with respect to the global $x = 0$ plane.

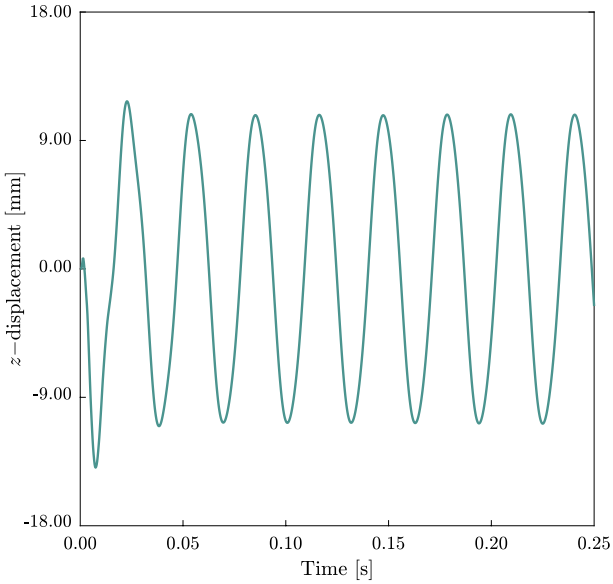

Fig. 20 Hingding: displacement of the leftmost point over time for eight periods

6.2 Hindwing model: results and discussion

Before addressing the full-model simulation in the following Section, we consider here a preliminary test in which only the hindwing is present and has vanishing angle of attack, i.e. $\beta_h = 0°$. The wing hence rotates around an axis parallel to $\boldsymbol{e}_y$ passing through the attachment point located at $(x_0, z_0) = (-10.0, 0.0)$ mm. The imposed oscillation is:

$$\Phi_h(t) = \phi_h \sin(2\pi f t), \quad t \in [0, T], \tag{78}$$

where $\phi_h = 10°$, $T = 8T_p$, $T_p = 1/f$, and $f = 32.3$ Hz, as illustrated in Fig. 18.

To simulate the movement of the hindwing, we consider a fluid domain represented by a rectangular box $\Omega_f = [0, 100] \times [-10, 10] \times [-10, 10]$. Zero pressure in imposed on Γ_{out}, while Dirichlet homogeneous boundary conditions are prescribed on the rest of the boundaries. The physical parameters of the fluid are set as $\rho_f = 1.2$ kg/m^3 and $\nu_f = 1.48 \times 10^{-5}$ m^2/s, leading to a Reynolds number of $Re \simeq 4000$ [85]. The mesh is made up of 120,000 tetrahedral elements and 600,134 degrees of freedom, while the mechanical part is modeled with 4650 triangular membrane elements and 812 linear beam Timoshenko elements. Moreover, the mesh size around the wing surface is sufficiently small to capture the leading edge vortices, as illustrated in [86, 87]. Aitken acceleration is employed, with a parameter value of $q_m = 15$ and tolerance of $\varepsilon_v = 10^{-4}$ and $\varepsilon_f = 10^{-3}$ and a constant time step $\Delta t = 0.0025$ s. Fluid subiterations are applied, with $b_m = 2$ and tolerances of $\varepsilon_u = 10^{-4}$ and $\varepsilon_P = 10^{-3}$, as well as re-connection with refinement, with $Q^* = 0.05$, $f_m = 50$, and $\boldsymbol{\alpha} = [0.4,\ 0.9]$.

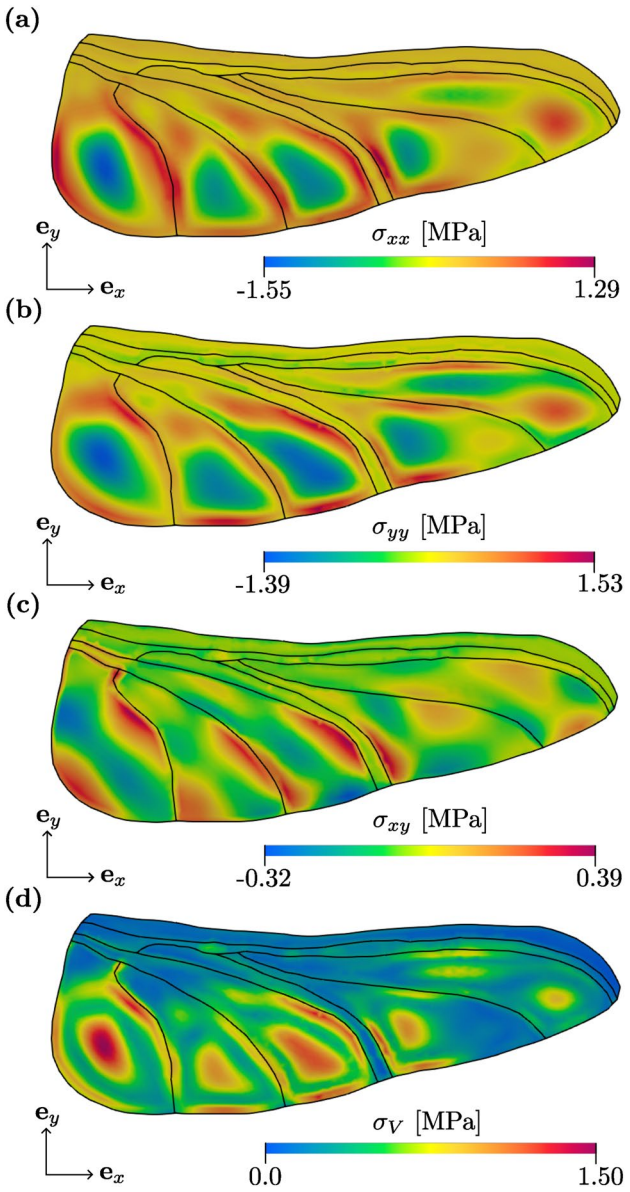

Fig. 21 Hingding: stress distribution at time $t = T$. The plots refer to different quantities: σ_{xx} (**a**), σ_{yy} (**b**), shear σ_{xy} (**c**), and Von Mises σ_V (**d**)

Figure 17 shows velocity magnitude, pressure and streamlines in the plane $y = 0$, for four time-steps in a single period T. We can notice the formation of a large vortex at the end edges and the pressure discontinuity across the membrane is evident. This is a crucial point because the lift of the dragonfly wing is due to the leading edge vortex effect, thus it is important to define the mesh in order to capture these phenomena. Furthermore, the choice of modeling the surrounding air as incompressible is justified, as the Mach number is much lower than the compressible threshold $Ma = 0.004 < Ma^* = 0.3$, computed at constant ambient temperature.

In Fig. 19, we compute the lift force exerted on the wing surfaces in the z−direction being the most meaningful component according to the flapping kinematics imposed. Secondly, we can notice that the force time history is periodic, with period T and a positive mean value. This result is expected, as we are neglecting the interaction with the second wing, and we are only enforcing a symmetric rotation around the y-axis. Lastly, to highlight the importance of the structural element, the displacement of the tip point is reported in Fig. 20 and the stress distribution is depicted in Fig. 21. We can notice that the displacement is periodic with period T, and a membrane behavior well captures the interaction with the air.

6.3 Dragonfly model: results and discussion

In this section, we present the results of the complete dragonfly model simulation. The wings move in anti-phase with the following imposed kinematics, for $t \in [0, T]$:

$$\Phi_h(t) = \Phi_h^0 + \phi_h \sin(2\pi f t), \tag{79}$$

$$\Phi_f(t) = \Phi_f^0 + \phi_f \sin(2\pi f t + \varphi_0), \tag{80}$$

where $\phi_h = \phi_f = 30°$, $\varphi_0 = -0.0077$. Moreover, the axis of rotation lies on the plane defined by the angle of attack and it is located at the hinge with the thorax, namely:

$$\boldsymbol{\omega}_h = \cos(\beta_h)\mathbf{e}_y + \sin(\beta_h)\mathbf{e}_z, \tag{81}$$

$$\boldsymbol{\omega}_f = \cos(\beta_f)\mathbf{e}_y + \sin(\beta_f)\mathbf{e}_z, \tag{82}$$

and the oscillation occurs around a stationary position $\Phi_h^0 = 15°$ and $\Phi_f^0 = 5°$, respectively. The fluid domain, depicted in Fig. 16b, consists of a fixed box $\Omega_f^t = \Omega_f = [0, 100] \times [-10, 10] \times [-10, 10]$, where a sphere and an ellipsoid are removed to model the dragonfly's body. The fluid mesh is composed of 151,121 tetrahedra and 689,123 degrees of freedom, while the solid mesh is composed of the hindwing mesh already presented in the previous Subsection and the forewing mesh is made up of 4121 membrane triangular elements and 765 linear beam elements. Regarding fluid parameters, they are chosen as in Sect. 6.1. The initial fluid domain is assumed at rest and a reference zero-pressure is imposed on the lateral face next to the wing's tips, while everywhere else but on fluid–structure interfaces homogeneous Dirichlet boundary conditions are prescribed.

Figure 22 depicts the streamlines around and between the two different wings. The leading edge vortex is correctly captured and additional vortexes are generated by the aerodynamic interaction of the two deformable wings. Finally, Fig. 23 shows the total lift force exerted on the

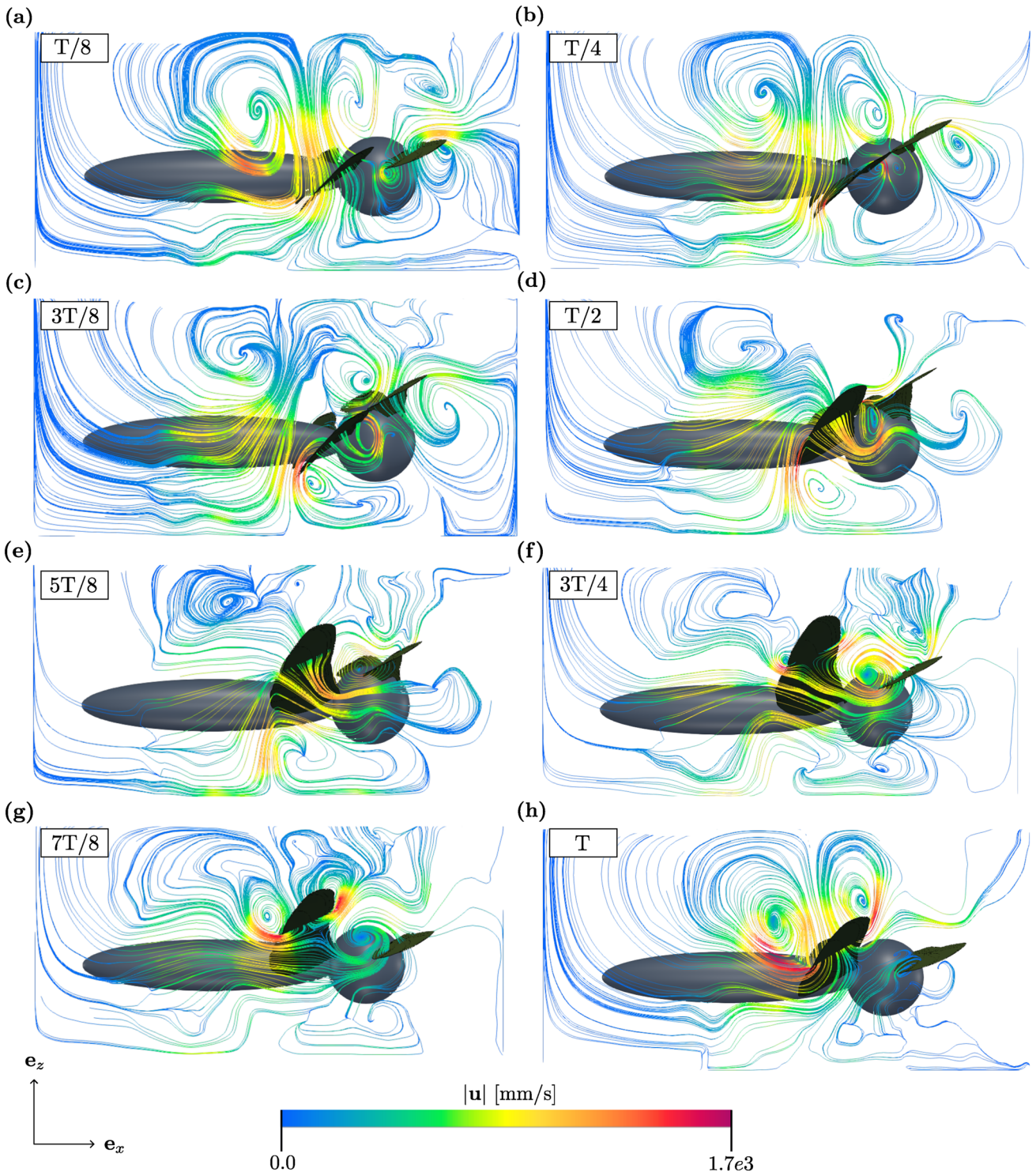

Fig. 22 Dragonfly: streamlines in a section generated by the $\mathbf{e}_x - \mathbf{e}_z$ plane passing through the centre of the fluid domain. The plots refer to several instants along the period: $T/8$ (**a**), $T/4$ (**b**), $3T/8$ (**c**), $T/2$ (**d**), $5T/8$ (**e**), $3T/4$ (**f**), $7T/8$ (**g**), and T (**h**)

dragonfly wings. Compared to the single wing analysis, the mean force value has increased by about 50 %, thus underlying the importance of the wing aerodynamic interaction during the flight.

7 Conclusions

In this paper, a partitioned approach has been employed to simulate FSI phenomena involving lower dimensional

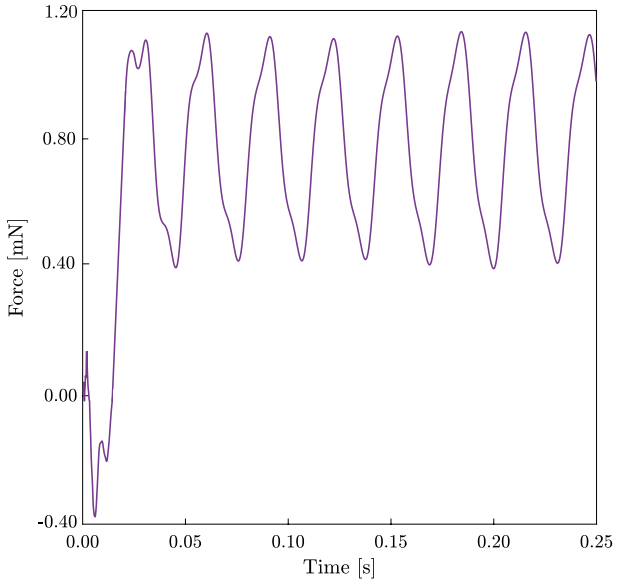

Fig. 23 Dragonfly: total lift force exerted on the wing surfaces over time for eight periods

structural Finite Elements. Regarding the fluid subproblem, incompressible Navier-Stokes equations have been solved with a Finite Element ALE approach, while the solid and coupling problems have been treated by the out-of-the-shelf software Abaqus. For this work, a novel reconnection algorithm has been developed and successfully applied to a simplified model of dragonfly MAV. Moreover, the bi-dimensional structural element has been represented in the fluid domain as an immersed surface, and the mesh connectivity has been modified duplicating the fluid–structure nodes. Firstly, the deformation of a single hindwing has been studied, where a rotation is imposed on the joint between the thorax and wings, maintained fixed over time. In this model, the solid part has been discretised with structural finite elements, such as membrane and beam, and a positive mean lift force is achieved. Secondly, we have enriched the dragonfly model incorporating also the forewing and a sketched thorax geometry. In this scenario, we have addressed the interaction between the two types of wings, highlighting the complex fluid dynamics and the creation of vortexes due not only to the movement of single wings but also to their anti-phase oscillations. Although we do not have a detailed model to capture with extreme accuracy the physical phenomena, the leading edge vortex generation has been reproduced. Among the several assumptions taken in the modelisation phase, the following ones have a more crucial role and might be considered for future developments. Corrugation phenomena are excluded, with the hypothesis of wings possessing uniform thickness. Additionally, physical properties are assumed constant throughout the analysis. Furthermore, the composite nature of the wing material has not been accounted for. Moreover, wettability considerations are disregarded as a contributing factor in the analysis. Concerning wing kinematics, we have considered a simple rotation, whereas the model could have been enriched by incorporating a more realistic motion, decomposed into rotations around different axes.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Data Availability All data are available upon reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Floreano D, Zufferey J-C, Srinivasan MV, Ellington C (2009) Flying insects and robots. Springer, Berlin
2. Mueller T (2001) Fixed and flapping wing aerodynamics for Micro Air Vehicle applications. Prog Astronaut Aeronaut 195:1–10
3. Benedict M, Ramasamy M, Chopra I (2010) Improving the aerodynamic performance of micro-air-vehicle-scale cycloidal rotor: an experimental approach. J Aircr 47(4):1117–1125
4. Moschetta J-M, Thipyopas C (2007) Aerodynamic performance of a biplane micro air vehicle. J Aircr 44(1):291–299
5. Davis WR, Kosicki BB, Boroson DM, Kostishack D (1996) Micro air vehicles for optical surveillance. Lincoln Lab J 9(2):197–214
6. Harne R, Wang K (2015) Dipteran wing motor-inspired flapping flight versatility and effectiveness enhancement. J R Soc Interface 12(104):20141367
7. Estrada MA, Mintchev S, Christensen DL, Cutkosky MR, Floreano D (2018) Forceful manipulation with micro air vehicles. Sci Robot 3(23):6903
8. Lentink D, Jongerius SR, Bradshaw NL (2010) The scalable design of flapping micro-air vehicles inspired by insect flight. Flying insects and robots, 185–205
9. Gerdes JW, Gupta SK, Wilkerson SA (2012) A review of bird-inspired flapping wing miniature air vehicle designs
10. Yang W, Wang L, Song B (2018) Dove: a biomimetic flapping-wing micro air vehicle. Int J Micro Air Vehicles 10(1):70–84
11. Nabawy MR, ElNomrossy M, Abdelrahman M, ElBayoumi G (2012) Aerodynamic shape optimisation, wind tunnel measurements and CFD analysis of a MAV wing. Aeronaut J 116(1181):685–708
12. Combes SA, Daniel T (2003) Flexural stiffness in insect wings i. scaling and the influence of wing venation. J Exp Biol 206(17):2979–2987

13. Hu Z, Deng X-Y (2014) Aerodynamic interaction between forewing and hindwing of a hovering dragonfly. Acta Mech Sin 30(6):787–799
14. Jongerius S, Lentink D (2010) Structural analysis of a dragonfly wing. Exp Mech 50:1323–1334
15. Tamai M, Wang Z, Rajagopalan G, Hu H, He G (2007) Aerodynamic performance of a corrugated dragonfly airfoil compared with smooth airfoils at low Reynolds numbers. In: 45th AIAA aerospace sciences meeting and exhibit, p. 483
16. Keennon M, Klingebiel K, Won H, Andriukov A (2012) Development of the nano hummingbird: a tailless flapping wing micro air vehicle 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. Nashville, TN, 9–12 January 2012)(AIAA: Reston, VA), 1–24
17. De Croon G, De Clercq K, Ruijsink R, Remes B, De Wagter C (2009) Design, aerodynamics, and vision-based control of the DelFly. Int J Micro Air Veh 1(2):71–97
18. Combes SA (2010) Materials, structure, and dynamics of insect wings as bioinspiration for MAVs. Encycl Aerosp Eng **7**(Part 34)
19. Coppola M, McGuire KN, De Wagter C, De Croon GC (2020) A survey on swarming with micro air vehicles: fundamental challenges and constraints. Front Roboti AI 7:18
20. Mueller TJ (1999) Aerodynamic measurements at low reynolds numbers for fixed wing micro-air vehicles. RTO AVT/VKI special course on development and operation of UAVs for military and civil applications 8:1–32
21. Aboelezz A, Hassanalian M, Desoki A, Elhadidi B, El-Bayoumi G (2020) Design, experimental investigation, and nonlinear flight dynamics with atmospheric disturbances of a fixed-wing micro air vehicle. Aerosp Sci Technol 97:105636
22. Lozovskiy A, Olshanskii MA, Vassilevski YV (2019) Analysis and assessment of a monolithic FSI finite element method. Comput Fluids 179:277–288
23. Heil M, Hazel AL, Boyle J (2008) Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. Comput Mech 43:91–101
24. Spenke T, Delaissé N, Degroote J, Hosters N (2023) On the number of subproblem iterations per coupling step in partitioned fluid-structure interaction simulations. arXiv preprint arXiv:2303.08513
25. Zhang Q, Hisada T (2004) Studies of the strong coupling and weak coupling methods in FSI analysis. Int J Numer Meth Eng 60(12):2013–2029
26. Löhner R, Cebral JR, Yang C, Baum JD, Mestreau EL, Soto O (2006) Extending the range and applicability of the loose coupling approach for FSI simulations. In: Fluid-Structure Interaction: Modelling, Simulation, Optimisation, pp. 82–100. Springer
27. Degroote J, Bruggeman P, Haelterman R, Vierendeels J (2008) Stability of a coupling technique for partitioned solvers in FSI applications. Comput Struct 86(23–24):2224–2234
28. Smith M (2014) ABAQUS/Standard User's Manual, Version 6.14. Dassault Systèmes Simulia Corp, United States
29. Xue D, Song B, Song W, Yang W (2016) Effect of wing flexibility on flight dynamics stability of flapping wing mavs in forward flight. Int J Micro Air Veh 8(3):170–180
30. Tirri A, Nitti A, Sierra-Ausin J, Giannetti F, Tullio MD (2023) Linear stability analysis of fluid-structure interaction problems with an immersed boundary method. J Fluids Struct 117:103830
31. Tay WB, Deng S, Van Oudheusden B, Bijl H (2015) Validation of immersed boundary method for the numerical simulation of flapping wing flight. Comput Fluids 115:226–242
32. Wang L, Tian F-B, Lai JC (2020) An immersed boundary method for fluid-structure-acoustics interactions involving large deformations and complex geometries. J Fluids Struct 95:102993
33. Zhang X, Theissen P, Schlüter J (2013) Towards simulation of flapping wings using immersed boundary method. Int J Numer Meth Fluids 71(4):522–536
34. Bolsman C, Goosen J, Keulen F (2009) Design overview of a resonant wing actuation mechanism for application in flapping wing mavs. Int J Micro Air Veh 1(4):263–272
35. Stanford B, Albertani R, Ifju P (2007) Static finite element validation of a flexible micro air vehicle. Exp Mech 47:283–294
36. Sarrate J, Huerta A, Donea J (2001) Arbitrary Lagrangian-Eulerian formulation for fluid-rigid body interaction. Comput Methods Appl Mech Eng 190(24–25):3171–3188
37. Anderson R, Elliott N, Pember R (2004) An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations. J Comput Phys 199(2):598–617
38. Saksono P, Dettmer W, Perić D (2007) An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction. Int J Numer Meth Eng 71(9):1009–1050
39. Berger MJ, Colella P (1989) Local adaptive mesh refinement for shock hydrodynamics. J Comput Phys 82(1):64–84
40. Ito Y, Murayama M, Yamamoto K, Shih AM, Soni BK (2009) Efficient computational fluid dynamics evaluation of small-device locations with automatic local remeshing. AIAA J 47(5):1270–1276
41. Alauzet F, Loseille A (2016) A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. Comput Aided Des 72:13–39
42. Di Cristofaro D, Opreni A, Cremonesi M, Carminati R, Frangi A (2022) An arbitrary Lagrangian Eulerian approach for estimating energy dissipation in micromirrors. In: Actuators, vol. 11, p. 298. MDPI
43. Re B, Dobrzynski C, Guardone A (2017) An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. J Comput Phys 340:26–54
44. Gaburro E, Boscheri W, Chiocchetti S, Klingenberg C, Springel V, Dumbser M (2020) High order direct Arbitrary-Lagrangian-Eulerian schemes on moving Voronoi meshes with topology changes. J Comput Phys 407:109167
45. Loubere R, Maire P-H, Shashkov M, Breil J, Galera S (2010) ReALE: a reconnection-based Arbitrary-Lagrangian-Eulerian method. J Comput Phys 229(12):4724–4761
46. Bank RE, Sherman AH, Weiser A (1983) Some refinement algorithms and data structures for regular local mesh refinement. Sci Comput Appl Math Comput Phys Sci 1:3–17
47. Casadei F, Diez P, Verdugo F (2013) An algorithm for mesh refinement and un-refinement in fast transient dynamics. Int J Comput Methods 10(04):1350018
48. Donea JH (2004) A., ponthot, j.-p, rodríguez-ferran, a.: Arbitrary lagrangian–eulerian methods. enc. Comput. Mech
49. Guzmán J, Salgado AJ, Sayas F-J (2013) A note on the Ladyženskaja-Babuška-Brezzi condition. J Sci Comput 56:219–229
50. Tezduyar TE (1991) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28:1–44
51. Tobiska L, Lube G (1991) A modified streamline diffusion method for solving the stationary navier-stokes equation. Numer Math 59(1):13–29
52. Blasco J, Codina R, Huerta A (1998) A fractional-step method for the incompressible Navier-Stokes equations related to a predictor-multicorrector algorithm. Int J Numer Meth Fluids 28(10):1391–1419
53. Codina R (2001) Pressure stability in fractional step finite element methods for incompressible flows. J Comput Phys 170(1):112–140

54. Murea C (2010) Arbitrary Lagrangian Eulerian approximation with remeshing for Navier-stokes equations. Int J Numer Methods Biomed Eng 26(11):1435–1448
55. Lee MJ, Do OhB, Kim YB (2001) Canonical fractional-step methods and consistent boundary conditions for the incompressible Navier-stokes equations. J Comput Phys 168(1):73–100
56. Codina R, Folch A (2004) A stabilized finite element predictor-corrector scheme for the incompressible Navier-stokes equations using a nodal-based implementation. Int J Numer Meth Fluids 44(5):483–503
57. Bottasso CL, Detomi D, Serra R (2005) The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. Comput Methods Appl Mech Eng 194(39–41):4244–4264
58. Liu S, Tang X, Li J (2022) A decoupled Arbitrary Lagrangian-Eulerian method for large deformation analysis of saturated sand. Soils Found 62(2):101110
59. Berzins M (1999) Mesh quality: a function of geometry, error estimates or both? Eng Comput 15:236–247
60. Sorgente T, Biasotti S, Manzini G, Spagnuolo M (2023) A survey of indicators for mesh quality assessment. In: Computer graphics forum, vol. 42, pp. 461–483. Wiley Online Library
61. Dapogny C, Dobrzynski C, Frey P (2014) Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. J Comput Phys 262:358–378
62. Shewchuk JR (2008) General-dimensional constrained Delaunay and constrained regular triangulations, i: Combinatorial properties. In: Twentieth anniversary volume: discrete & computational geometry, pp. 1–58. Springer, New York, NY
63. Pont A, Codina R, Baiges J (2017) Interpolation with restrictions between finite element meshes for flow problems in an ALE setting. Int J Numer Meth Eng 110(13):1203–1226
64. Vavourakis V, Loukidis D, Charmpis DC, Papanastasiou P (2013) Assessment of remeshing and remapping strategies for large deformation elastoplastic finite element analysis. Comput Struct 114:133–146
65. Bohbot J, Gillet N (2006) Impact of different mesh remapping techniques on 3D simulations in internal combustion engines. In: ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond Aan Zee, The Netherlands, September 5-8, 2006. Delft University of Technology; European Community on Computational Methods
66. Sohail MA, Ullah R (2011) CFD of oscillating airfoil pitch cycle by using piso algorithm. Eng Technol 60:1929–1933
67. Hilber HM, Hughes TJ (1978) Collocation, dissipation and [overshoot] for time integration schemes in structural dynamics. Earthq Eng Struct Dyn 6(1):99–117
68. Nguyen-Thanh N, Valizadeh N, Nguyen M, Nguyen-Xuan H, Zhuang X, Areias P, Zi G, Bazilevs Y, De Lorenzis L, Rabczuk T (2015) An extended isogeometric thin shell analysis based on Kirchhoff-love theory. Comput Methods Appl Mech Eng 284:265–291
69. Hutchinson J (2001) Shear coefficients for timoshenko beam theory. J Appl Mech 68(1):87–92
70. Tezduyar TE, Sathe S, Senga M, Aureli L, Stein K, Griffin B (2005) Finite element modeling of fluid–structure interactions with space–time and advanced mesh update techniques. In: Proceedings of the 10th international conference on numerical methods in continuum mechanics (CD-ROM), Zilina, Slovakia
71. Lorentzon J, Revstedt J (2022) On stability and relaxation techniques for partitioned fluid-structure interaction simulations. Eng Rep 4(10):12514
72. Delaissé N, Demeester T, Haelterman R, Degroote J (2023) Quasi-Newton methods for partitioned simulation of fluid–structure interaction reviewed in the generalized Broyden framework. Archiv Comput Methods Eng 1–30
73. Broyden CG (1967) Quasi-newton methods and their application to function minimisation. Math Comput 21(99):368–381
74. Reddy JN, Gartling DK (2010) The finite element method in heat transfer and fluid dynamics. CRC Press, United States
75. Winterstein A, Lerch C, Bletzinger K-U, Wüchner R (2018) Partitioned simulation strategies for fluid-structure-control interaction problems by gauss-seidel formulations. Adv Model Simul Eng Sci 5:1–29
76. Ghia U, Ghia KN, Shin C (1982) High-Re solutions for incompressible flow using the Navier-stokes equations and a multigrid method. J Comput Phys 48(3):387–411
77. Valdés Vázquez JG (2007) Nonlinear analysis of orthotropic membrane and shell structures including fluid-structure interaction
78. Fernandes JWD, Coda HB, Sanches RAK (2019) ALE incompressible fluid-shell coupling based on a higher-order auxiliary mesh and positional shell finite element. Comput Mech 63:555–569
79. Mok DP, Wall W (2001) Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. Trends in computational structural mechanics **1**
80. Glück M, Breuer M, Durst F, Halfmann A, Rank E (2003) Computation of wind-induced vibrations of flexible shells and membranous structures. J Fluids Struct 17(5):739–765
81. Nitti A, Kiendl J, Reali A, Tullio MD (2020) An immersed-boundary/isogeometric method for fluid-structure interaction involving thin shells. Comput Methods Appl Mech Eng 364:112977
82. Azuma A, Azuma S, Watanabe I, Furuta T (1985) Flight mechanics of a dragonfly. J Exp Biol 116(1):79–107
83. Gorb S, Kesel A, Berger J (2000) Microsculpture of the wing surface in odonata: evidence for cuticular wax covering. Arthropod Struct Dev 29(2):129–135
84. Sun J, Bhushan B (2012) The structure and mechanical properties of dragonfly wings and their role on flyability. Comptes Rendus Mécanique 340(1–2):3–17
85. Swift KM (2009) An experimental analysis of the laminar separation bubble at low reynolds numbers
86. Chen L, Cheng B, Wu J (2023) Vorticity dynamics and stability of the leading-edge vortex on revolving wings. Phys Fluids **35**(9)
87. Bomphrey RJ, Nakata T, Henningsson P, Lin H-T (2016) Flight of the dragonflies and damselflies. Philos Trans R Soc B: Biol Sci 371(1704):20150389

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.