

A fixed-grid b-spline finite element technique for fluid–structure interaction

T. Rüberg and F. Cirak*,†

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

SUMMARY

We present a fixed-grid finite element technique for fluid–structure interaction problems involving incompressible viscous flows and thin structures. The flow equations are discretised with isoparametric b-spline basis functions defined on a logically Cartesian grid. In addition, the previously proposed subdivision-stabilisation technique is used to ensure inf–sup stability. The beam equations are discretised with b-splines and the shell equations with subdivision basis functions, both leading to a rotation-free formulation. The interface conditions between the fluid and the structure are enforced with the Nitsche technique. The resulting coupled system of equations is solved with a Dirichlet–Robin partitioning scheme, and the fluid equations are solved with a pressure–correction method. Auxiliary techniques employed for improving numerical robustness include the level-set based implicit representation of the structure interface on the fluid grid, a cut-cell integration algorithm based on marching tetrahedra and the conservative data transfer between the fluid and structure discretisations. A number of verification and validation examples, primarily motivated by animal locomotion in air or water, demonstrate the robustness and efficiency of our approach. Copyright © 2013 John Wiley & Sons, Ltd.

Received 4 August 2013; Accepted 2 October 2013

KEY WORDS: finite elements; fluid–structure interaction; Navier–Stokes; pressure–correction method; b-splines; isogeometric analysis

1. INTRODUCTION

There are a multitude of challenges in fluid–structure analysis, which depend, amongst others, on the characteristic dimensionless flow parameters, such as the Reynolds and Mach numbers, the geometric complexity of the fluid–structure interface, the extent of structural deformations and the ratio of the fluid to the solid density. The state of the art in computational fluid–structure interaction is well advanced so that most of the aforementioned challenges have been widely studied, and a variety of complementary solution techniques are available. Nevertheless, one of the still open questions is the tracking of geometrically complex fluid–structure interfaces in case of large deformations and possible topology changes. One application area in which robust and efficient interface tracking is crucial is the interaction of fluids with lightweight structures, such as shells, membranes and beams, which are prone to large deformations. This is, for instance, relevant for the study of animal locomotion in air or water with beam or shell-like appendages.

It is becoming apparent that immersed, or embedded, boundary discretisation methods are particularly suitable for fluid–structure interaction problems with large interface deformations. Motivated by the original work of Peskin [1], a number of new finite element based immersed discretisation methods have recently been proposed, see, for instance, [2–6]. In these

*Correspondence to: F. Cirak, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.

†E-mail: f.cirak@eng.cam.ac.uk

methods, the fluid flow is considered in an Eulerian frame of reference, and the solid, that may undergo large deformations, is considered in a Lagrangian frame of reference. Consequently, at the fluid–structure interface, the fluid and the structure discretisations are nonmatching, and the interface conditions need to be enforced using auxiliary techniques. Our method of choice for enforcing the interface equilibrium conditions at the fluid–structure interface is the Nitsche technique [6, 7].

In immersed methods, the fluid equations are usually discretised with a logically Cartesian grid because of its algorithmic simplicity and efficiency. This in turn facilitates the use of tensor-product b-spline basis functions, which have a number of appealing properties as known from isogeometric analysis [8]. In this paper, we discretise the incompressible Navier–Stokes equations with the subdivision-stabilisation technique previously introduced in Rüberg *et al.* [5]. The velocity and pressure fields are discretised with b-splines of same polynomial degree using the same grid. The inf–sup stability condition is satisfied by recourse to the refinability property of b-splines. For alternative approaches for ensuring inf–sup stability with b-splines see [9, 10]. On the Cartesian grid, the enforcement of the Dirichlet and fluid–structure interface conditions is conceptually identical. To this end, as in Rüberg *et al.* [5], we use the Nitsche method in combination with a stabilisation technique to sidestep the numerical instabilities associated with very small cut-cells. Different from [5], in the present implementation, we obtain the computational grid by an isoparametric mapping from a uniform tensor-product grid. Hence, the grid lines can be curved, and the size of the cells does not need to be uniform.

In the applications considered in this paper, the structure is either a beam or a shell. The structure is assumed to be thin, and the Euler–Bernoulli and the Kirchhoff–Love models are used for the beam and the shell, respectively. The only unknowns in the corresponding weak forms are the displacements of the structure. Because the weak forms contain the second order derivatives of the displacements, they have to be discretised with smooth basis functions. In this work, we discretise beams with cubic b-splines and shells with Loop subdivision basis functions [11–13]. The Loop basis functions are able to provide smooth interpolants even on unstructured meshes and reduce to quartic box-splines on structured meshes. The absence of rotational degrees of freedom in spline-based finite elements makes them theoretically and algorithmically very appealing for fluid–structure coupling [14].

The efficient and robust solution of the discretised fluid–structure equations is challenging [15–19]. As mentioned, in our implementation the interface conditions between the fluid and the structure are imposed with the Nitsche method. We solve the resulting discrete system of equations with a partitioned approach, which is algebraically equivalent to a block Gauss–Seidel method. The coupling is implicit (or strong) in the sense that within each time step the fluid and the structure equations are iterated until the interface conditions are satisfied. As shown in Burman *et al.* [7], the partitioning of the discretised fluid–structure equations with Nitsche coupling naturally leads to a Dirichlet–Robin coupling scheme. In each iteration step, the fluid is solved with the known structure velocities as the Dirichlet boundary condition, and subsequently, the structure is solved with the known fluid tractions and velocities as the Robin boundary condition. The consistent transfer of variables between the fluid and the structure is crucial because the interface is by construction nonmatching [20].

The outline of this paper is as follows. Section 2 begins with a review of the subdivision-stabilised b-spline finite elements for incompressible Navier–Stokes equations and discusses its generalisation to non-equidistant Cartesian grids. More specifically, in Section 2.1, the solution of the flow equations with an incremental pressure-correction scheme is introduced, and the weak enforcement of Dirichlet boundary conditions with the Nitsche method is discussed. In Section 2.2, the discretisation with b-spline basis functions and associated algorithmic aspects, including implicit boundary representation, cut-cell stabilisation and cut-cell integration, are reviewed. Subsequently, in Section 3, an isogeometric Euler–Bernoulli type beam element for finite displacements is introduced. The coupling of the fluid and structure solvers with a Dirichlet–Robin partitioned approach is considered in Section 4. Finally, in Section 5, we study the interaction of flows with rigid and flexible structures. Most of the included examples are motivated by animal locomotion in air or water.

2. SUBDIVISION-STABILISED FLUID FINITE ELEMENTS

In the following, we introduce the subdivision-stabilised b-spline fluid finite elements proposed in our previous paper [5]. In [5], we solved the Navier–Stokes equations with an implicit scheme; whereas in the present work, we use a pressure-correction method. In comparison with the fully implicit scheme, the pressure-correction method leads to substantially smaller systems of equations. For the considered class of problems, this leads to solution schemes that are more efficient in terms of computing time and storage. As shown in Appendix A, the pressure-correction scheme can also be iteratively applied in order to recover the implicit Navier–Stokes solution.

2.1. Governing equations

2.1.1. Strong form. The initial–boundary value problem for an incompressible viscous fluid which occupies the domain Ω^f is given by

$$\rho^f \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{0} \quad \text{in } \Omega^f \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega^f \quad (1b)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_D^f \quad (1c)$$

$$\mathbf{t}(\mathbf{u}, p) = \boldsymbol{\sigma}(\mathbf{u}, p) \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_N^f \quad (1d)$$

$$\mathbf{u}(\cdot, 0) = \mathbf{0} \quad \text{in } \Omega^f. \quad (1e)$$

In the above, (1a) and (1b) are the momentum and mass balance equations, respectively; ρ^f is the mass density; \mathbf{u} is the velocity vector; $\boldsymbol{\sigma}$ is the Cauchy stress tensor; and p is the pressure. As boundary conditions, on the Dirichlet boundary Γ_D^f the velocity is prescribed to be $\bar{\mathbf{u}}$; and on the outflow boundary Γ_N^f with the normal \mathbf{n} the traction \mathbf{t} is prescribed to be zero.

For an incompressible Newtonian fluid, the constitutive equation depends only on a scalar fluid viscosity μ and is given by

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p \mathbf{I} + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^\top). \quad (2)$$

As will be seen in the next section, when a pressure-correction method is used for solving (1), it is advantageous to consider the conventional form of the momentum balance (1a) that is

$$\rho^f \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{0}, \quad (3)$$

which is obtained by introducing the constitutive equation (2) into (1a) and using the incompressibility condition (1b). The corresponding Neumann boundary condition is slightly different from (1d) and reads

$$-p \mathbf{n} + \mu \partial_{\mathbf{n}} \mathbf{u} = \mathbf{0} \quad (4)$$

with the derivative in the normal direction $\partial_{\mathbf{n}} \mathbf{u} = (\nabla \mathbf{u}) \mathbf{n}$.

2.1.2. Time semi-discretisation and pressure-correction method. The Navier–Stokes equations are discretised in time by subdividing the interval of interest $[0, T]$ into subintervals $0 = t^0 < t^1 < \dots < t^n < \dots < t^N = T$. For simplicity, in the following, we assume a constant time interval length $\Delta t = t^{n+1} - t^n$ and use the backward Euler method. At the time instant t^{n+1} , the momentum and mass balance equations read

$$\frac{\rho^f}{\Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n) + \rho^f (\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1} - \mu \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{0} \quad (5a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (5b)$$

where the superscripts denote the time instant. Note that the considered pressure-correction method is not restricted to the first order backward Euler method, many other choices are possible.

The nonlinear time-discretised Navier–Stokes equations (5) comprise $d + 1$ coupled nonlinear equations (d being the space dimension). When a fully implicit solution scheme is used at each time instant, the nonlinear system has to be solved with an iterative procedure, such as the Newton–Raphson method. In a semi-implicit scheme, the nonlinear convection term $(\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1}$ in the momentum conservation (5a) is replaced by $(\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1}$ so that only the linear equation

$$\frac{\rho^f}{\Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n) + \rho^f (\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} - \mu \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{0} \quad (6)$$

has to be solved in conjunction with (5b).

It is computationally appealing to solve the time-discretised semi-implicit Navier–Stokes equation (6) with a pressure-correction, or a fractional-step, method. Pressure-correction methods provide a simple means to decouple the computation of the velocity and pressure fields. They have been pioneered by Chorin [21] and Temam [22] and have a rich history in computational fluid dynamics. A review and mathematical analysis of some of the prevalent pressure-correction methods can be found, for example, in [23, 24]. The specific method we use is the incremental pressure-correction method in *rotational* form as summarised below. First, we define a pressure increment ψ^k in order to keep the subsequent expressions compact,

$$\psi^{n+1} = p^{n+1} - p^n + \mu \nabla \cdot \mathbf{u}^{n+1}. \quad (7)$$

With this definition at hand, the algorithm for computing the unknown velocity \mathbf{u}^{n+1} and pressure p^{n+1} at time t^{n+1} reads as follows:

1. Compute the new velocity \mathbf{u}^{n+1} by solving a convection–diffusion problem using the known field variables \mathbf{u}^n , p^n and ψ^n from the last time step

$$\frac{\rho^f}{\Delta t} \mathbf{u}^{n+1} + \rho^f (\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} - \mu \nabla^2 \mathbf{u}^{n+1} = \frac{\rho^f}{\Delta t} \mathbf{u}^n - \nabla (p^n + \psi^n) \quad \text{in } \Omega^f \quad (8a)$$

$$\mathbf{u}^{n+1} = \bar{\mathbf{u}}^{n+1} \quad \text{on } \Gamma_D^f \quad (8b)$$

$$\mu \partial_{\mathbf{n}} \mathbf{u}^{n+1} = p^n \mathbf{n} \quad \text{on } \Gamma_N^f. \quad (8c)$$

In this problem, each coordinate direction is decoupled, and, hence, each component of the velocity vector field can be determined independently.

2. Compute the pressure increment ψ^{n+1} by solving a Poisson problem involving the velocity field \mathbf{u}^{n+1} obtained in step 1

$$\nabla^2 \psi^{n+1} = \frac{\rho^f}{\Delta t} \nabla \cdot \mathbf{u}^{n+1} \quad \text{in } \Omega^f \quad (9a)$$

$$\partial_{\mathbf{n}} \psi^{n+1} = 0 \quad \text{on } \Gamma_D^f \quad (9b)$$

$$\psi^{n+1} = 0 \quad \text{on } \Gamma_N^f. \quad (9c)$$

Note that on the Dirichlet boundary Γ_D^f of the original boundary value problem (3), a homogeneous Neumann boundary condition is applied for ψ^{n+1} and vice versa.

3. Update the pressure field p^{n+1} according to the definition (7) by using \mathbf{u}^{n+1} from Step 1 and the pressure increment ψ^{n+1} from step 2.

There is only one velocity field in this algorithm. The intermediate and the end-of-step velocities familiar from conventional pressure-correction methods have been consolidated; see [23] for a step-by-step derivation. As a result of this elimination, the term $\nabla \psi^n$ appears on the right-hand side of (8a) and is associated with the field variables of the previous time step.

2.1.3. Weak form with weakly enforced Dirichlet boundary conditions. We consider the weak form of the Navier–Stokes equations in preparation for their finite element discretisation. The Dirichlet boundary conditions are enforced weakly because non-boundary-conforming meshes are used. To this end, we use the Nitsche method similar to our previous work [5]. In contrast to [5], here, we use a pressure-correction scheme and consider the weak form of the two associated subproblems (8) and (9).

The weak form of the convection–diffusion equation (8) appearing in step 1 of the pressure-correction method is obtained with trial functions from the Sobolev space $[\mathcal{H}^1(\Omega^f)]^d$, where d is the space dimension

$$\begin{aligned} \frac{\rho^f}{\Delta t}(\mathbf{u}^{n+1}, \mathbf{v}) + \rho^f ((\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1}, \mathbf{v}) + \mu (\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) - \mu \langle \partial_{\mathbf{n}} \mathbf{u}^{n+1}, \mathbf{v} \rangle_{\Gamma_D^f} = \\ \frac{\rho^f}{\Delta t} (\mathbf{u}^n, \mathbf{v}) - (\nabla(p^n + \psi^n), \mathbf{v}) \end{aligned} \quad (10)$$

with the inner products

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega^f} \mathbf{u} \cdot \mathbf{v} d\Omega \quad \text{and} \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\Gamma_D^f} = \int_{\Gamma_D^f} \mathbf{u} \cdot \mathbf{v} d\Gamma. \quad (11)$$

In (10), the integral over the Dirichlet boundary does not vanish because the test functions \mathbf{v} are not zero on the boundary. As will become clear, this facilitates the implementation of an immersed boundary finite element discretisation with non-boundary-conforming grids. Moreover, (10) does not contain an integral over the Neumann boundary due to the assumed do-nothing outflow condition (4).

The Dirichlet boundary condition (8b) is enforced weakly with the Nitsche method [25]. Specifically, the weak form (10) is augmented with the two terms

$$-\mu \langle \mathbf{u}^{n+1} - \bar{\mathbf{u}}^{n+1}, \partial_{\mathbf{n}} \mathbf{v} \rangle_{\Gamma_D^f} \quad \text{and} \quad + \frac{\gamma \mu}{h} \langle \mathbf{u}^{n+1} - \bar{\mathbf{u}}^{n+1}, \mathbf{v} \rangle_{\Gamma_D^f}, \quad (12)$$

where the second term represents the penalty contribution with the parameter $\gamma > 0$ and the grid spacing h . In contrast to conventional penalty methods, the parameter γ in the Nitsche method is only required for numerical stability and typically a small value is sufficient [5]. Finally, the weak form of the momentum equation with the Nitsche terms reads

$$\begin{aligned} \frac{\rho^f}{\Delta t} (\mathbf{u}^{n+1}, \mathbf{v}) + \rho^f ((\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1}, \mathbf{v}) + \mu (\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) \\ - \mu \langle \partial_{\mathbf{n}} \mathbf{u}^{n+1}, \mathbf{v} \rangle_{\Gamma_D^f} - \mu \langle \mathbf{u}^{n+1}, \partial_{\mathbf{n}} \mathbf{v} \rangle_{\Gamma_D^f} + \frac{\gamma \mu}{h} \langle \mathbf{u}^{n+1}, \mathbf{v} \rangle_{\Gamma_D^f} \\ = \frac{\rho^f}{\Delta t} (\mathbf{u}^n, \mathbf{v}) - (\nabla(p^n + \psi^n), \mathbf{v}) - \mu \langle \bar{\mathbf{u}}^{n+1}, \partial_{\mathbf{n}} \mathbf{v} \rangle_{\Gamma_D^f} + \frac{\gamma \mu}{h} \langle \bar{\mathbf{u}}^{n+1}, \mathbf{v} \rangle_{\Gamma_D^f}. \end{aligned} \quad (13)$$

Next, we consider the weak form of the Poisson equation (9) appearing in Step 2 of the pressure-correction method. Following the same procedure as for the convection–diffusion equation, we obtain

$$(\nabla \psi^{n+1}, \nabla q) - \langle \partial_{\mathbf{n}} \psi^{n+1}, q \rangle_{\Gamma_N^f} - \langle \partial_{\mathbf{n}} q, \psi^{n+1} \rangle_{\Gamma_N^f} + \frac{\gamma}{h} \langle \psi^{n+1}, q \rangle_{\Gamma_N^f} = -\frac{\rho^f}{\Delta t} (\nabla \cdot \mathbf{u}^n, q), \quad (14)$$

where q is a test function from the Sobolev space $\mathcal{H}^1(\Omega^f)$. Again, in (14) the Dirichlet boundary condition (9c) is weakly enforced with the Nitsche method so that the test functions need not be zero on the boundary Γ_N^f . Recall that the Neumann boundary Γ_N^f in the original problem (1) is the Dirichlet boundary for the auxiliary Poisson problem (9).

In Step 3 of the pressure-correction method, the computed auxiliary field ψ^{n+1} and velocity field \mathbf{u}^{n+1} are combined to update the pressure field p^{n+1} according to (7). Because (7) depends on

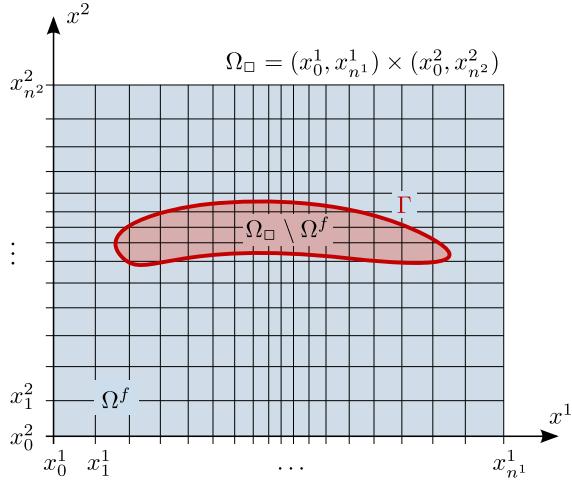


Figure 1. Domain of the Cartesian grid Ω_{\square} , fluid domain Ω^f and an embedded interface Γ .

the divergence of the velocity field that can be discontinuous, we use an \mathcal{L}_2 -projection to update the pressure field

$$(p^{n+1}, q) = (p^n + \psi^{n+1} - \mu \nabla \cdot \mathbf{u}^{n+1}, q), \quad (15)$$

where $q \in \mathcal{L}_2(\Omega^f)$ is the test function.

2.2. Immersed b-spline finite element discretisation

We use a logically Cartesian grid and the associated tensor-product b-spline basis functions for discretising the weak forms introduced in the previous section. Although, the size of the cells need not be uniform, the grid has to have the connectivity of a Cartesian grid. Figure 1 shows a typical set-up in two space dimensions, where the grid domain Ω_{\square} contains the fluid domain Ω^f and the immersed solid domain $\Omega_{\square} \setminus \Omega^f$. Because the Dirichlet boundary conditions are weakly enforced, it is not required that the boundaries of the fluid domain conform to the grid lines of the Cartesian grid. In the following, we introduce the isoparametric b-spline basis functions and discuss aspects of our implementation relating to the robustness and stability of the finite element discretisation.

2.2.1. Isoparametric b-spline basis functions and inf–sup stability. To begin with, we briefly review the definition and a few relevant properties of one-dimensional b-splines. For further details on b-splines, we refer to standard textbooks, for example, [26, 27]. We consider the uniform b-spline basis functions over a parameter space with equidistantly spaced knots $\xi_i = 0, 1, 2, 3, \dots$. The corresponding b-spline basis functions B_i^α of degree α are defined with the recursive averaging formula

$$\begin{aligned} B_i^0(\xi) &= \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_i^\alpha(\xi) &= \frac{\xi - \xi_i}{\alpha \Delta \xi} B_i^{\alpha-1}(\xi) + \frac{\xi_{i+\alpha+1} - \xi}{\alpha \Delta \xi} B_{i+1}^{\alpha-1}(\xi). \end{aligned} \quad (16)$$

Here, $\Delta \xi = 1$ is the distance between two consecutive knots ξ_i and ξ_{i+1} . A b-spline of degree α has a support size of $\alpha + 1$ knot intervals and within each knot interval, it consists of a polynomial of degree α . The polynomials over each interval are connected with $C^{\alpha-1}$ -continuity over the knots. As an example, in the top of Figure 2 the quadratic b-splines (i.e. $\alpha = 2$) are plotted. Notice in the indexing scheme implied by (16) and indicated in Figure 2, each b-spline is enumerated with respect to the knot at its support's lower boundary. All the basis functions computed with (16) are the shifted (or translated) versions of each other,

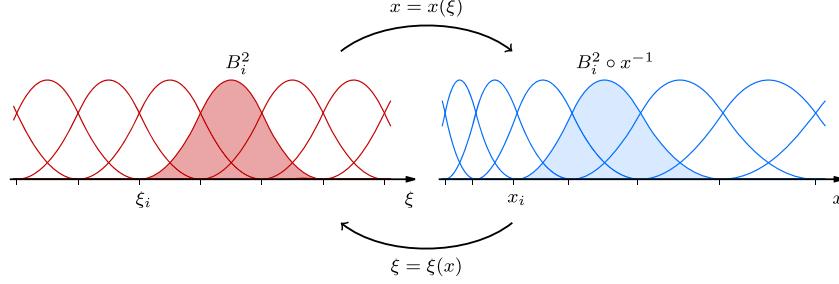


Figure 2. One-dimensional quadratic b-splines in the parameter space (left) and in the physical space (right). The isoparametric concept is used to establish the mapping between the two spaces.

$$B_i^\alpha(\xi) = B_0^\alpha(\xi - i \Delta \xi). \quad (17)$$

A unique property of b-splines is their refinability, which as proposed in [5] can be utilised to establish inf-sup stable finite element spaces. In order to introduce the refinability of b-splines, in addition to the knot sequence considered so far, we introduce the refined knot sequence $\tilde{\xi}_i = 0, 1/2, 1, 1/2, 2, \dots$. The b-splines \tilde{B}_i^α over this fine knot sequence can be obtained with (16). It is straightforward to show that the b-splines on the original coarse knot sequence can be represented as a linear combination of the b-splines on the fine knot sequence

$$B_i^\alpha(\xi) = \sum_k S_{ik}^\alpha \tilde{B}_{2i+k}^\alpha(\xi), \quad (18)$$

where S_{ik}^α is the subdivision matrix. For uniform b-splines, S_{ik}^α is a banded matrix and its entries depend only on the polynomial degree α [26, 27]. Figure 3 provides a graphic illustration of the two-scale relation in case of linear and quadratic b-splines ($\alpha = 1$ and $\alpha = 2$, respectively). Note in Figure 3 the indexing convention for the refined grid which is helpful to interpret (18).

With the one-dimensional b-splines at hand, the b-splines over a d -dimensional parameter space are defined as their tensor products

$$B_i^\alpha(\xi) = B_{i^1}^\alpha(\xi^1) \times \cdots \times B_{i^d}^\alpha(\xi^d), \quad (19)$$

where $i = (i^1, \dots, i^d)$ in bold type is a multi-index and B_{i^1}, \dots, B_{i^d} are the one-dimensional b-splines of degree α in the directions $\xi = (\xi^1, \dots, \xi^d)$, respectively. The introduced refinement

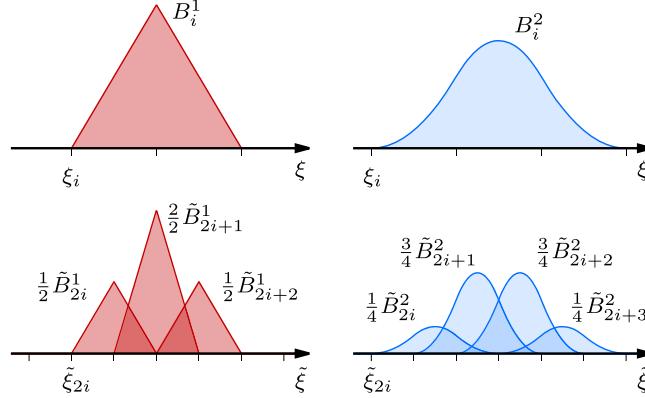


Figure 3. The two-scale relationship for linear and quadratic b-splines (left and right, respectively). In the top row are the b-splines defined on the coarse knot sequence and below are the (scaled) b-splines defined on the fine knot sequence. The sum of the scaled fine b-splines yields the coarse b-spline. The indicated scaling weights form the entries of the subdivision matrix.

relation (18) for one-dimensional b-splines carries over to the d -dimensional case as can be shown by introducing (18) into (19).

The weak forms appearing in the incremental pressure-correction method introduced in Section 2.1.3 are discretised with tensor-product b-spline basis functions. In accordance with the isoparametric concept, the trial and test functions and the geometry are approximated with the same b-spline basis functions. The geometry is represented with b-splines belonging to the refined grid

$$\mathbf{x} = \mathbf{x}(\xi) = \sum_i \tilde{B}_i^\alpha(\xi) \mathbf{x}_i , \quad (20)$$

where \mathbf{x}_i is the control point with the multi-index i . This representation induces a mapping of the b-spline basis functions from the parameter space to the physical space, that is, $B_i^\alpha(\mathbf{x}) = B_i^\alpha \circ \mathbf{x}^{-1}$.

As known, for incompressible problems the velocity and pressure approximations have to be chosen such that they are inf-sup stable (e.g. [28]). As we proposed in [5], this can be accomplished by interpolating the velocity field with the fine grid b-splines and the pressure related fields with the coarse grid b-splines

$$\mathbf{u}_h(\mathbf{x}) = \sum_i \tilde{B}_i^\alpha(\mathbf{x}) \mathbf{u}_i , \quad \psi_h(\mathbf{x}) = \sum_i B_i^\alpha(\mathbf{x}) \psi_i , \quad \text{and} \quad p_h(\mathbf{x}) = \sum_i B_i^\alpha(\mathbf{x}) p_i . \quad (21)$$

Recall that in the implemented pressure-correction method, the velocity field $\mathbf{u}_h(\mathbf{x})$ and the pressure increment field $\psi_h(\mathbf{x})$ are the primary variables. The pressure field $p_h(\mathbf{x})$ depends on $\mathbf{u}_h(\mathbf{x})$ and $\psi_h(\mathbf{x})$. For the sake of legibility, in (21) the time dependence of the coefficients has been ignored. Introducing the approximations (21) into the weak forms (13) and (14) yields, after numerical integration, the discrete finite element equations. The two-scale relation (18) greatly facilitates the integration and assembly of the element integrals. All the element matrices and vectors are first computed on the fine grid and are then projected to the correct level by multiplying with the subdivision matrix. This projection operation can be deferred until the assembly stage of the global system matrices and vectors. For a mathematical study of the convergence and stability of subdivision stabilised Stokes elements see [29].

2.2.2. Implicit geometry representation. On the fluid grid, we represent the domain boundaries implicitly with a signed distance function (or, in other terms, a level set). This is carried out despite there is a parametric representation of the structure domain in form of a finite element mesh is available, see Section 3. The aim of the switch from a parametric to an implicit representation is twofold: first, all geometric operations, such as the identification of cells, which lie on the boundary, can be performed much quicker; second, all pathological geometries and topologies, such as multiple crossings of a cell by the boundary, are automatically eliminated. This geometric filtering is particularly appealing for problems with large boundary deformations and propensity to topology changes to which thin beams and shells are susceptible. The advantages of this approach have been extensively demonstrated in our previous work [14, 30] in the context of fluid–structure interaction of compressible high-speed flows with thin shells and membranes.

The beam or shell meshes representing the fluid–structure interface are possibly open and hence cannot in general be represented by a signed distance function. Therefore, we first determine the unsigned distance function $\phi_u(\mathbf{x})$. Subsequently, the signed distance function $\phi_s(\mathbf{x})$ is computed by subtracting an offset from the unsigned distance function

$$\phi_s(\mathbf{x}) = \phi_u(\mathbf{x}) - \sqrt{d} (\alpha + 1) h = \phi_u(\mathbf{x}) - \delta , \quad (22)$$

where d is the space dimension, α is the polynomial degree of the b-spline, and h is the grid size in the vicinity of the considered point. This particular choice of the offset δ ensures that the supports of the b-splines located above and below the beam axis or shell midsurface do not intersect. This is important in order to enable discontinuous fluid tractions and C^0 -continuous fluid velocities across the thickness of the structure. As an illustrative example, Figure 4 shows an immersed fibre and some of its level set contour lines with $\phi_s = \text{const}$. Using the offset given by (22), the immersed fibre has a thickness of 2δ as seen by the fluid.

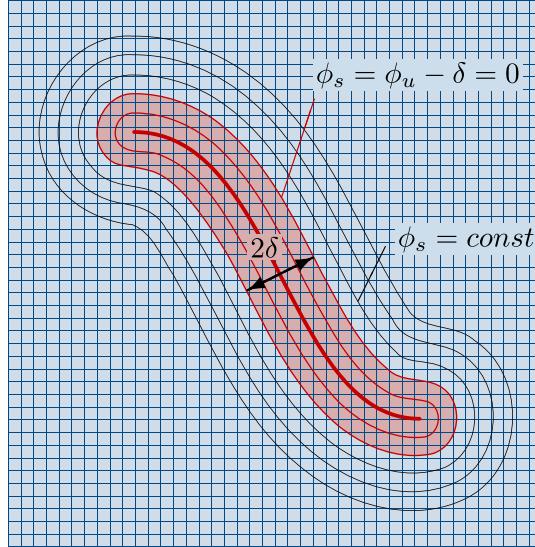


Figure 4. Immersed fibre with fluid boundary.

The computation of the distance function is best implemented as a preprocessing step prior to the solution of the fluid equations. A brute-force approach would require that the distance of each fluid grid point to each surface element is determined and the minimum value retained. This leads to an algorithm with quadratic complexity and hence is not practical. Following an approach similar to collision or contact search, we compute the distance function in two steps. First, the grid points lying within the bounding box of a surface element are found. Subsequently, for each grid point in the bounding box the distance to the surface is determined. There are efficient algorithms with logarithmic complexity that can be used in the first step of this algorithm. In [31] a possible C++ implementation is sketched. The computational cost is further reduced by only computing the distance for grid points within a slightly expanded bounding box of surface elements. This is sufficient because the distance function only in the vicinity of the structure is needed. After determining the distance values $\phi_{u,i} = \phi_u(x_i)$ at the fluid grid points, the distance function is interpolated with the b-spline basis functions,

$$\phi_u(\mathbf{x}) = \sum_i \tilde{B}_i^\alpha(\mathbf{x}) \phi_{u,i}. \quad (23)$$

As a by-product of the distance computation, for every fluid grid point \mathbf{x}_i , the closest surface point \mathbf{x}_i^* and the index of the corresponding surface element are obtained. This information is used to store the prescribed surface velocity $\bar{\mathbf{u}}$ on the fluid grid nodes, that is,

$$\bar{\mathbf{u}}(\mathbf{x}_i) \approx \bar{\mathbf{u}}_i = \bar{\mathbf{u}}(\mathbf{x}_i^*). \quad (24)$$

Later, during the evaluation of the surface integrals in the weak forms, the prescribed surface velocity is recovered by

$$\bar{\mathbf{u}}(\mathbf{x}_i) = \sum_i \tilde{B}_i^\alpha(\mathbf{x}_i^*) \bar{\mathbf{u}}_i. \quad (25)$$

In a similar fashion, in case of moving boundaries, the pressure field within the domain is extended into the inactive domain $\Omega_\square \setminus \Omega^f$, which amounts to a constant extrapolation of the pressure field. This greatly improves the solution quality when the immersed object moves by large amounts and uncovers several grid points within one time step. Without such an extrapolation, the pressure in the inactive domain is undetermined.

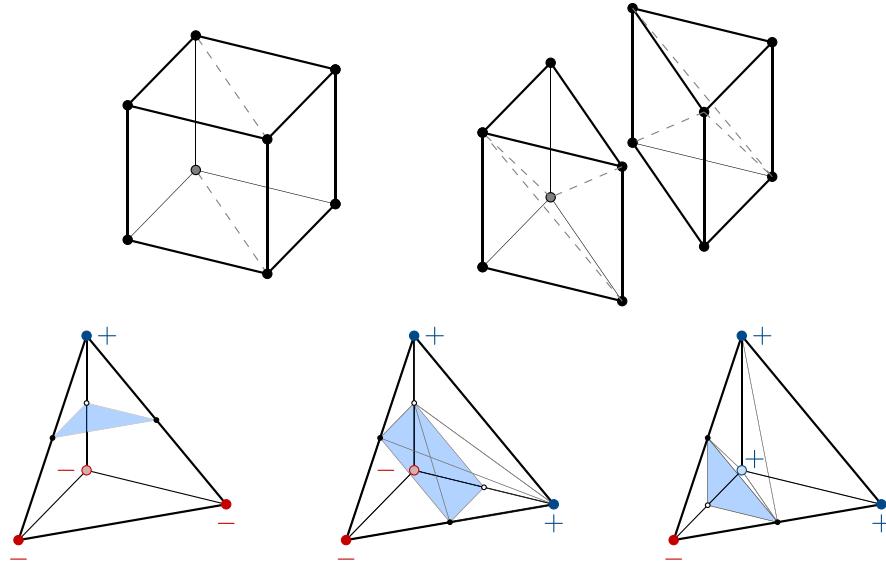


Figure 5. Decomposition of a cut cell into tetrahedra. A cell is first decomposed into two prisms and subsequently each prism is decomposed into three tetrahedra (top row). There are three canonical ways how one of the tetrahedra can be cut by a surface (bottom row).

2.2.3. Cut-cell integration. The location of the boundary is needed for the evaluation of the element integrals appearing in the weak form of the fluid equations. The cells which lie entirely inside the fluid domain are integrated using a tensor-product Gauss quadrature. The cut cells which partly lie in the fluid domain are first triangulated prior to integration. To this end, the signed distance values $\phi_s(\mathbf{x}_i)$ stored at the Cartesian grid points are linearly interpolated for determining the position of the boundary within each cell. Subsequently, the resulting cut cells are decomposed into triangles (in 2D) or tetrahedra (in 3D) and then integrated with a standard Gauss quadrature.

Whereas the triangulation of the cut cells in two dimensions is straightforward, special care needs to be taken in three dimensions. We use the marching tetrahedra algorithm similar to the one presented in [32]. Any given cell that contains a portion of the implicit surface is initially decomposed into six tetrahedra as shown on the top row of Figure 5. Conceptually, the cell is first decomposed into two identical prisms, and then each prism is decomposed into three tetrahedra [28]. This decomposition of the cell into six tetrahedra ensures that the diagonal cuts on the cell's surface coincide with the diagonal cuts of the neighbouring cells. An alternative decomposition of cells into five tetrahedra leads to gaps between two neighbouring cells. In the next step, each of the six tetrahedra is considered individually. From Figure 5, one can deduce that there are three canonical ways that a surface can transverse a tetrahedron. In each case, the volume on the positive side (i.e. the interior of the fluid domain) is recovered by another tetrahedralisation from which surface triangles are inherited in order to recover the surface.

As an aside, a limitation of the implicit geometry representation is that the signed distance function allows only for a piecewise linear reconstruction of the boundary. This may lead to suboptimal convergence rates for the numerical solution with higher order finite elements. However, depending on the application, the robustness provided by the implicit geometry representation may outweigh its disadvantages.

2.2.4. Cut-cell stabilisation. In the employed b-spline finite element technique, similar to conventional finite elements, each degree of freedom is paired to a specific basis function. The b-spline basis functions intersecting the domain boundaries can have a small physically active support size. The magnitude of the corresponding entries in the system matrix are proportional to the size of the physically active support size, which can become arbitrarily small. This compromises the conditioning of the system matrix and, hence, the robustness of the overall approach. The basis functions

with a support entirely outside the physical domain are discarded from the outset and do not cause any problems.

One compelling solution for sidestepping the ill-conditioning problem is to eliminate the basis functions with a small physically active support size from the global system of equations. Formally, the b-splines and their coefficients in the interpolation equation (21) can be split into two disjoint sets so that, for instance, the velocity interpolation reads

$$\mathbf{u}_h(\mathbf{x}) = \sum_i \tilde{B}_i^\alpha(\mathbf{x}) \mathbf{u}_i = \sum_{j \in \mathbb{I}} \tilde{B}_j^\alpha(\mathbf{x}) \mathbf{u}_j + \sum_{k \in \mathbb{E}} \tilde{B}_k^\alpha(\mathbf{x}) \mathbf{u}_k. \quad (26)$$

Here, the index set \mathbb{E} collects the *exterior b-splines* with a small physically active support size and the index set \mathbb{I} collects the *interior b-splines*. Simply discarding the exterior b-splines \tilde{B}_k^α and their coefficients \mathbf{u}_k , with $k \in \mathbb{E}$, inhibits the polynomial reproduction property of the b-spline basis. A better approach is to determine the coefficients of the exterior b-splines by way of extrapolation of the solution from inside the physical domain. This means that in (26), the coefficients \mathbf{u}_k with $k \in \mathbb{E}$ are expressed in terms of the coefficients \mathbf{u}_j with $j \in \mathbb{I}$. This approach goes back to Höllig *et al.* [33] and resembles the construction of one-sided differences used in finite difference methods, see, for example, [34]. Formally, the extrapolation allows us to express the coefficients of exterior b-splines as a linear combination of interior b-spline coefficients from the subset $\mathbb{I}(k) \subset \mathbb{I}$

$$\mathbf{u}_k = \sum_{j \in \mathbb{I}(k)} E_{k,j} \mathbf{u}_j. \quad (27)$$

The coefficients $E_{k,j}$ are determined by constructing tensor-product Lagrange polynomials in the parameter space. The Lagrange polynomials must have the same degree as the b-spline basis functions, so that their polynomial reproduction property is retained. Hence, the $(\alpha + 1)^d$ Lagrange polynomials supported at the knots ξ_j with $j \in \mathbb{I}(k)$ evaluated at the knot ξ_k yield the entries $E_{k,j}$. Therefore, the subset $\mathbb{I}(k)$ has the dimension $(\alpha + 1)^d$, its elements must all be from the set \mathbb{I} of interior b-splines, and their location is chosen as close as possible to the considered exterior spline with index k . Insertion of expression (27) into (26) and regrouping the sum according to the coefficients \mathbf{u}_j lead to a modified approximation space, which depend only on interior b-splines. In Figure 6 a graphic illustration of the extension process in the one-dimensional case is shown.

In terms of computer implementation, the extension procedure can be deferred until the assembly stage of the element matrices and vectors. This means that the element matrices and vectors pertaining to all cells are first computed using the standard b-spline basis. The cut cells are integrated using the approach outlined in Section 2.2.3. Subsequently, the element matrices and vectors are, prior to the assembly to global matrices and vectors, multiplied with extrapolation matrices derived from (27). After the global system of equations is solved, the relation (27) is used to obtain the coefficients of \mathbf{u}_k of the exterior splines. For further details on implementation we refer to [5].

3. ISOGEOMETRIC BEAM AND SHELL FINITE ELEMENTS

In this section, the governing equation and the discretisation of a planar beam with Euler–Bernoulli kinematics is introduced. For the corresponding derivation of a Kirchhoff–Love type shell finite element we refer to Cirak *et al.* [12, 13]. Although we use also shells in our computational examples, their derivation will not be repeated as it is inconsequential for the discussions on fluid–structure interaction.

In the Euler–Bernoulli beam model, the transverse shear deformations are not considered and, hence, it is only suitable for thin beams. Transverse shear deformations may become important with increasing beam thickness. In order to compute thick as well as thin beams the Euler–Bernoulli model can be extended as proposed in Long *et al.* [35].

3.1. Governing equations

The beam occupies in its reference configuration the domain $\Omega_0^s \times (-\frac{t}{2}, \frac{t}{2})$ with the beam axis Ω_0^s and the thickness t (Figure 7). It is assumed that the beam is embedded in the coordinate plane

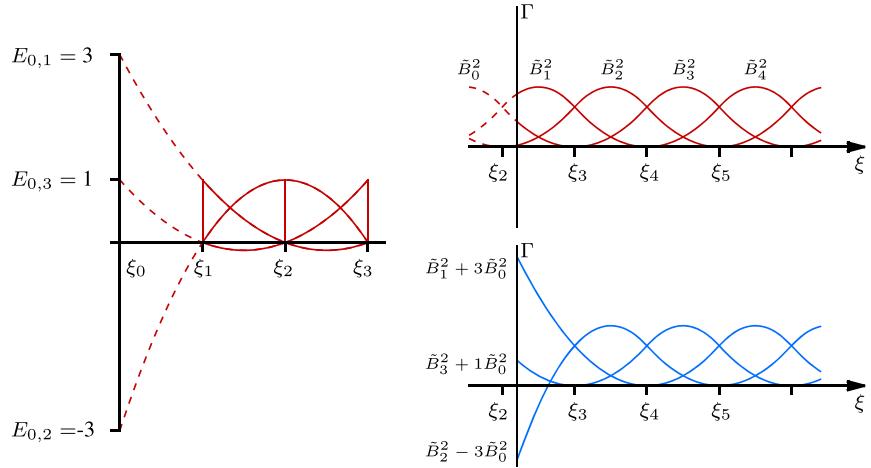


Figure 6. One-dimensional example illustrating the cut-cell stabilisation. On the top right, the b-spline \tilde{B}_0^2 intersects the domain boundary Γ , and its physically active support size is too small (less than one knot interval). Hence, the coefficient u_0 is to be eliminated as it will lead to an ill-conditioned system matrix. The index set of exterior b-splines is $\mathbb{E} = \{0\}$, and the index set of interior b-splines is $\mathbb{I} = \{1, 2, 3, \dots\}$. As shown in the left figure, the quadratic Lagrange polynomials are constructed using the knots from the index set $\mathbb{I}(0) = \{1, 2, 3\}$. The extrapolation equation for the coefficient u_0 can be deduced from the left figure, that is, $u_0 = \sum_j E_{0,j} u_j = 3u_1 - 3u_2 + 1u_3$. The corresponding stabilised basis is shown on bottom right.

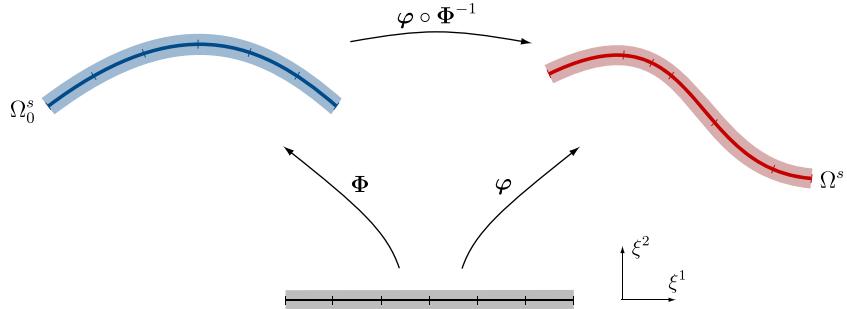


Figure 7. The beam in the reference and deformed configurations with the chosen parameterisation.

spanned by the orthonormal basis vectors e_1 and e_2 , and any deformations are constrained to this plane. The position vector of a material point in the reference configuration of the beam is parameterised with the convective coordinates $\{\xi^1, \xi^2\}$ as

$$\Phi(\xi^1, \xi^2) = X(\xi^1) + \xi^2 N(\xi^1) \quad \text{with } -\frac{t}{2} \leq \xi^2 \leq \frac{t}{2}, \quad (28)$$

where X is the position vector of the beam axis, and N is its unit normal. In its deformed configuration, the beam occupies the domain $\Omega^s \times (-\frac{t}{2}, \frac{t}{2})$. The position vector of a material point with the coordinates $\{\xi^1, \xi^2\}$ in the deformed configuration is assumed to be

$$\varphi(\xi^1, \xi^2) = x(\xi^1) + \xi^2 n(\xi^1) \quad \text{with } -\frac{t}{2} \leq \xi^2 \leq \frac{t}{2}, \quad (29)$$

where x is the position vector of the deformed beam axis, and n is its unit normal. With these definitions, the deformation of the beam can be expressed as $\varphi \circ \Phi^{-1}$.

The kinematics of the beam is best described with a convective covariant coordinate frame with the basis vectors

$$G_1 = \frac{\partial \Phi}{\partial \xi^1} = \frac{dX}{d\xi^1} + \xi^2 \frac{dN}{d\xi^1} = A_1 + \xi^2 \frac{dN}{d\xi^1} \quad \text{and} \quad G_2 = \frac{\partial \Phi}{\partial \xi^2} = N. \quad (30)$$

The corresponding basis vectors of the deformed configuration are the following

$$\mathbf{g}_1 = \frac{\partial \boldsymbol{\varphi}}{\partial \xi^1} = \frac{d\mathbf{x}}{d\xi^1} + \xi^2 \frac{d\mathbf{n}}{d\xi^1} = \mathbf{a}_1 + \xi^2 \frac{d\mathbf{n}}{d\xi^1} \quad \text{and} \quad \mathbf{g}_2 = \frac{\partial \boldsymbol{\varphi}}{\partial \xi^2} = \mathbf{n}. \quad (31)$$

The unit normal to the beam axis in both configurations is given by

$$\mathbf{N} = \mathbf{e}_3 \times \frac{\mathbf{A}_1}{|\mathbf{A}_1|} \quad \text{and} \quad \mathbf{n} = \mathbf{e}_3 \times \frac{\mathbf{a}_1}{|\mathbf{a}_1|}, \quad (32)$$

where \mathbf{e}_3 is the out-of-plane basis vector, that is, $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$.

The deformation gradient of the beam according to [13, 36] reads

$$\mathbf{F} = \frac{\partial \boldsymbol{\varphi}}{\partial \Phi} = \sum_i \mathbf{g}_i \otimes \mathbf{G}^i. \quad (33)$$

Here and in the following, the vectors with a superscript denote contravariant vectors and are defined according to

$$\mathbf{G}_i \cdot \mathbf{G}^j = \delta_i^j \quad (34)$$

with the Kronecker delta δ_i^j .

The Green–Lagrange strain tensor of the beam can be expressed as

$$\mathbf{E}(\mathbf{x}) = \boldsymbol{\alpha}(\mathbf{x}) + \xi^2 \boldsymbol{\beta}(\mathbf{x}) \quad (35)$$

with

$$\boldsymbol{\alpha}(\mathbf{x}) = \frac{1}{2} (\mathbf{a}_1 \cdot \mathbf{a}_1 - \mathbf{A}_1 \cdot \mathbf{A}_1) \mathbf{A}^1 \otimes \mathbf{A}^1 \quad (36a)$$

$$\boldsymbol{\beta}(\mathbf{x}) = \left(\mathbf{a}_1 \cdot \frac{d\mathbf{n}}{d\xi^1} - \mathbf{A}_1 \cdot \frac{d\mathbf{N}}{d\xi^1} \right) \mathbf{A}^1 \otimes \mathbf{A}^1. \quad (36b)$$

The only nonzero component of the membrane strain tensor, $\alpha_{11} = \frac{1}{2} (\mathbf{a}_1 \cdot \mathbf{a}_1 - \mathbf{A}_1 \cdot \mathbf{A}_1)$, measures the straining of the beam axis. The components α_{12} measuring the transverse shear and α_{22} measuring the stretching of the beam cross-section are zero. The only nonzero component of the bending strain tensor, $\beta_{11} = (\mathbf{a}_1 \cdot d\mathbf{n}/d\xi^1 - \mathbf{A}_1 \cdot d\mathbf{N}/d\xi^1)$, measures the change of curvature of the beam axis in the \mathbf{e}_1 - \mathbf{e}_2 plane.

Next, we outline the derivation of the weak form of the beam equilibrium equations starting from the corresponding equations for a 2D solid

$$\begin{aligned} & \int_{\Omega_0^s} \int_{-t/2}^{t/2} \rho^s \ddot{\boldsymbol{\varphi}}(\mathbf{x}) \cdot \boldsymbol{\varphi}(\mathbf{y}) \bar{J} d\xi^2 d\Omega_0 + \int_{\Omega_0^s} \int_{-t/2}^{t/2} \mathbf{S}(\mathbf{x}) : \mathbf{E}(\mathbf{y}) \bar{J} d\xi^2 d\Omega_0 \\ &= \int_{\Omega_0^s} [\bar{\mathbf{T}} \cdot \boldsymbol{\varphi}(\mathbf{y}) \bar{J}]_{-t/2}^{t/2} d\Omega_0, \end{aligned} \quad (37)$$

where ρ^s is the solid mass density; \mathbf{y} is a test function, or virtual displacement of the beam axis, that satisfies the Dirichlet boundary conditions; \mathbf{S} is the second Piola–Kirchhoff stress; and $\bar{\mathbf{T}}$ are the prescribed tractions on the top and bottom surfaces of the beam. The Jacobian \bar{J} in (37) is related to the integral across the thickness and takes into account the curvature of the beam,

$$\bar{J} = \frac{|\mathbf{G}_1|}{|\mathbf{A}_1|}. \quad (38)$$

The second term in (37) in combination with the Green–Lagrange strain tensor (35) motivates the definition of the stress resultants

$$\widetilde{\mathbf{N}} = \int_{-t/2}^{t/2} \mathbf{S} \bar{J} d\xi^2 \quad \text{and} \quad \widetilde{\mathbf{M}} = \int_{-t/2}^{t/2} \mathbf{S} \xi^2 \bar{J} d\xi^2. \quad (39)$$

Furthermore, by invoking the thin beam assumption, the weak form (37) can be rewritten according to [12, 13] as

$$\int_{\Omega_0^s} \rho^s A \ddot{\mathbf{x}} \cdot \mathbf{y} d\Omega_0 + \int_{\Omega_0^s} \widetilde{\mathbf{N}}(\mathbf{x}) : \boldsymbol{\alpha}(\mathbf{y}) + \widetilde{\mathbf{M}}(\mathbf{x}) : \boldsymbol{\beta}(\mathbf{y}) d\Omega_0 = \int_{\Omega_0^s} \mathbf{y} \cdot [\overline{\mathbf{T}}]_-^+ d\Omega_0, \quad (40)$$

where A is the area of the beam cross-section, and in the last term, $\overline{\mathbf{T}}^+$ and $\overline{\mathbf{T}}^-$ are the prescribed tractions above and below the beam axis, respectively.

Notice that the prescribed surface tractions $\overline{\mathbf{T}} = \mathbf{F} \mathbf{S} \mathbf{N}$ in (40) are with respect to the reference configuration. However, in fluid–structure interaction the fluid tractions are given as Cauchy tractions $\bar{\mathbf{t}} = \boldsymbol{\sigma} \mathbf{n}$ (i.e. in the deformed configuration). The two can be related to each other by the following sequence of transformations (see, e.g., [37]):

$$\begin{aligned} \int_{\Omega_0^s} \mathbf{y} \cdot [\overline{\mathbf{T}}]_-^+ d\Omega_0 &= \int_{\Omega_0^s} \mathbf{y} \cdot [\mathbf{F} \mathbf{S} \mathbf{N}]_-^+ d\Omega_0 = \int_{\Omega_0^s} \mathbf{y} \cdot \left[\frac{1}{J} \mathbf{F} \mathbf{S} \mathbf{F}^\top \mathbf{n} \right]_-^+ d\Omega \\ &= \int_{\Omega_0^s} \mathbf{y} \cdot [\bar{\mathbf{t}}]_-^+ \overline{J} d\Omega_0 \end{aligned} \quad (41)$$

under the assumption of a small thickness. We used here Nanson's relation $\mathbf{n} d\Omega = J \mathbf{F}^{-\top} \mathbf{N} d\Omega_0$ with $J = \det(\mathbf{F})$ and the surface Jacobian

$$\overline{J} = \frac{|\mathbf{a}_1|}{|A_1|}. \quad (42)$$

Finally, a constitutive model is needed to complete the governing equations. We can write for the assumed St. Venant–Kirchhoff material model (see, e.g., [36]):

$$\widetilde{\mathbf{N}} = \widetilde{\mathbf{N}}^{11} \mathbf{A}_1 \otimes \mathbf{A}_1 = E \mathbf{A} \mathbf{H} : \boldsymbol{\alpha} \quad (43a)$$

$$\widetilde{\mathbf{M}} = \widetilde{\mathbf{M}}^{11} \mathbf{A}_1 \otimes \mathbf{A}_1 = \frac{EA t^2}{12} \mathbf{H} : \boldsymbol{\beta} \quad (43b)$$

with the cross-section A , Young's modulus E and the geometric tensor

$$\mathbf{H} = (\mathbf{A}^1 \cdot \mathbf{A}^1) (\mathbf{A}^1 \cdot \mathbf{A}^1) \mathbf{A}_1 \otimes \mathbf{A}_1 \otimes \mathbf{A}_1 \otimes \mathbf{A}_1. \quad (44)$$

The St. Venant–Kirchhoff material model is only suitable for modelling beams with large displacements but small strains. For modelling beams and shells with finite deformations and nonlinear material models we refer to [12].

3.2. Finite element discretisation

The weak form (40) depends on the deformed coordinates of the beam axis and the corresponding test functions. Smooth (formally, \mathcal{H}^2 -conforming) basis functions are necessary for discretising (40) as it depends on the second order derivatives of the reference and deformed beam axes. In the present approach, we discretise beams with b-splines and shells with basis functions derived from subdivision surfaces [13]. In contrast to tensor-product b-splines discussed in Section 2.2, subdivision basis functions are able to represent surfaces with arbitrary topology. On structured meshes subdivision basis functions reduce to tensor-product b-splines or box splines.

Only b-splines with polynomial degree $\alpha \geq 2$ have the requisite smoothness properties for discretising the weak form (40). Because the most common subdivision surfaces lead to basis functions of degree $\alpha = 3$, we discretise beams with b-splines of same degree. Without further elaboration, the b-spline interpolation leads to equations of the form

$$\mathbf{X}_h = \sum_i N_i \mathbf{X}_i \quad \text{and} \quad \mathbf{x}_h = \sum_i N_i \mathbf{x}_i, \quad (45)$$

where N_i are either the cubic b-spline or subdivision basis functions, the coefficients X_i represent the known nodal coordinates in the reference configuration, and \mathbf{x}_i represent the unknown nodal coordinates in the deformed configuration. Introducing the interpolation (45) into the weak form and evaluating the integrals with numerical integration yields the semi-discrete systems of equations

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f}^{\text{int}}(\mathbf{x}) = \mathbf{f}^{\text{ext}}, \quad (46)$$

where \mathbf{M} is the mass matrix, $\mathbf{f}^{\text{int}}(\mathbf{x})$ is the internal force array, and \mathbf{f}^{ext} is the external force array. The arrays $\ddot{\mathbf{x}}$ and \mathbf{x} represent the assembled nodal acceleration vector and deformed coordinates, respectively. The semi-discrete equations are first discretised with the implicit Newmark scheme and subsequently solved with the Newton–Raphson scheme.

4. FLUID–STRUCTURE COUPLING

In this section, we consider the solution of the coupled fluid–structure interaction problem. In the immersed boundary approach introduced in Section 2, the incompressible fluid flow is described in the entire domain using an Eulerian frame of reference. In contrast, the beam or shell equations introduced in Section 3 are expressed using a Lagrangian frame of reference and may undergo large deformations. In addition to the governing equations of the fluid and the structure, at the common interface Γ_I , the following kinematic and traction continuity conditions have to be satisfied:

$$\mathbf{u}|_{\text{fluid}} = \dot{\mathbf{x}}|_{\text{struct.}} \quad \text{and} \quad \boldsymbol{\sigma}|_{\text{fluid}} \mathbf{n} = \boldsymbol{\sigma}|_{\text{struct.}} \mathbf{n} \quad \text{on } \Gamma_I. \quad (47)$$

The weak form of the fully coupled fluid–structure interaction problem is obtained by adding the weak forms of the fluid (1) and structure (40) and enforcing the continuity conditions (47) with the Nitsche method [6, 7]. The related derivations are given in Appendix B. In the following, we focus on the solution of the fully coupled fluid–structure interaction problem with a partitioned approach. In each time step this is conceptually similar to solving the discrete systems of equations resulting from the fluid–structure interaction problem with a block Gauss–Seidel method. The iterations terminate when the discretised fluid and structure equations and the interface conditions are satisfied up to a desired convergence tolerance.

4.1. Review of the Dirichlet–Robin partitioned approach

The specific partitioning technique we implemented is the Dirichlet–Robin partitioning. In each iteration step, the fluid is solved with a Dirichlet interface condition and the solid with a Robin interface condition. The Dirichlet–Robin partitioning is a natural consequence of enforcing the interface conditions with the Nitsche method and partitioning of the resulting coupled fluid–structure interaction problem (see Appendix B and [7]). According to this, the interface boundary condition for the structure subproblem is a Robin condition

$$\boldsymbol{\sigma}|_{\text{struct.}} \mathbf{n} - \frac{\gamma\mu}{h} \dot{\mathbf{x}}_\Gamma = \boldsymbol{\sigma}|_{\text{fluid}} \mathbf{n} - \frac{\gamma\mu}{h} \mathbf{u} \quad \text{on } \Gamma_I, \quad (48)$$

where γ is the Nitsche parameter. The fluid viscosity μ and the fluid cell size h are introduced for dimensional consistency. The corresponding interface boundary condition for the fluid subproblem is a Dirichlet boundary condition

$$\mathbf{u}|_{\text{fluid}} = \dot{\mathbf{x}}|_{\text{struct.}} \quad \text{on } \Gamma_I. \quad (49)$$

In Figure 8, the partitioning of the fluid–structure interaction problem into fluid and structure subproblems and the information flow between the two are illustrated. As has been shown in [7], Dirichlet–Robin coupling alleviates the extensively reported convergence problems of the Dirichlet–Neumann schemes (see, e.g., [17, 18]).

The solution of the coupled fluid–structure interaction problem using the Dirichlet–Robin partitioning requires only minor modifications to the fluid and structure solvers introduced in Sections 2 and 3. In the fluid subproblem, the Dirichlet boundary condition (49) is applied with the Nitsche method as introduced in Section 2.1.3. We emphasise that the Nitsche parameter appearing in (13)

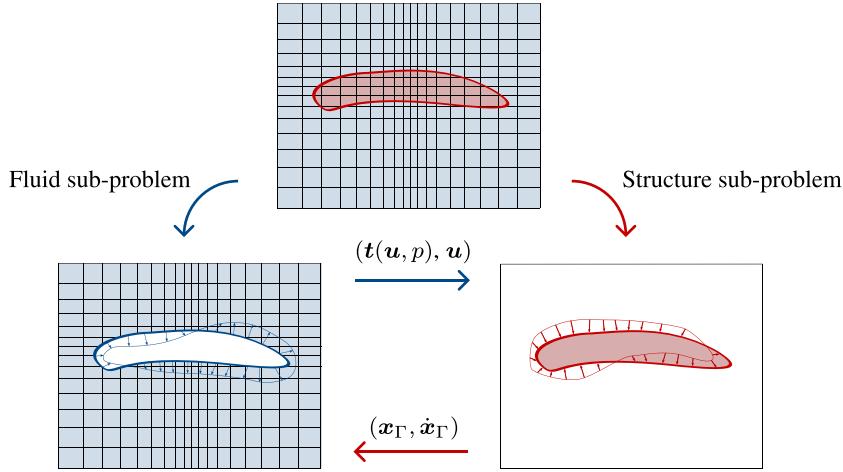


Figure 8. Fluid–structure coupling: partitioning of the problem and the data transfer.

for enforcing the Dirichlet boundary condition is the same as in Robin boundary condition (48). In the structure subproblem, the incorporation of the Robin boundary condition into the structure weak form (40) leads to a discrete dynamic equilibrium equation of the form

$$\mathbf{M}\ddot{\mathbf{x}} + \frac{\gamma\mu}{h}\mathbf{M}\dot{\mathbf{x}} + \mathbf{f}^{\text{int}}(\mathbf{x}) = \mathbf{f}^{\text{fluid}} + \frac{\gamma\mu}{h}\mathbf{M}\mathbf{u}. \quad (50)$$

The right-hand side terms are determined from the previous fluid solution. The computation of the nodal forces $\mathbf{f}^{\text{fluid}}$ is described in Section 4.3. The fluid nodal velocities on the right-hand side are the same as the structure velocities from the previous iteration step. Hence, there is no need to transfer them from the fluid to the structure.

4.2. Dirichlet–Robin coupling algorithm

The precise sequence of the fluid and structure solutions and transfer of interface data will be given next. To this end, it is useful to introduce several abstract operators.

- The fluid operator $\hat{\mathbf{F}}$ maps the Dirichlet boundary Γ with the prescribed velocities $\bar{\mathbf{u}}$ to a new fluid solution state

$$\hat{\mathbf{F}}: (\Gamma, \bar{\mathbf{u}}) \mapsto (\mathbf{u}, p). \quad (51)$$

- The structure operator $\hat{\mathbf{S}}$ maps nodal forces \mathbf{f}_j to a new structure solution state

$$\hat{\mathbf{S}}: (\{f_j\}) \mapsto (x_\Gamma, \dot{x}_\Gamma). \quad (52)$$

- The transfer operator $\hat{\mathbf{T}}$ maps the fluid solution state to structural nodal forces, and the reverse transfer operator $\hat{\mathbf{T}}^\dagger$ maps the structure solution state to the fluid Dirichlet boundary

$$\hat{\mathbf{T}}: (\mathbf{u}, p) \mapsto \{f_j\}, \quad \hat{\mathbf{T}}^\dagger: (x_\Gamma, \dot{x}_\Gamma) \mapsto (\Gamma, \bar{\mathbf{u}}). \quad (53)$$

The realisation of these two transfer operators will be discussed in the next section.

Now, we consider the solution of the fluid–structure interaction problem at a specific time step t^n . To simplify the notation, the upper index n referring to the time step will be dropped, and a new upper index (k) will be introduced as an iteration counter. With these definitions at hand, the Dirichlet–Robin coupling algorithm reads as follows:

1. For $(k) = (0)$, initialise the structure state by extrapolating from the previous time step t^{n-1}

$$\mathbf{x}_\Gamma^{(0)} = \mathbf{x}_\Gamma^{n-1} + \Delta t \dot{\mathbf{x}}_\Gamma^{n-1} \quad \text{and} \quad \dot{\mathbf{x}}_\Gamma^{(0)} = \dot{\mathbf{x}}_\Gamma^{n-1}. \quad (54)$$

2. Transfer the structure state to the fluid

$$\left(\Gamma^{(k)}, \bar{\mathbf{u}}^{(k)} \right) = \hat{\mathbf{T}}^\dagger \left(\mathbf{x}_\Gamma^{(k-1)}, \dot{\mathbf{x}}_\Gamma^{(k-1)} \right). \quad (55)$$

3. Solve the fluid problem

$$\left(\mathbf{u}^{(k)}, p^{(k)} \right) = \hat{\mathbf{F}} \left(\Gamma^{(k)}, \bar{\mathbf{u}}^{(k)} \right). \quad (56)$$

4. Compute the nodal forces applied to the structure

$$\left(\left\{ f_j^{(k)} \right\} \right) = \hat{\mathbf{T}} \left(\mathbf{u}^{(k)}, p^{(k)} \right). \quad (57)$$

5. Solve the structure problem

$$\left(\mathbf{x}_\Gamma^{(k)}, \dot{\mathbf{x}}_\Gamma^{(k)} \right) = \hat{\mathbf{S}} \left(\left\{ f_j^{(k)} \right\} \right). \quad (58)$$

6. If not converged, set $(k + 1) \rightarrow (k)$ and go to step 2.

As a convergence criterion, we use the deformed position of the structure, that is,

$$\| \mathbf{x}_\Gamma^{(k)} - \mathbf{x}_\Gamma^{(k-1)} \| < \varepsilon. \quad (59)$$

4.3. Conservative load transfer

Most, if not all, fixed-grid fluid–structure interaction approaches lead to non-matching interfaces between the fluid and the structure. Therefore, the transfer of data between the fluid and structure solvers requires some care, so that energy is conserved during the transfer.

First, we specify the operator $\hat{\mathbf{T}}^\dagger$, introduced in (53), which maps the structure solution to the fluid Dirichlet boundary. It is implemented as a consequent extension of the approach introduced in 2.2.2 for describing and enforcing boundary conditions using an implicit geometry representation. As an illustration, in Figure 9, a fluid cell with an immersed control polygon Γ_I^s representing the structure is shown. In Figure 9 (left), the distance of a cell vertex i to Γ_I^s is denoted with $\phi(\mathbf{x}_i)$ and the corresponding closest point with \mathbf{x}_i^* . By linearly interpolating the nodal distances $\phi(\mathbf{x}_i)$ on a fluid cell, it is possible to reconstruct a surrogate interface Γ_I^f , see Figure 9 (right). The prescribed velocities for this surrogate interface are obtained by first assigning each fluid vertex the velocity of the closest point on the structure control polygon, that is, $\mathbf{u}_i = \dot{\mathbf{x}}(\mathbf{x}_i^*)$. Subsequently, the surrogate interface velocities are obtained by interpolation from the fluid vertex velocities.

Next, we consider the operator $\hat{\mathbf{T}}$, introduced in (53), which maps the fluid solution to nodal forces of the structure. This operator is implemented, as proposed by Farhat *et al.* [20], such that work is

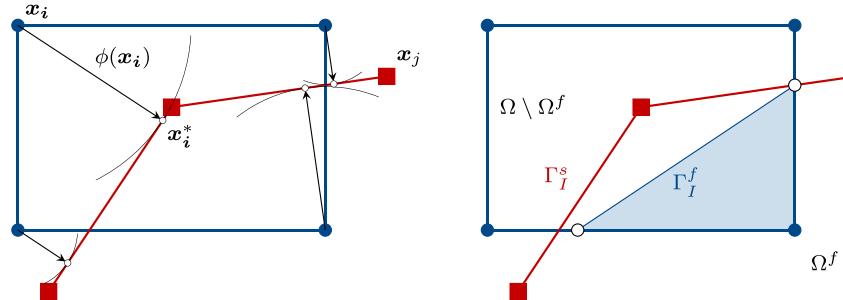


Figure 9. A fluid cut-cell with an immersed control polygon Γ_I^s representing the structure (left) and the surrogate interface Γ_I^f on the fluid grid (right). The scalar $\phi(\mathbf{x}_i)$ denotes the distance of grid point \mathbf{x}_i to the control polygon, and $\mathbf{x}_i^* \in \Gamma_I^s$ is the corresponding closest point.

conserved during the transfer. The fluid tractions are first integrated on the surrogate interface Γ_I^f and then consistently transferred to the structure. To begin with, we express with the mapping $\hat{\mathbf{T}}^\dagger$, the coefficient \mathbf{v}_i of the fluid test function at the cell vertex i as a linear combination of the structure test function coefficients y_j , that is,

$$\mathbf{v}_i = \sum_j N_j(\mathbf{x}_i^*) y_j. \quad (60)$$

Recall that we denote structure basis functions with N and fluid basis functions with B . Next, we require that the virtual work performed by the nodal fluid and structure forces is identical

$$\sum_i \left[\int_{\Gamma_I^f} (\sigma|_{\text{fluid}} \mathbf{n}) B_i d\Gamma \right] \cdot \mathbf{v}_i = \sum_j f_j \cdot y_j. \quad (61)$$

The left-hand side of this expression is obtained by numerical integration over the surrogate interface on the fluid mesh. The structure nodal force vector f_j is the yet to be determined force vector at node j of the structure mesh. After inserting (60) into (61), the nodal forces fulfilling the virtual work equality are obtained:

$$f_j = \sum_i \int_{\Gamma_I^f} (\sigma|_{\text{fluid}} \mathbf{n}) B_i d\Gamma N_j(\mathbf{x}_i^*). \quad (62)$$

In computing the conservative load vector, the main complexity lies in the determination of the closest points \mathbf{x}_i^* on the structure mesh. This can efficiently be performed using the approach sketched in Section 2.2.2.

5. EXAMPLES

We introduce several examples with rigid and elastic structures to showcase the accuracy and robustness of the developed method. Most of the considered applications are motivated by animal locomotion in air and water. First, we study the optimal thrust generation by a rigid elliptical airfoil with prescribed motion. Subsequently, the self-excited oscillation of an elastically mounted rigid cylinder subjected to a fluid flow is investigated. As a fluid–structure interaction problem with an elastic beam, we study the drag of a flexible fibre in a gravity-driven 2D flow. In the last two examples, the fluid flow resulting from a harmonic heaving motion of a flexible airfoil and a wing are considered. In all computations sufficiently fine grids are used in order to avoid the need for convection stabilisation. If not explicitly mentioned, the fluid equations are discretised with linear b-splines.

5.1. Heaving elliptic airfoil

The heaving elliptic airfoil is a well-studied problem representative for thrust and lift generation in flapping flight (see, e.g., [38, 39]). In our particular set-up, an elliptic airfoil is embedded in a uniform flow field and is subjected to prescribed harmonic heaving motion, that is, it moves orthogonal to the flow. The density of the fluid is $\rho^f = 1$. Following Wang [39], the thickness of the ellipse is chosen as 1/8 of its chord length $c = 1$. The airfoil is held at zero angle of attack with respect to the freestream flow $\mathbf{u}_\infty = (1, 0)^\top$. The simulation set-up is shown in Figure 10. The airfoil’s centre with the coordinates $X = (X_1, X_2)$ is forced to move with

$$X_1(t) = 0 \quad \text{and} \quad X_2(t) = A \sin(2\pi f t), \quad (63)$$

where A and f are the prescribed heaving amplitude and frequency, respectively.

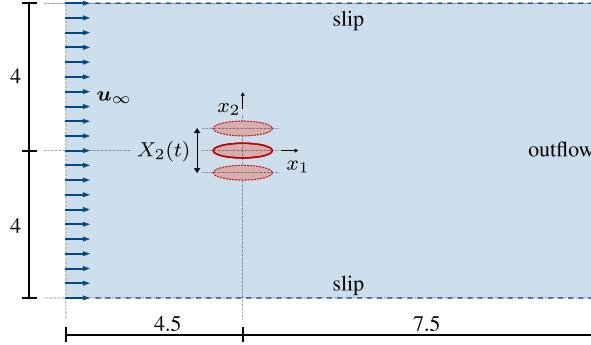


Figure 10. Computational fluid domain and the elliptic airfoil at its initial and maximally deflected positions.

In order to quantify the airfoil's performance at different heaving frequencies, the time-averaged input and output powers, P_{in} and P_{out} , are considered. Their ratio is the propulsive efficiency coefficient η . The time-averaged input and output powers are defined with

$$P_{\text{in}} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \dot{X}_2(t) F_2(t) dt, \quad P_{\text{out}} = \frac{-1}{t_2 - t_1} \int_{t_1}^{t_2} |\mathbf{u}_{\infty}(t)| F_1(t) dt \quad \text{and} \quad \eta = \frac{P_{\text{out}}}{P_{\text{in}}}, \quad (64)$$

where F_1 and F_2 are the horizontal and vertical force resultants, respectively. The convention is that $F_1 > 0$ is drag and $F_1 < 0$ is thrust. Hence, the negative sign in the definition of the output power implies that thrust generation is associated with a positive output power.

5.1.1. Convergence study. In a preliminary study, we investigate the mesh resolution needs and convergence of the developed fixed-grid finite element technique. In order to reduce the numerical problem size, the fluid domain is reduced to a box of size $(-1.5, 2.5) \times (-1, 1)$ around the airfoil. The Reynolds number is set to $Re = 100$, based on the chord length c and the inflow speed $|\mathbf{u}_{\infty}|$. The airfoil itself is held fixed, that is $A = 0$. The inflow speed is smoothly increased during the time interval $0 \leq t \leq 0.2$ and then kept constant. The considered time interval of interest is $0 \leq t \leq 1.25$, and the chosen time step size is $\Delta t = 0.0025$. The computed flow field is, after the initial transient, stationary.

The fluid domain is discretised with linear or quadratic b-splines with decreasing cell size. Because our actual aim is the efficiency analysis of the heaving airfoil, the average output power P_{out} in time interval $0 \leq t \leq 1.25$ is monitored for the different discretisations. In Figure 11, P_{out} for two different polynomial degrees and different cell sizes are plotted. The resulting numbers clearly converge towards a value slightly larger than -0.37 with increasing number of cells. As to be expected, the quadratic b-splines yield a value which is closer to the estimated limit value. Obviously, the improved convergence of quadratic b-splines comes with increased computational cost due to an

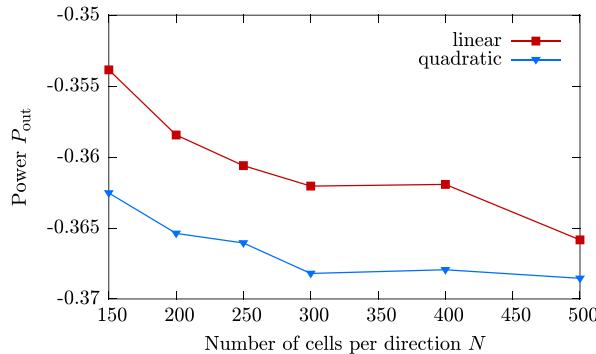


Figure 11. Heaving elliptic airfoil. Convergence of the time-averaged output power.

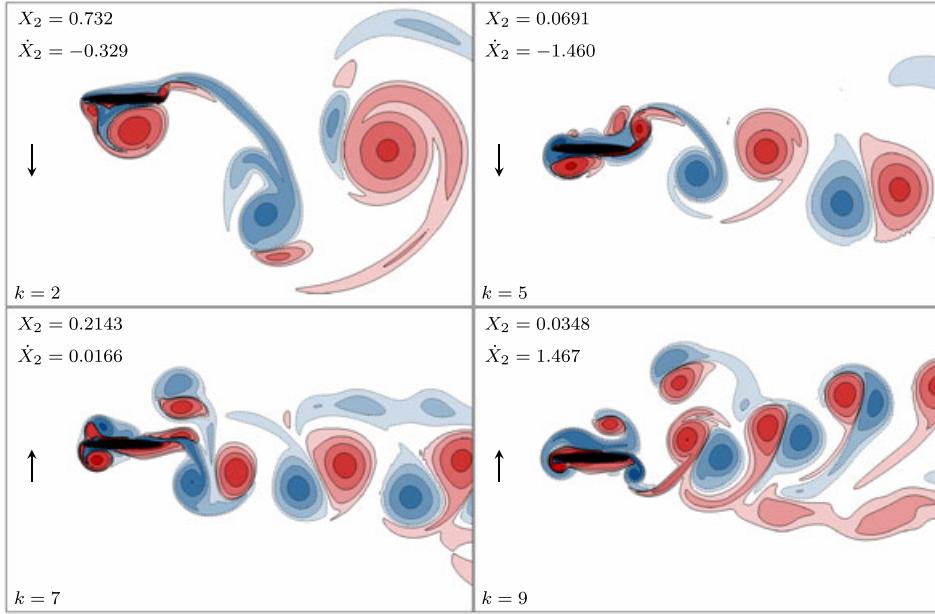


Figure 12. Heaving elliptic airfoil. Vorticity contour plots at $\tau = 8.75$ and selected reduced frequencies k . Contour lines indicate vorticity isocontours $\omega = \pm 0.25, \pm 1, \pm 4, \pm 16, \pm 64$, negative isocontours are dashed.

increase in bandwidth of the system matrix. When a direct solution algorithm is used for solving the systems equations, the increase in cost is significant. Moreover, due to the linear representation of the ellipse geometry on the fluid grid as discussed in Section 2.2.2, we cannot guarantee optimal convergence rates for polynomial degrees higher than linear. Based on our current implementation linear b-splines appear as the most economical in terms of quality of the results and the numerical cost involved.

5.1.2. Efficiency analysis. We now quantify the propulsive efficiency of the heaving elliptic airfoil at $Re = 500$ as a function of the heaving frequency. In order to eliminate the influence of the domain size, we use the larger fluid domain depicted in Figure 10. Furthermore, in accordance with Lewin *et al.* [38], we make use of the following set of dimensionless parameters

$$\tau = \frac{|\mathbf{u}_\infty|t}{c}, \quad \lambda = \frac{A}{c} \quad \text{and} \quad k = \frac{2\pi c f}{|\mathbf{u}_\infty|}, \quad (65)$$

which are the time and amplitude, and the reduced frequency, respectively. With these parameters, the vertical position of the ellipse is prescribed with $X_2 = c\lambda \sin(k\tau)$. Thus, the vertical velocity becomes $\partial_t X_2 = |\mathbf{u}_\infty|k\lambda \cos(k\tau)$. Here, $k\lambda$ denotes the maximal dimensionless velocity and is commonly referred to as *advance ratio*. Its value is related to the Strouhal number by a factor π or 2π , depending whether A or $2A$ is used in the definition of the Strouhal number ([39] or [38], respectively).

In our computations, the advance ratio is held fixed with $k\lambda = 1.5$, and the reduced frequencies $k = 2, 3, \dots, 9$ are considered. The grid size is chosen with 900×600 cells. The minimal grid distances are $h = 0.00444$ in x_1 -direction and 0.00667 in x_2 -direction within a box $(-1, 1)^2$ around the airfoil and increase towards the boundaries. The computations cover the time interval $0 \leq \tau \leq 8.75$. The time step is chosen with $\Delta t = 0.00125$. Based on the maximal relative velocity between the fluid and the ellipse u_{rel} , the Courant number $C = (u_{rel} \Delta t)/h$ is approximately 0.51.

Figure 12 shows vorticity snapshots for four different reduced frequencies $k = \{2, 5, 7, 9\}$ at time $\tau = 8.75$. The small arrows indicate the instantaneous velocity direction of the ellipse. As observed by Lewin *et al.* [38], there is a large variety of vortex shedding patterns depending on the

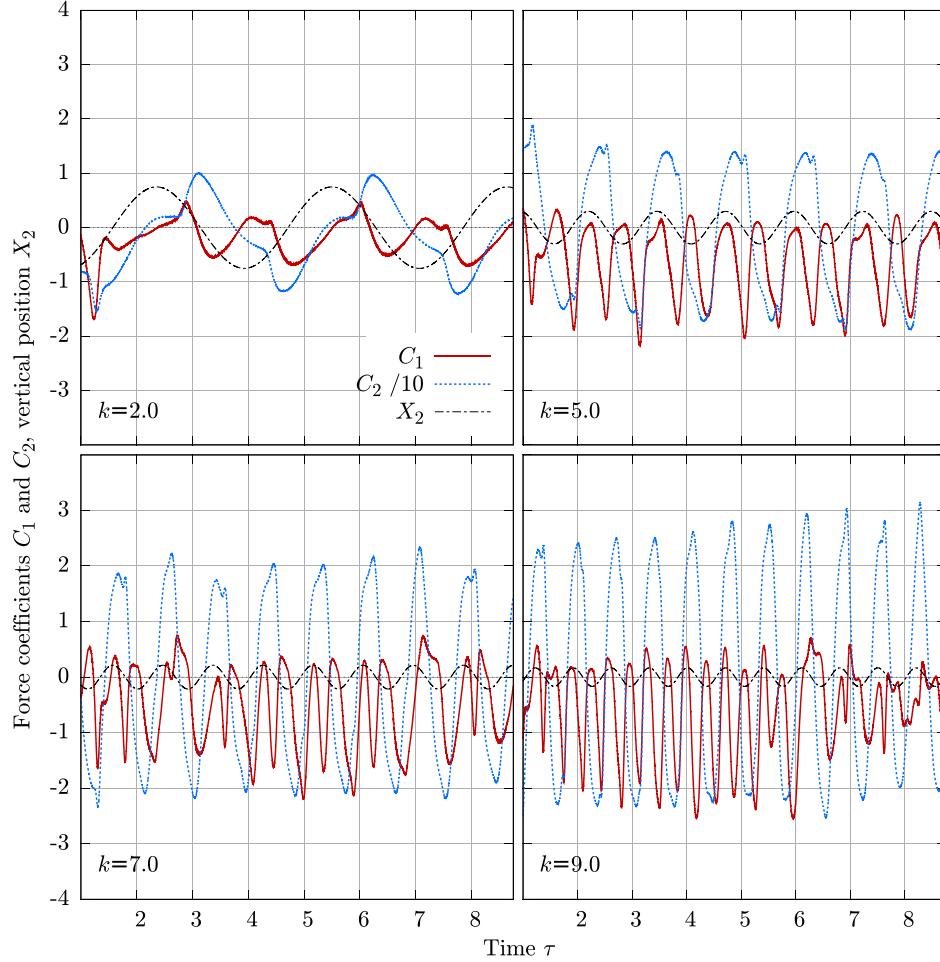


Figure 13. Heaving elliptic airfoil. Horizontal and vertical force coefficients C_1 and C_2 and the vertical position X_2 versus time for selected reduced frequencies k . Notice that in all plots the vertical force coefficient has been divided by 10 for display purposes.

reduced frequency k . Without going into detail, the vortex patterns for a few cases will be qualitatively described. At $k = 2.0$, a leading edge vortex (LEV) is generated during each half-stroke and is recaptured after the stroke reversal. Subsequently, the LEV is convected along the ellipse and merges with the trailing edge vertex being formed of the same orientation (Figure 12). The resulting strengthened vortex is stretched and eventually pinches off. In each half-stroke, one vortex of alternating sign is shed, and the wake flow is symmetric and periodic.

At $k = 5.0$, on the other hand, the vortex pattern is periodic but not symmetric. Here, the LEV remains longer close to the leading edge and does not immediately convect downstream. After stroke reversal, it is partially stretched around the leading edge and is dissipated by the newly generated LEV (of opposite orientation). Finally, a very weakened LEV is convected along the ellipse and joins the trailing edge vertex. The resulting vortex is stretched and eventually pinches off. An asymmetry in the flow develops because of the asymmetric stretching and dissipation of the LEV. As visible in Figure 12, the wake is deflected downwards. The flow pattern for the highest frequency $k = 9.0$ is more intricate and completely aperiodic. The LEVs formed in each half-stroke form a pair and convect downstream along the upper surface of the ellipse. But, at some point, such a pair remains slightly upstream of the leading edge and forms a much larger vortex pair with the following vortex pair. This vortex constellation then moves in a big arc around the upper side of the ellipse. It is

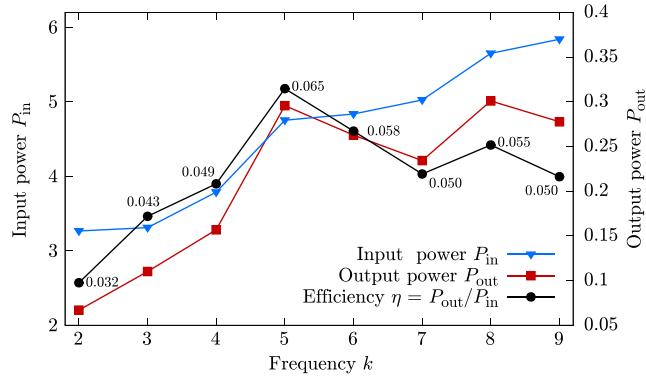


Figure 14. Heaving elliptic airfoil. Input and output powers, P_{in} and P_{out} , and efficiency coefficient η .

visible as the pair slightly above the vortex street in Figure 12. Moreover, this event seems to alter the side to which the wake is deflected. Such a sudden switch in the deflection of the wake from one side to the other is also reported in [38].

The observations pertaining to the vortex patterns are reflected in the variations of the nondimensional force coefficients in Figure 13. For nondimensionalisation, the chord length c , the free-stream velocity $|\mathbf{u}_\infty|$ and the density ρ are used. At the reduced frequency $k = 2.0$, the vertical force coefficient C_2 is periodic and symmetric. Increasing the heaving frequency leads to the loss of periodicity and symmetry. Importantly, the horizontal force coefficients C_1 are mostly negative, which implies that the force on the ellipse is opposite to the free-stream velocity so that in average thrust is generated. For the lower frequencies, the horizontal force coefficient lies almost entirely below the zero axis with an amplitude that increases with frequency. Beyond a certain frequency, the amplitude still increases but there are more and more positive forces, that is, drag is produced.

The computed average input and output powers and the efficiency are shown in Figure 14. Both, the input and the output powers increase for the frequencies from $k = 2.0$ to 5.0 . For higher frequencies, the input power still increases, but the output power levels off to a constant value. This behaviour is reflected in the variation of the efficiency, which increases quite rapidly with frequency up to $k = 5.0$, but slowly decreases afterwards.

As argued in [38, 39], there exists an optimal-reduced frequency k for a given advance ratio $k\lambda$. Recall that in all our computations, the advance ratio is $k\lambda = 1.5$. The results of our simulation are in good agreement with the findings of the studies by Lewin and Haj-Hariri [38], where the optimal heaving frequencies cluster between $k = 4$ and $k = 6$ for the advance ratios that have been considered. Also, the input power curve for $k\lambda = 1.5$ in [38] increases from $P_{in} \approx 3$ to 6 , very similar to our outcome. Nevertheless, our output power, even though with the same tendency, is lower than in [38]. We believe this is due to the small domain size we have chosen. In our set-up, the flow is much more confined between the inflow boundaries and, consequently, our thrust output is at a slightly lower level.

Finally, we plot in Figure 15 the time-averaged fluid velocity profile downstream of the airfoil. The horizontal velocity u_1 along the vertical axis at $x_1 = 1.5$, that is, one chord length behind the ellipse, and $-4 < x_2 < 4$ is averaged over the time period $1 < \tau < 8.75$. As can be seen in Figure 15, the velocity approaches to the freestream velocity $u_\infty = 1$ for $x_2 \rightarrow \pm 4$. In the direct wake of the airfoil for x_2 close to zero, two characteristic features are discernible, namely loss of symmetry and a jet-like profile. The momentum excess in the wake is a direct indication of the thrust generated.

5.2. Self-excited oscillating cylinder

A well-studied prototype problem for the study of aeroelastic flutter phenomena is the analysis of flow-induced vibrations of elastically mounted cylinders, see, e.g., [40]. The specific computational set-up we have chosen is shown in Figure 16. In the horizontal x_1 -direction, the cylinder is fixed,

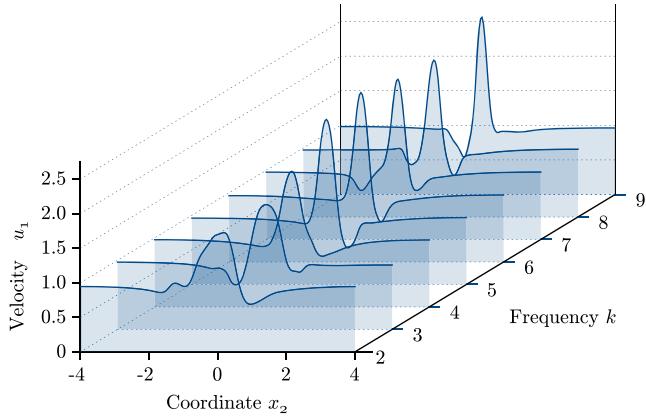


Figure 15. Heaving elliptic airfoil. Time-averaged horizontal velocity profile in the wake.

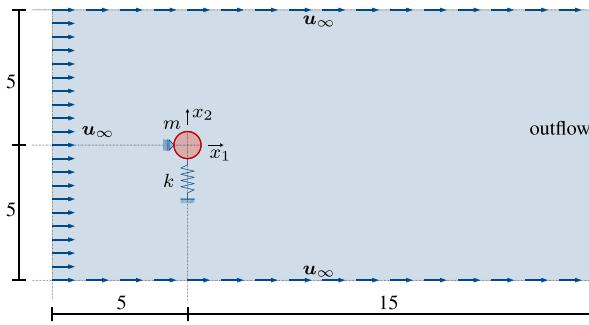


Figure 16. Elastically mounted cylinder in freestream flow. The left, lower and upper boundaries are inflow boundaries with prescribed velocity u_∞ , and the right boundary is an outflow boundary. The cylinder's diameter is $c = 1$ and mass is m , and the spring stiffness is k .

and in the vertical x_2 -direction, it is supported by a spring with stiffness k . The cylinder's mass is $m = 2.5$ and its diameter is $c = 1$. For the fluid, the freestream velocity is $\mathbf{u}_\infty = (1, 0)^\top$, the density is $\rho^f = 1$, and the viscosity is chosen such that $Re = 100$ (based on diameter c). The fluid domain of size 20×10 is discretised with 500×250 Cartesian cells of uniform size using linear b-splines. The time step size is chosen with $\Delta t = 0.05$ so that the Courant number is $C = 1.25$.

Initially, the cylinder is held fixed until the vortex street is fully developed. For this case, the obtained drag and lift coefficients are $C_D = 1.46$ and $C_L = 0.38$, respectively, and the Strouhal number is $St = 0.16$. These values agree well with numerical and experimental reference values [41].

After the start-up phase, the motion of the cylinder in the vertical x_2 -direction is released, and its dynamic is governed by

$$m \frac{d^2 x_2}{dt^2} + k x_2 = F_2(t), \quad (66)$$

where F_2 refers to the vertical component of the fluid force resultant acting on the cylinder. The equation of motion (66) is integrated in time with the trapezoidal rule. In each time step, once (66) is solved, the new vertical position and velocity of the cylinder provide the new fluid domain and its Dirichlet boundary conditions. In turn, solving the fluid problem provides a new fluid force resultant. As described in Section 4, this process is iterated until an equilibrium position is found.

As known, the spring stiffness k and the fluid velocity $|u_\infty|$ are critical for the vortex pattern appearing in the fluid and the structure displacements. Figure 17 shows the vertical displacement of the cylinder over time for three different spring stiffnesses $k = 1.11$, $k = 4.94$ and $k = 27.78$. The cylinder's displacement is significantly larger for $k = 4.94$ than for $k = 1.11$ (soft) and $k = 27.78$

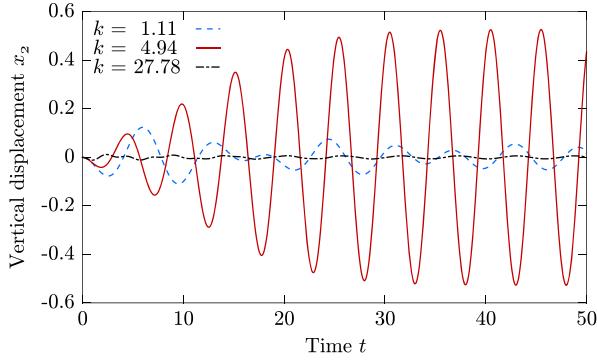


Figure 17. Self-excited oscillating cylinder. Vertical displacement of the cylinder over time for the cases $k = 1.11$, 4.94 and 27.78 ($U_R = 1.50$, 0.71 and 0.30).

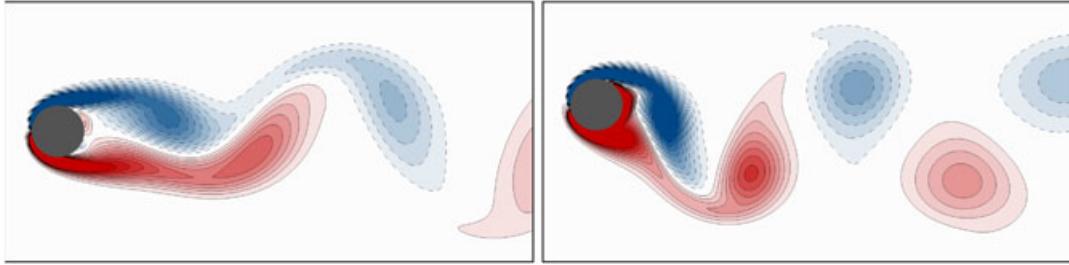


Figure 18. Self-excited oscillating cylinder. Vorticity contour plots for $k = 1.11$ (left) and $k = 4.94$ (right).

(stiff), which is an indication for resonance. The differences in the displacements are also reflected in the vorticity isocontours in Figure 18 comparing the cases $k = 1.11$ (soft) and $k = 4.94$ (resonant). In contrast to the soft case, the strong excitation of the cylinder in the resonant case leads to a larger number of shed vortices of more rounder shape. Also, the cores of vortices of opposite sign are vertically separated by a larger distance. The vorticity field for the stiff spring is almost identical to the one for the soft spring and is not shown.

For a more systematic study of the observed resonance, we consider a larger number of spring stiffnesses. Following Shiels *et al.* [42], the so-called reduced velocity U_R is a useful dimensionless steering parameter for the analysis. It is defined with

$$U_R = \frac{|\mathbf{u}_\infty|}{\omega_n c}, \quad (67)$$

where $\omega_n = \sqrt{k/m}$ is the natural frequency of the oscillator. By visually fitting the harmonic function $A \sin(\omega t - \phi)$ to the computed cylinder displacements, values for the amplitude A and the frequency ω are obtained (the phase shift ϕ is not reported in the following). These values for all considered reduced velocities U_R are plotted in Figure 19. One can distinguish two regions: the resonance interval, which lies approximately between $0.6 < U_R < 1.2$, and the flow-dominant region for values below or above this interval. In the flow-dominant region, the response of the oscillator is mainly governed by ordinary vortex shedding. We observe a comparably low amplitude, and the frequency is close to the Strouhal number of the fluid flow, that is, $\omega = 2\pi St U_R \omega_n = 2\pi f_V$ with f_V being the frequency of vortex shedding. In the resonance window the amplitude is much larger (by more than an order of magnitude) and the oscillation frequency is very close to the natural frequency of the cylinder, that is, $\omega/\omega_n \approx 1$. This phenomenon is referred to as *lock-in*. Our computational results are in very good agreement, qualitatively and quantitatively, with Shiels *et al.* [42].

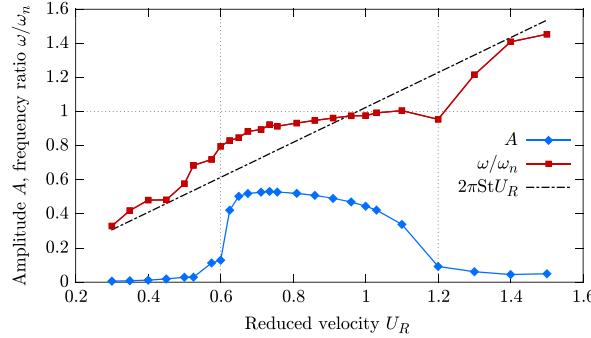


Figure 19. Self-excited oscillating cylinder. Amplitude A and frequency ratio ω/ω_n of the oscillating cylinder with natural frequency ω_n for different reduced velocities. The dash-dotted line indicates vortex shedding for a rigidly mounted cylinder.

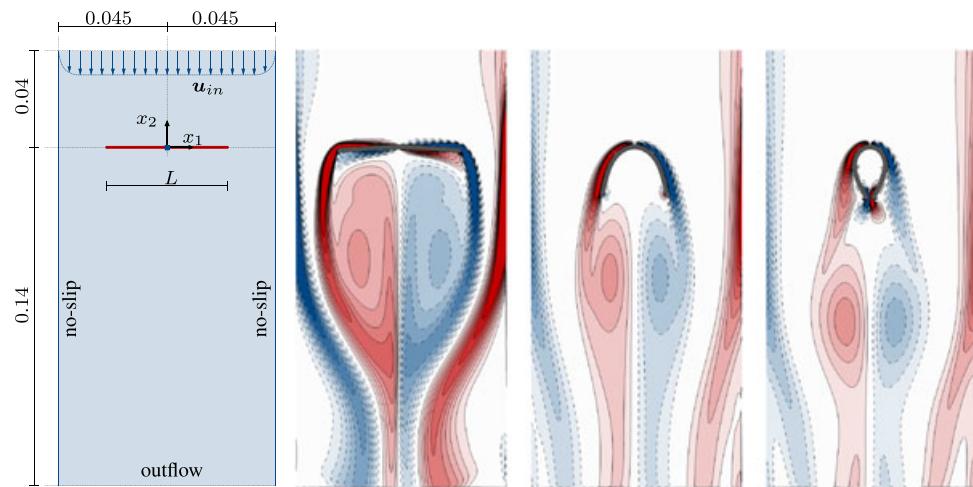


Figure 20. Computational fluid domain and fibre (left). Representative vorticity contour plots for fibres with three different bending stiffnesses $EI = 1.25 \cdot 10^{-4}, 1.25 \cdot 10^{-7}$ and $4.17 \cdot 10^{-8}$ (from left to right). In the contour plots, the inflow velocity ($|\mathbf{u}_{in}| = 2.5$) and the time ($t = 0.085$) are the same.

5.3. Flexible fibre in a soap film

A gravity-driven soap film flow is used in Alben *et al.* [43] to experimentally investigate the drag forces on flexible fibres subject to quasi-2D flows. The experiments yield drag scaling laws as a function of the soap film velocity and fibre bending rigidity and length. Similar to Zhu *et al.* [44], we use this experiment as a case study to demonstrate the accuracy and robustness of our fluid–structure coupling technique in the 2D setting.

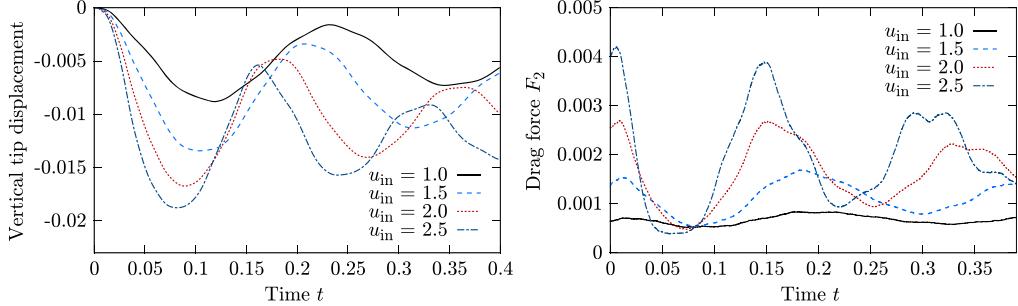
Our computational domain is depicted in Figure 20, left. The fluid model for the soap film takes into account the air resistance and the weight of the film using a modified momentum equation:

$$\rho_f \frac{\partial \mathbf{u}}{\partial t} + \rho_f (\mathbf{u} \cdot \nabla) \mathbf{u} + \lambda \mathbf{u} - \nabla \cdot \boldsymbol{\sigma} = \rho_f \mathbf{g}. \quad (68)$$

Here, λ is the scalar air resistance parameter, and $\mathbf{g} = -g \mathbf{e}_2$ is the gravitational acceleration acting downwards. In line with [44], we chose $\lambda = \rho_f g / |u_{in}|$ and apply the inflow velocity profile indicated in Figure 20, left. The boundary condition for the left and right boundaries is no-slip and for the bottom boundary outflow. In Table I, the parameters used in the simulations are summarised. The fibre is assumed to be inextensible, and a relatively high value for the axial stiffness EA is chosen. The highest value for the bending stiffness is $EI = 1.25 \cdot 10^{-4}$, which models effectively a rigid fibre. Using the fibre's length L and the inflow velocity u_{in} , the computed Reynolds numbers

Table I. Flexible fibre in a soap film. Parameters for the computation.

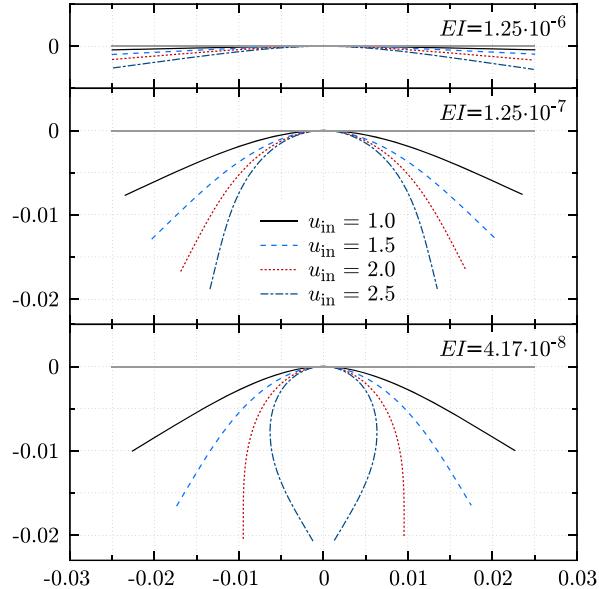
ρ_f	μ	u_{in}	$\rho_s A$	EA	EI	L	g
0.003	$1.2 \cdot 10^{-6}$	1.0–2.5	$7 \cdot 10^{-3}$	∞	$(4.17-125) \cdot 10^{-8}, 1.25 \cdot 10^{-4}$	0.05	9.80

Figure 21. Flexible fibre in a soap film. Temporal behaviour of the vertical tip displacement (left) and the drag force (right) for a bending stiffness of $EI = 1.25 \cdot 10^{-7}$ and different inflow velocities.

range from $Re = 50$ to 312.5. The vorticity contours in Figure 20 show representative vorticity distributions and fibre deflections for three different bending stiffnesses.

The temporal variation of the vertical displacement of the fibre tip and the drag force are plotted in Figure 21 for four different inflow velocities and a fibre with a medium bending stiffness. It can be observed how the fibre tip initially deflects by a large amount and subsequently swings back and forth with a slightly diminishing amplitude. The drag force reduces to a minimum each time the fibre is maximally bent and, in turn, attains a large value when it swings back.

The deflected fibre shapes at $t = 0.085$ for three different bending stiffnesses, and all considered inflow velocities are shown in Figure 22. The deflected shapes for $u_{in} = 2.5$ correspond to the three vorticity contours shown in Figure 20. As to be expected, the deflection of the fibre is proportional to the inflow velocity and inversely proportional to its stiffness. For a more quantitative analysis,

Figure 22. Flexible fibre in a soap film. Deflected fibre shapes at $t = 0.085$ for different bending stiffnesses and all considered inflow velocities.

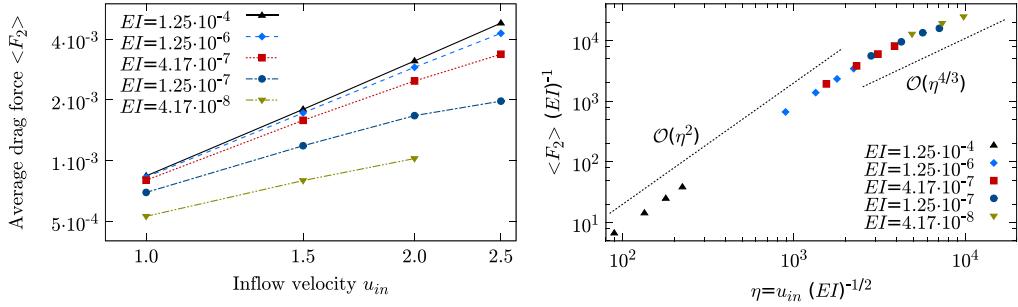


Figure 23. Flexible fibre in a soap film. Time average of drag forces versus inflow velocity (left) and scaling analysis of these results (right).

Figure 23 shows the time-averaged drag force versus inflow velocity for different stiffnesses. The average drag force $\langle F_2 \rangle$ is taken as the root mean square of the values of F_2 in the last cycle of oscillation. In Figure 23, one can clearly see that the drag force scales quadratically with the inflow velocity for the most stiff (practically rigid) fibre. For softer fibres, the drag grows more slowly as the velocity increases. This phenomenon is, of course, due to the reconfiguration of the fibre shape into a more streamlined shape.

To account for the varying stiffness of the fibre, we consider the average drag divided by the stiffness $\langle F_2 \rangle/EI$ as a function of the parameter $\eta = u_{in}/\sqrt{EI}$. Our definition of η is motivated by the analysis of Alben *et al.* [43], who include also the fluid density and fibre thickness and length. Since in our study, these values are constant (cf. Table I), they are not essential for the subsequent scaling results. Figure 23 shows how $\langle F_2 \rangle/EI$ varies with η . The functional dependence changes from $\mathcal{O}(\eta^2)$ to $\mathcal{O}(\eta^{4/3})$ for increasing values of η . This outcome is in agreement with the experimental observations made in [43]. Note that in reference [43], the range of considered Reynolds numbers is significantly higher than in our analysis.

Next, we consider in more detail the computation with the softest bending stiffness $EI = 4.17 \cdot 10^{-8}$ and the largest inflow speed $u_{in} = 2.5$. In Figure 24, the temporal evolution of the drag force F_2 and the tip displacement are overlayed. Mostly, the evolution of the drag force is in phase with the tip displacement. Only in the time interval $0.08 < t < 0.1$ is there a distinct difference between the force and displacement evolution. In fact in this time interval, the drag force becomes negative meaning that thrust is generated. The dip in the drag force is best explained with the vorticity contours, and the deflected shape of the fibre shown in Figure 25. The first two contour plots in Figure 25, at $t = 0.02$ and $t = 0.04$, show vortices forming in the wake of the fibre and the shear layers at the film boundaries. The fibre itself begins to deform with a delay due to its inertia.

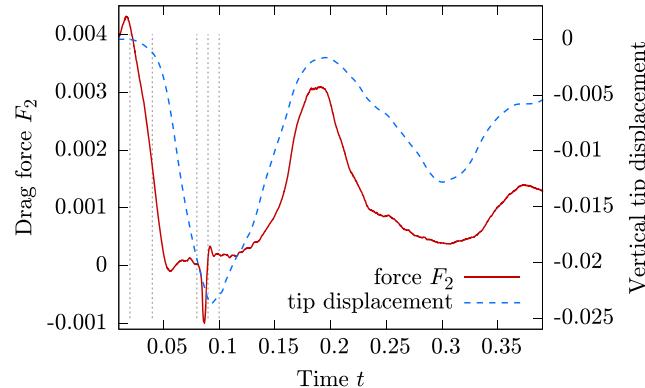


Figure 24. Flexible fibre in a soap film. Vertical force and tip displacement versus time for $EI = 4.17 \cdot 10^{-8}$ and $u_{in} = 2.5$ (the vertical lines refer to the time instants in Figure 25).

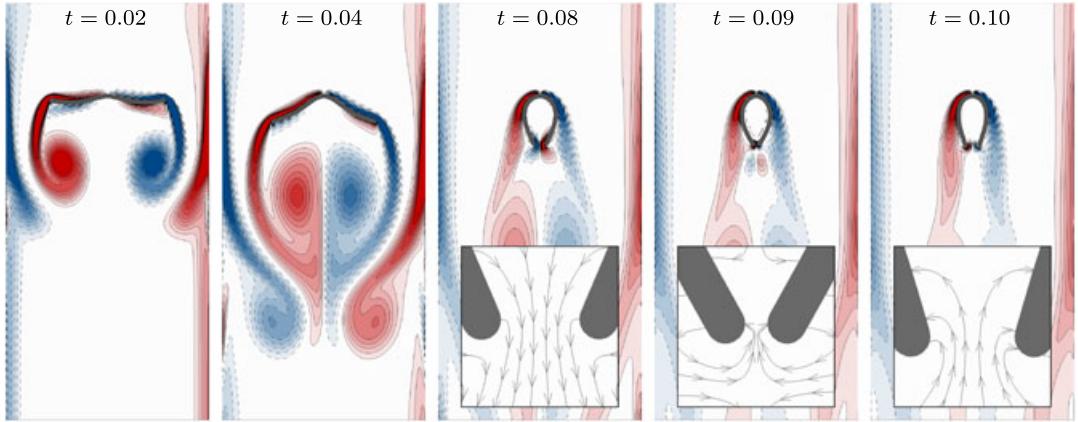


Figure 25. Flexible fibre in a soap film. Vorticity contours for the softest fibre with $EI = 4.17 \cdot 10^{-8}$ and inflow $u_{in} = 2.5$; the inset snapshots show the streamlines around the closing tips of the fibre.

The other three contour plots in Figure 25 show the closing and opening of the fibre causing the dip in the drag force. Additional inset figures show a zoom of the closing tip area together with streamlines. At $t = 0.08$, the fibre is in the process of closing, and fluid is ejected downstream. This state corresponds to the dip of drag force in Figure 24. Approximately, at $t = 0.09$ the state of maximal displacement is reached, and a small vortex pair formed by the ejected fluid begins to travel downstream. Subsequently, at $t = 0.1$ one can see how fluid is sucked into the interior of the shape formed by the closed fibre.

Finally, it is worth emphasising that the configuration shown in Figure 25 with the near contact of the fibre tips comes close to a topology change of the fluid domain. It is challenging to treat such configurations with a body-fitted mesh without significant remeshing. In our fixed-grid approach, the Cartesian background grid remains the same throughout the simulation.

5.4. Flexible heaving fibre

The aim of this example is the study of the effect of wing inertia and flexibility on hovering performance of insects. A similar study has been reported, for instance, in [45]. In our 2D study, a flexible fibre of unit length is placed at the centre of the fluid domain with zero freestream velocity (Figure 26). The fibre is clamped at its left end and free at its right end. The clamped end undergoes

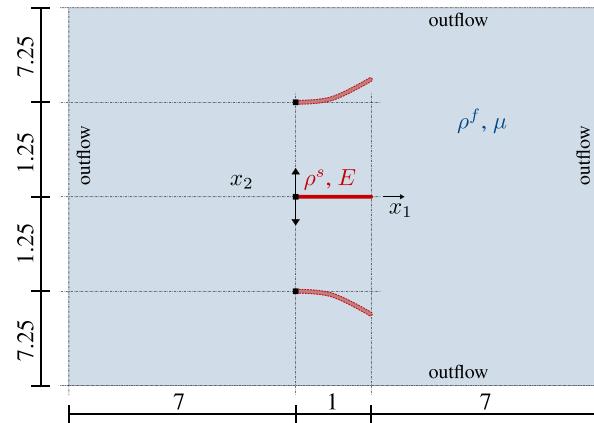


Figure 26. Flexible heaving fibre in a fluid domain with zero freestream velocity and only outflow boundaries (not drawn to scale). The fibre is shown in its initial and (possible) extremal positions.

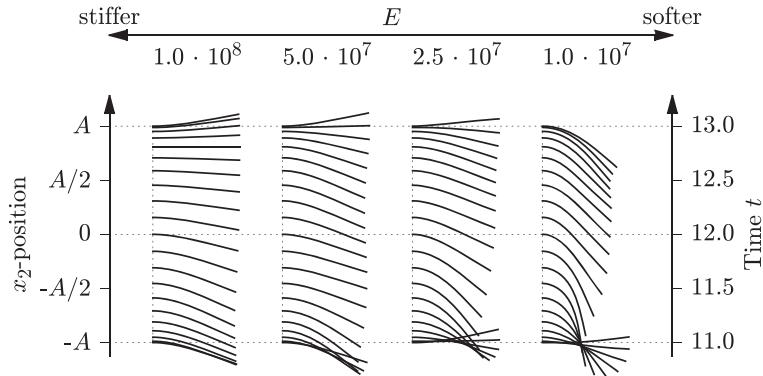


Figure 27. Flexible heaving fibre. Fibre deflections during one upstroke $11.0 \leq t \leq 13.0$.

a prescribed harmonic vertical motion, that is, heaving,

$$x_2(t) = A \sin(2\pi f t) \quad \text{with } A = 1.25 \text{ and } f = 0.25.$$

The initial position of the fibre and possible extremal positions are depicted in Figure 26. The fluid has a mass density of $\rho^f = 1000$ and viscosity of $\mu = 10$. The maximal Reynolds number based on the fibre length $c = 1$ and the prescribed maximum velocity is approximately $Re = 200$.

The spatial discretisation of the fluid domain consists of 200×350 cells with varying sizes from $h_{\min} = 0.026$ to $h_{\max} = 0.41$. The smallest cells are placed in the region swiped by the elastic fibre with the cell sizes gradually increasing towards the outflow boundaries. The time step size for the computation is chosen with $\Delta t = 0.0025$. The elastic fibre, on the other hand, is discretised with 24 cubic b-spline basis functions in space and the trapezoidal rule in time. The mass density of the fibre is $\rho^s = 8000$ and its thickness is $t = 0.08$. Fibres with Young's moduli $E = 1.0 \cdot 10^7, 2.5 \cdot 10^7, 5.0 \cdot 10^7, 1.0 \cdot 10^8$ and a rigid fibre are considered.

Figure 27 shows the fibre deflections during one upstroke, $11.0 \leq t \leq 13.0$, for the four considered flexible fibres (the rigid case is left out). One can see how the free right end deflects significantly and follows the motion of the driven left end. This effect is rather small for the stiffest case and drastically increases with flexibility. In the most flexible case, the right end lags about half a chord length behind the left end.

Figure 28 shows the vorticity contours for three different time instants during an upstroke for the rigid and the most flexible case. In the rigid case, even though the vortex pattern is not symmetric, there is no distinct bias towards the left or the right, and the vortices move mainly in the vertical direction away from the fibre. On the contrary, in the flexible case, the vortices are mainly shed to the right during the stroke.

The breaking of the symmetry in the vortex pattern has an immediate effect on the horizontal and vertical resultant fluid forces acting on the fibre. The evolution of both force components is shown in Figure 29 for the rigid and all four flexible fibres for the time interval $2.0 \leq t \leq 14.0$. For the rigid fibre, the horizontal force resultant F_1 is initially almost zero but starts to oscillate with increasing simulation time. Nevertheless, as pointed out before, its average over time remains close to zero. For all flexible fibres, the horizontal force resultant F_1 remains mostly negative indicating thrust generation. With decreasing Young's modulus, F_1 becomes more and more positive for extended periods of time, hence impeding the average thrust. As expected increasing flexibility leads to a decrease in the peak vertical force F_2 .

Similar as in the heaving elliptic airfoil from Section 5.1, we carry out an efficiency analysis by comparing the average input power P_{in} (64) and the average thrust defined as the time-averaged value of the negative horizontal force resultant, i.e. $T = \langle -F_1 \rangle$. In order to capture the rigid case, these quantities are plotted in Figure 30 as functions of the inverse Young's modulus. As to be expected from the force plots in Figure 29, the increase in flexibility drastically reduces the

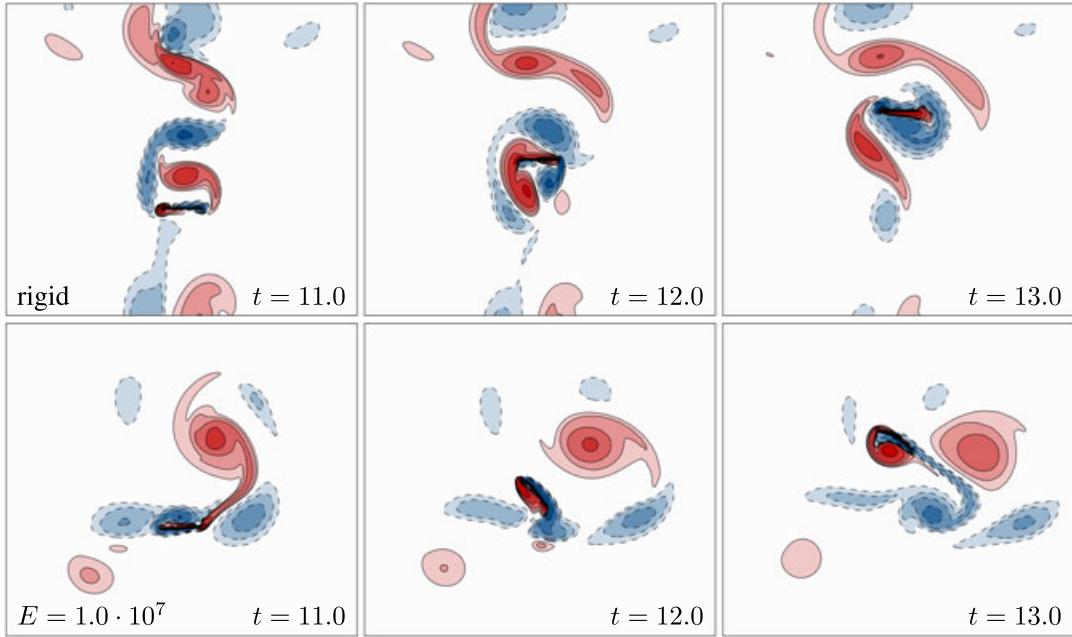


Figure 28. Flexible heaving fibre. Vorticity contours during one upstroke between $t = 11.0$ to $t = 13.0$. The top row shows a rigid fibre and the bottom row a flexible fibre. The vorticity isocontours are at $\omega_z = \pm 1, 2, 4, 8, 16$, negative values are dashed.

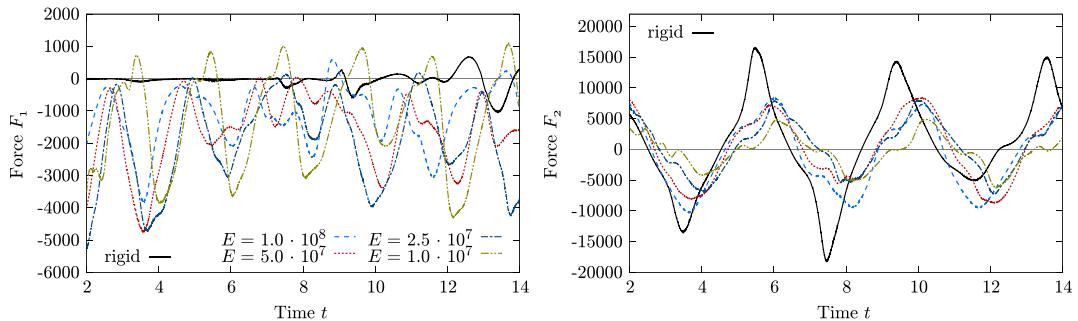


Figure 29. Flexible heaving fibre. Horizontal (left) and vertical (right) fluid force resultant on the fibre.

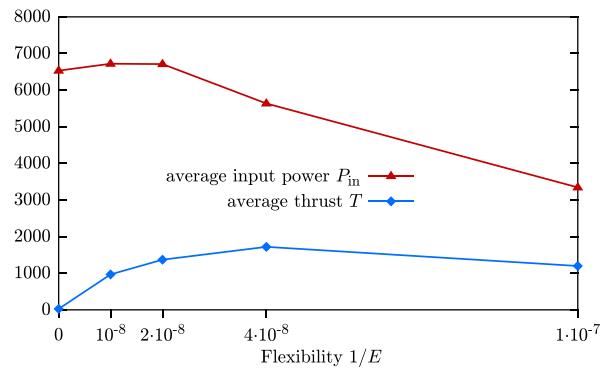


Figure 30. Flexible heaving fibre. Average input power and thrust for the five considered fibres.

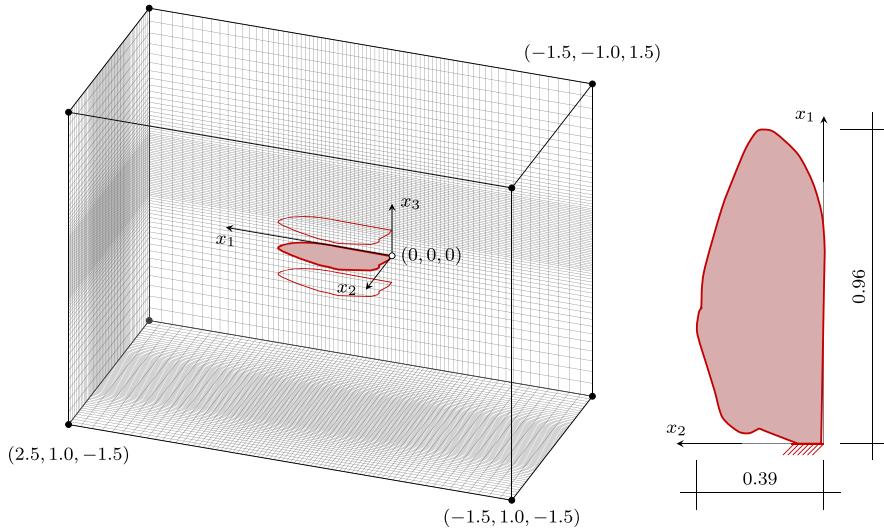


Figure 31. Flexible heaving drosophila wing immersed in a three-dimensional fluid box with zero freestream velocity and only outflow boundaries. The wing is shown in its initial and (undeflected) extremal positions. Some of the grid lines on the surface of the box are shown to indicate the volume grid. On the right the plan view of the wing is shown.

input power. In contrast, the average thrust generated appears to have a maximum in the range of considered Young's moduli. These plots strongly suggest the existence of an optimal flexibility.

5.5. Flexible heaving drosophila wing

As a last example, we consider the forces generated by a heaving Drosophila wing in a fluid with zero freestream velocity. This example is the extension of the heaving fibre example in Section 5.4 to three dimensions. The computational domain and the elastic wing are shown in Figure 31. The plan-form of the wing resembles a drosophila wing [46]. The clamped root is subjected to a prescribed oscillatory vertical motion,

$$x_3(t) = A \sin(2\pi f t) \quad \text{with } A = 0.25 \text{ and } f = 0.25.$$

The amplitude of the motion A is approximately a quarter of the span and more than half of the wing's chord length. The fluid parameters are chosen as for the flexible heaving fibre in Section 5.4. This leads to a Reynolds number of $Re = 40$, which has been chosen relatively low in order to ensure a stable fluid solution for a relatively coarse grid. The mass density of the wing is chosen with $\rho^s = 8000$ and its Poisson's ratio with $\nu = 0$. Wings with three different Young's moduli $E = 1.0 \cdot 10^8$, $5.0 \cdot 10^7$ and $2.5 \cdot 10^7$ are considered.

The flow domain is as shown in Figure 31 discretised with $76^3 = 438,976$ cells with sizes ranging from $h_{\min} = 0.04$ near the wing to $h_{\max} = 0.17$ close to the outflow boundaries. As basis functions linear b-splines are used. The wing is discretised with 106 triangular subdivision shell elements. The time step size is chosen as $\Delta t = 0.0025$.

In order to visualise the flow behaviour, Figure 32 shows five different snapshots of vortex rings generated during the wing motion. The vortices are identified with the Q -criterion [47] and the isosurfaces for $Q = 0.5$ are shown in the pictures. These rings are clipped in order to increase the visibility of their internal structure and their structure close to the wing. The iso-surfaces are coloured by their direction of rotation as indicated by the curved arrows in the top pictures. The left two pictures show the downstroke, the bottom picture the stroke reversal at the bottom and the right two pictures the upstroke. During the downstroke, the starting vortex ring (blue) stays attached to the wing and increases in intensity until the wing decelerates, and the ring begins to detach. When moving in the reverse direction, a new ring forms (red), which stays attached and increases

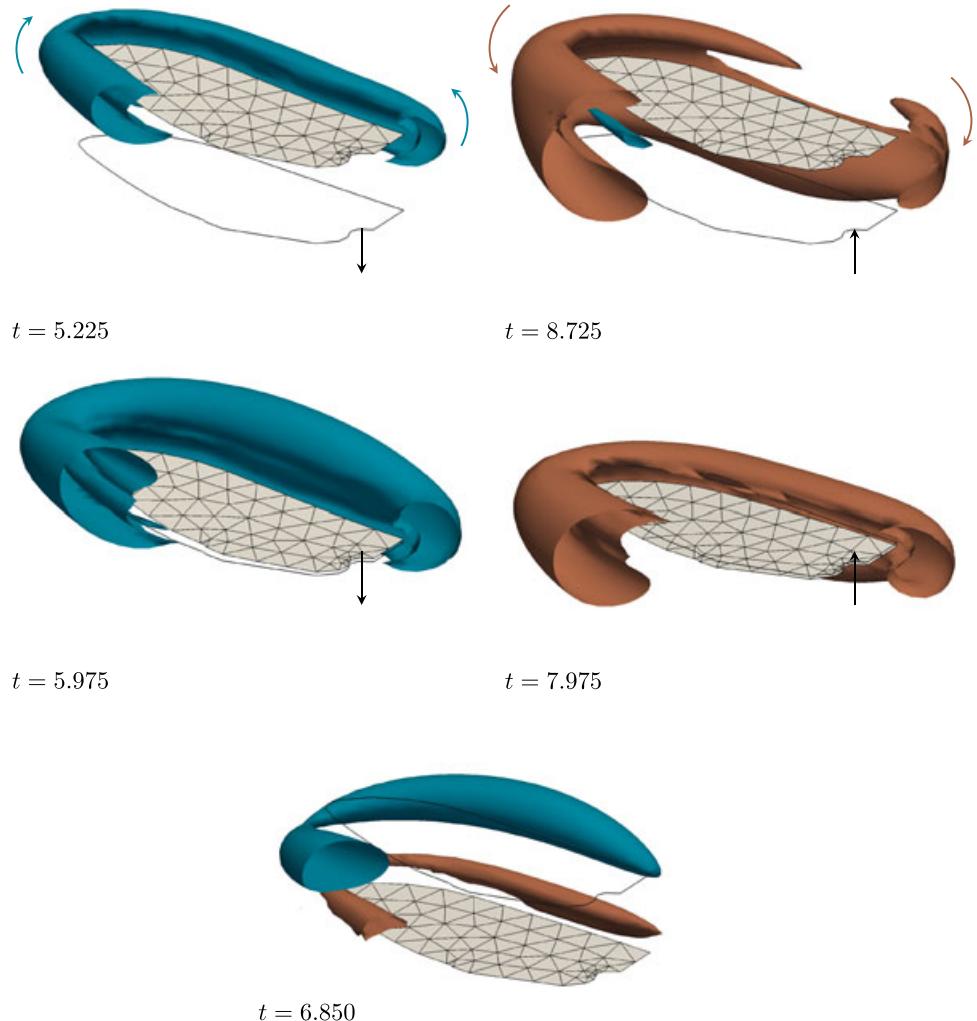


Figure 32. Flexible heaving drosophila wing. Snapshots of vortex rings visualised using clipped isosurfaces during a stroke. The straight arrows indicate the velocity direction of the wing root, the curved arrows in the top two snapshots indicate the rotation direction of the vortex rings, and the solid black lines give the initial location of the wing.

in intensity during the upward motion until it detaches when the wing decelerates again. For a more in-depth discussion of vortex generation and shedding in flapping flight see, for instance, Ellington *et al.* [48].

A more quantitative analysis that reveals the effect of the wings flexibility is obtained by considering the temporal evolution of the integral force acting on the wing. Figure 33 shows the short- and the long-term behaviours of the force components F_1 (spanwise) and F_3 (orthogonal to the wing). The short-term responses are characterised by high amplitudes and irregular transients, which diminish after $t > 3$. Whereas the curves for all three wing stiffnesses are very similar for $t < 0.3$, they differ significantly for later times, especially for the spanwise force component F_1 . Nevertheless, after the transients vanish the force responses vary like a harmonic curve. The right column of plots in Figure 33 displays the outcome for $4 < t < 11$ and the two lower stiffnesses. There is effectively no difference in the long-term behaviour ($t > 4$) for the orthogonal force F_3 , but the other component F_1 shows a notable difference for the two cases. The amplitude of F_1 is inversely proportional to the wing's stiffness. The long-term plots additionally display the time average of the values for an interval $5 < t < 9$, which corresponds to a full stroke. Relative to the amplitudes of the F_3 responses,

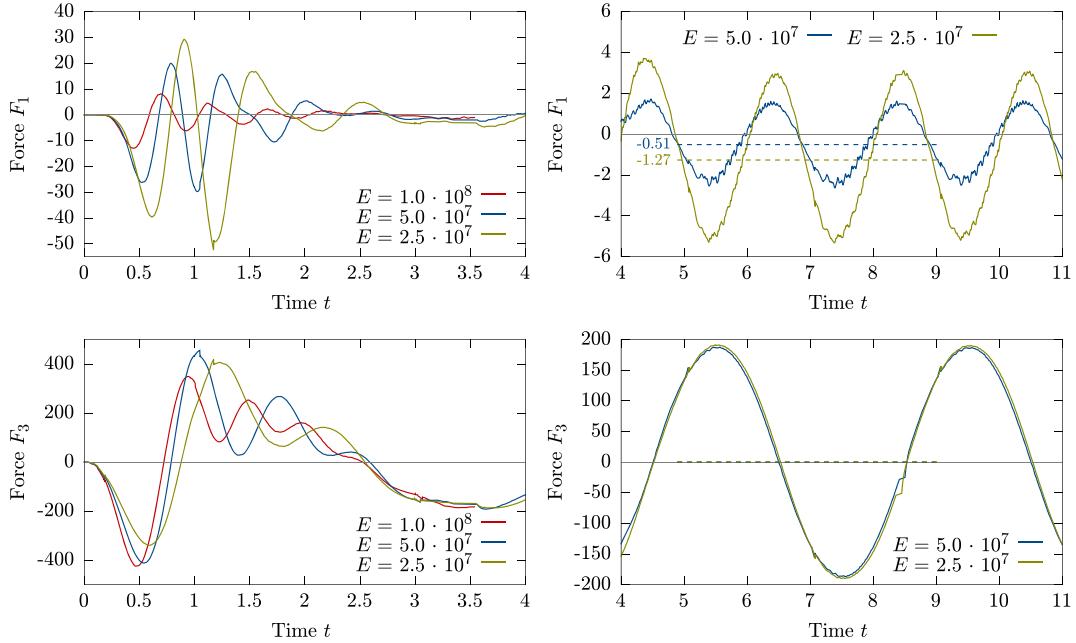


Figure 33. Flexible heaving drosophila wing. Components of the integral surface force \mathbf{F} over time t . The initial phase $0 < t < 4$ is shown on the left column and long-term behaviour $4 < t < 11$ on the right. F_1 is the spanwise force and F_3 is the force orthogonal to the undeflected wing. The dashed lines in the long-term plots show the temporal average for the interval $5 < t < 9$.

their average is practically zero, which corresponds to the symmetry of the problem. Notably, this is not the case for the spanwise force component F_1 of which the time average is clearly below zero. This value indicates the tendency of the wing to move in the x_1 -direction towards the clamped end. Moreover, this bias is stronger for the wing with the lower stiffness as indicated by the values in the figure. The average thrust is approximately twice as large for the wing with half the flexibility.

6. SUMMARY AND CONCLUSIONS

We introduced a fixed-grid finite element technique for fluid–structure interaction problems by building on our previous work on subdivision-stabilised incompressible fluid elements [5], isogeometric beam and shell elements [12, 13], and immersed fluid–structure coupling techniques [4, 14]. Both the incompressible Navier–Stokes and the structural equations are discretised with spline basis functions. As it is becoming apparent, spline basis functions have certain advantages over conventional finite element basis functions. Amongst others, splines have higher smoothness and can be easily refined or degree elevated [8]. These unique properties promote the development of novel finite elements for fluids and structures. For instance, the refinability property enables the subdivision-stabilisation technique for incompressible elements [5], and the higher order smoothness enables the rotation-free beam and shell elements [12, 13].

In this work, we discretised the fluid equations with tensor-product b-splines and used the subdivision-stabilisation technique introduced in [5] for ensuring inf-sup stability. The Dirichlet boundary conditions are enforced with the Nitsche technique combined with a novel cut-cell stabilisation method. Different from our earlier work [5], the size of the cells can be nonuniform, although the connectivity of the grid still has to be Cartesian. It is crucial for numerical efficiency that the cell sizes are locally adapted to resolve the shear layers in the flow, especially in 3D. Additional efficiency gains are achieved by solving the Navier–Stokes equations with a pressure-correction scheme. At every time step, we solve one convection–diffusion problem for each velocity component, one Poisson problem for pressure and one auxiliary L_2 -projection problem. The size

of each of the discrete systems of equations is significantly smaller than for an implicit scheme, which leads to savings in memory and computational time. For instance, for the 3D heaving wing example from Section 5.5, we solve three equations with $\approx 4.4 \times 10^5$ and two equations with $\approx 5.5 \times 10^4$ unknowns. In case of an implicit scheme, it would be necessary to solve one equation with $\approx 1.4 \times 10^6$ unknowns. Moreover, the discrete equations resulting from a pressure-correction scheme are more amenable for parallel solution with standard domain decomposition techniques and preconditioners [49].

The robust coupling of fluid–structure equations with a partitioned approach is especially challenging when a fixed-grid approach is used, and the fluid is incompressible. This is further exacerbated when the fluid equations are solved with a pressure-correction scheme. The implemented Dirichlet–Robin coupling algorithm proved to be effective in addressing the widely reported instabilities specific to partitioned coupling schemes. An additional possible source of instability are inconsistencies at the nonmatching fluid–structure interface, which usually lead to large unphysical pressure oscillations and ultimately to the divergence of the solution. The introduced operator for transfer of fluid tractions to the structure and the reverse operator for the transfer of structure position and velocity to the fluid greatly increase the robustness of the coupling scheme. Additional subiterations in the pressure-correction scheme are effective in case of complex interface geometries and/or large structural deformations within one time step. Our examples motivated by animal locomotion in air and water demonstrate the robustness and accuracy of the proposed coupling technique. They cover a wide range of problems with different fluid to structure mass density ratios, structural stiffnesses, large deformations and near topology changes. The developed fixed-grid approach is particularly promising for large 3D applications due to the associated difficulties in mesh generation.

The proposed fluid–structure coupling approach is readily extensible and further gains in performance and accuracy can be achieved. It is necessary to have a better control of cell sizes in the shear layers close to the boundaries and interfaces, especially for higher Reynolds numbers and in 3D [50]. Although the used isoparametric mapping enables somewhat local refinement, it is beneficial to consider adaptive local h-refinement, such as with hierarchical b-splines [51–53] or T-splines [54]. As discussed, the approximation of the fluid–structure interface with level sets results in a linear representation of the interface on the grid. This leads to suboptimal convergence rates for b-splines of degree higher than one. The level set representation is however crucial for dealing with geometrically complex interfaces. It restricts the number of canonical cases how an interface intersects a cell. Specifically, there are two cases in two dimensions and, ultimately, three distinct cases in three dimensions. Without the geometric filtering provided by the level set representation, the number of possible intersections is unlimited. This makes it very challenging to integrate the resulting cut-cells accurately. We believe that robust techniques are needed that can integrate simple cut-cells accurately, but fall back to a linear level set representation in case of geometrically complex cut-cells.

APPENDIX A: ITERATIVE PROJECTION SCHEME

The accuracy of the projection scheme presented in Section 2.1.2 can be improved by employing subiterations within each time step. Although subiterations are usually not used in fluid-only applications with fixed boundaries, they may increase the robustness of the solution algorithm when moving boundaries are present. As the structure moves through the fixed fluid grid, cells outside the fluid domain are activated. The newly activated finite element nodes do not have a solution history. As introduced in Section 2.1.2, we assign each uncovered node, the solution data of its closest point on the fluid–structure interface. This is, however, only an approximation, which can be improved by employing iterations during the pressure correction solution at each time step.

The starting point for the derivation of an iterative projection scheme, which iteratively converges to the implicit Navier–Stokes solution, is the algorithm introduced in Section 2.1.2. Before giving the algorithm, first we introduce a second index (k) for labelling the iteration numbers such that at time point $n + 1$ the velocity, pressure and incremental pressure are denoted with $\mathbf{u}^{(k)}$, $p^{(k)}$ and $\psi^{(k)}$

during the iterations, respectively. Given the solution at time t^n , the new solution at time t^{n+1} is determined as follows:

1. For $(k) = (0)$, to begin the iteration initialise the velocity and pressure with the values from last time step

$$\mathbf{u}^{(0)} = \mathbf{u}^n \quad \text{and} \quad p^{(0)} = p^n. \quad (\text{A.1})$$

2. Compute the velocity $\mathbf{u}^{(k+1)}$ by solving

$$\begin{aligned} \frac{\rho_f}{\Delta t} \mathbf{u}^{(k+1)} + \rho_f (\mathbf{u}^{(k)} \cdot \nabla) \mathbf{u}^{(k+1)} - \mu \nabla^2 \mathbf{u}^{(k+1)} &= \frac{\rho_f}{\Delta t} \mathbf{u}^n - \nabla (p^{(k)} + \psi^n) && \text{in } \Omega^f \\ \mathbf{u}^{(k+1)} &= \bar{\mathbf{u}}^{n+1} && \text{on } \Gamma_D^f \\ \mu \partial_{\mathbf{n}} \mathbf{u}^{(k+1)} &= p^{(k)} \mathbf{n} && \text{on } \Gamma_N^f. \end{aligned} \quad (\text{A.2})$$

3. Compute the pressure increment $\psi^{(k+1)}$ using $\mathbf{u}^{(k+1)}$ from step 2

$$\begin{aligned} \nabla^2 \psi^{(k+1)} &= \frac{\rho_f}{\Delta t} \nabla \cdot \mathbf{u}^{(k+1)} && \text{in } \Omega^f \\ \partial_{\mathbf{n}} \psi^{(k+1)} &= 0 && \text{on } \Gamma_D^f \\ \psi^{(k+1)} &= 0 && \text{on } \Gamma_N^f. \end{aligned} \quad (\text{A.3})$$

4. Update the pressure field $p^{(k+1)}$ using $\mathbf{u}^{(k+1)}$ from step 2, $\psi^{(k+1)}$ from step 3 and the definition of $\psi^{(k+1)}$ (cf. (7)),

$$p^{(k+1)} = \psi^{(k+1)} + p^{(k)} - \mu \nabla \cdot \mathbf{u}^{(k+1)}. \quad (\text{A.4})$$

5. If not converged, set $(k + 1) \rightarrow (k)$ and go to step 2. If converged, the solution for the time t^{n+1} is identified with the latest iterates

$$\mathbf{u}^{n+1} = \mathbf{u}^{(k+1)}, \quad p^{n+1} = p^{(k+1)}, \quad \text{and} \quad \psi^{n+1} = \psi^{(k+1)}. \quad (\text{A.5})$$

A useful stopping criterion in step 5 is to control a norm of the pressure increment ψ . In comparison with the velocity, the pressure is more susceptible to large numerical oscillations in moving boundary flows.

APPENDIX B: FLUID-STRUCTURE COUPLING – GOVERNING EQUATIONS

A detailed derivation of the iterative coupling procedure presented in Section 4 is given in the following. The motivation is to show that the Robin boundary condition (48) naturally arises if the fluid–structure coupling is achieved by enforcing the interface velocity continuity (47) in a weak sense using the Nitsche method. For simplicity, we assume that all boundary conditions other than on the fluid–structure interface are fulfilled. Then, the weak form of the Navier–Stokes equations (1) reads

$$\begin{aligned} \rho^f (\dot{\mathbf{u}}, \mathbf{v})_{\Omega^f(t)} + \rho^f ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v})_{\Omega^f(t)} + (\boldsymbol{\sigma}(\mathbf{u}, p), \boldsymbol{\epsilon}(\mathbf{v}))_{\Omega^f(t)} \\ + (\nabla \cdot \mathbf{u}, q)_{\Omega^f(t)} - \left\langle \boldsymbol{\sigma}(\mathbf{u}, p) \mathbf{n}_t^f, \mathbf{v} \right\rangle_{\Gamma_I(t)} = 0, \end{aligned} \quad (\text{B.1})$$

where $\Omega^f(t)$ is used in order to emphasise the time dependence of the fluid domain. Adding the structure’s weak equilibrium (40) to (B.1), the following interface integrals arise and can be combined to

$$\int_{\Gamma_I(t)} \left(\boldsymbol{\sigma}(\mathbf{u}, p) \mathbf{n}_t^f \right) \cdot \mathbf{v} d\Gamma_I + \int_{\Omega_0^s} [\bar{T}]_-^+ \cdot \mathbf{y} d\Omega_0^s = \left\langle \boldsymbol{\sigma}(\mathbf{u}, p) \mathbf{n}_t^f, \mathbf{v} - \mathbf{y} \right\rangle_{\Gamma_I(t)} \quad (\text{B.2})$$

using the traction continuity condition (47). This leads to the global fluid–structure weak form

$$\begin{aligned} \rho^f (\dot{\mathbf{u}}, \mathbf{v})_{\Omega^f(t)} + \rho^f ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v})_{\Omega^f(t)} + (\sigma(\mathbf{u}, p), \boldsymbol{\epsilon}(\mathbf{v}))_{\Omega^f(t)} + (\nabla \cdot \mathbf{u}, q)_{\Omega^f(t)} \\ + \rho^s (A \ddot{\mathbf{x}}, \mathbf{y})_{\Omega_0^s} + (\widetilde{N}^{11}(\mathbf{x}), \alpha_{11}(\mathbf{y}))_{\Omega_0^s} + (\widetilde{M}^{11}(\mathbf{x}), \beta_{11}(\mathbf{y}))_{\Omega_0^s} \\ - \langle \sigma(\mathbf{u}, p) \mathbf{n}_t^f, \mathbf{v} - \mathbf{y} \rangle_{\Gamma_I(t)} = 0. \end{aligned}$$

Now, according to Nitsche [25], the following two consistent interface terms are added

$$- \langle \sigma(\mathbf{v}, q) \mathbf{n}_t^f, \mathbf{u} - \dot{\mathbf{x}} \rangle_{\Gamma_I(t)} \quad \text{and} \quad + \frac{\gamma\mu}{h} \langle \mathbf{u} - \dot{\mathbf{x}}, \mathbf{v} - \mathbf{y} \rangle_{\Gamma_I(t)}, \quad (\text{B.3})$$

with the penalty parameter $\gamma > 0$, the fluid viscosity μ and the characteristic fluid cell size h . Then, the globally coupled weak form becomes

$$\begin{aligned} \rho^f (\dot{\mathbf{u}}, \mathbf{v})_{\Omega^f(t)} + \rho^f ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v})_{\Omega^f(t)} + (\sigma(\mathbf{u}, p), \boldsymbol{\epsilon}(\mathbf{v}))_{\Omega^f(t)} + (\nabla \cdot \mathbf{u}, q)_{\Omega^f(t)} \\ + \rho^s (A \ddot{\mathbf{x}}, \mathbf{y})_{\Omega_0^s} + (\widetilde{N}^{11}(\mathbf{x}), \alpha_{11}(\mathbf{y}))_{\Omega_0^s} + (\widetilde{M}^{11}(\mathbf{x}), \beta_{11}(\mathbf{y}))_{\Omega_0^s} \\ - \langle \sigma(\mathbf{u}, p) \mathbf{n}_t^f, \mathbf{v} - \mathbf{y} \rangle_{\Gamma_I(t)} - \langle \sigma(\mathbf{v}, q) \mathbf{n}_t^f, \mathbf{u} - \dot{\mathbf{x}} \rangle_{\Gamma_I(t)} + \frac{\gamma\mu}{h} \langle \mathbf{u} - \dot{\mathbf{x}}, \mathbf{v} - \mathbf{y} \rangle_{\Gamma_I(t)} = 0. \end{aligned} \quad (\text{B.4})$$

The next step is a spatial decomposition, see, for example, [55], in which the test functions are chosen as either $\mathbf{y} = \mathbf{0}$ or $(\mathbf{v}, q) = (\mathbf{0}, 0)$ in order to obtain the local problems for the fluid or the structure side, respectively. The fluid subproblem now reads

$$\begin{aligned} \rho^f (\dot{\mathbf{u}}, \mathbf{v})_{\Omega^f(t)} + \rho^f ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v})_{\Omega^f(t)} + (\sigma(\mathbf{u}, p), \boldsymbol{\epsilon}(\mathbf{v}))_{\Omega^f(t)} + (\nabla \cdot \mathbf{u}, q)_{\Omega^f(t)} \\ - \langle \sigma(\mathbf{u}, p) \mathbf{n}_t^f, \mathbf{v} \rangle_{\Gamma_I(t)} - \langle \sigma(\mathbf{v}, q) \mathbf{n}_t^f, \mathbf{u} \rangle_{\Gamma_I(t)} + \frac{\gamma\mu}{h} \langle \mathbf{u} - \dot{\mathbf{x}}, \mathbf{v} \rangle_{\Gamma_I(t)} = 0. \end{aligned} \quad (\text{B.5})$$

This expression is exactly the weak form (B.1) with the interface condition $\mathbf{u} = \dot{\mathbf{x}}$ employed by means of the Nitsche's method. The other choice of test functions yields the weak form for the structure

$$\begin{aligned} \rho^s (A \ddot{\mathbf{x}}, \mathbf{y})_{\Omega_0^s} + (\widetilde{N}^{11}(\mathbf{x}), \alpha_{11}(\mathbf{y}))_{\Omega_0^s} + (\widetilde{M}^{11}(\mathbf{x}), \beta_{11}(\mathbf{y}))_{\Omega_0^s} \\ + \int_{\Gamma_I(t)} \left(\sigma(\mathbf{u}, p) \mathbf{n}_t^f + \frac{\gamma\mu}{h} (\dot{\mathbf{x}} - \mathbf{u}) \right) \cdot \mathbf{y} d\Gamma_I = 0. \end{aligned} \quad (\text{B.6})$$

The last integral of this expression combines the interface velocity continuity and the interface equilibrium (47). It gives rise to the Robin boundary condition (48) for the strong form of (B.6).

ACKNOWLEDGEMENT

The support of the EPSRC through grant # EP/G008531/1 is gratefully acknowledged.

REFERENCES

1. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
2. Baiges J, Codina R, Coppola-Owen H. The Fixed-Mesh ALE approach for the numerical simulation of floating solids. *International Journal for Numerical Methods in Fluids* 2011; **67**:1004–1023.
3. Gerstenberger A, Wall WA. An eXtended Finite Element Method/Lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:1699–1714.
4. Rüberg T, Cirak F. An immersed finite element method with integral equation correction. *International Journal for Numerical Methods in Engineering* 2010; **86**:93–114.
5. Rüberg T, Cirak F. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering* 2011; **209–212**:266–283.

6. Wang H, Chessa J, Liu WK, Belytschko T. The immersed/fictitious element method for fluid–structure interaction: volumetric consistency, compressibility and thin members. *International Journal for Numerical Methods in Engineering* 2008; **74**(1):32–55.
7. Burman E, Fernández MA. Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:766–784.
8. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4135–4195.
9. Buffa A, de Falco C, Sangalli G. Isogeometric analysis: new stable elements for the Stokes equation. *International Journal for Numerical Methods in Fluids* 2010; **65**:1407–1422.
10. Evans JA, Hughes TJR. Isogeometric divergence-conforming b-splines for the steady Navier–Stokes equations. *Technical Report 12-15*, ICES, 2012.
11. Cirak F, Long Q. Subdivision shells with exact boundary control and non-manifold geometry. *International Journal for Numerical Methods in Engineering* 2010; **88**:897–923.
12. Cirak F, Ortiz M. Fully C^1 -conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering* 2001; **51**:813–833.
13. Cirak F, Ortiz M, Schröder P. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering* 2000; **47**:2039–2072.
14. Cirak F, Deiterding R, Mauch SP. Large-scale fluid–structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations. *Computers & Structures* 2007; **85**:1049–1065.
15. Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4506–4527.
16. Dettmer W, Perić D. A computational framework for fluid–structure interaction: finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5754–5779.
17. Förster C, Wall WA, Ramm E. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1278–1293.
18. Joosten MM, Dettmer WG, Perić D. Analysis of the block Gauss–Seidel solution procedure for a strongly coupled model problem with reference to fluid–structure interaction. *International Journal For Numerical Methods In Engineering* 2009; **78**:757–778.
19. Küttler U, Wall WA. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics* 2008; **43**:61–72.
20. Farhat C, Lesoinne M, LeTallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**:95–114.
21. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
22. Temam R. Sur l’approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires (II). *Archive for Rational Mechanics and Analysis* 1969; **33**:377–385.
23. Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:6011–6045.
24. Quarteroni A, Saleri F, Veneziani A. Factorization methods for the numerical approximation of Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2000; **188**:505–526.
25. Nitsche J. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, Vol. 36. Springer: Berlin/Heidelberg, 1971; 9–15.
26. de Boor C. *A Practical Guide to Splines*. Springer: New York, 2001.
27. Rogers DF. *An Introduction to NURBS: With a Historical Perspective*. Academic Press: San Diego, 2001.
28. Ern A, Guermond JL. *Theory and Practice of Finite Elements*. Springer: New York, 2004.
29. Bressan A, Sangalli G. Isogeometric discretizations of the stokes problem: stability analysis by the macroelement technique. *IMA Journal of Numerical Analysis* 2013; **33**:629–651.
30. Karagiozis K, Kamakoti R, Cirak F, Pantano C. A computational study of supersonic disk-gap-band parachutes using Large-Eddy Simulation coupled to a structural membrane. *Journal of Fluids and Structures* 2011; **27**:175–192.
31. Cirak F, Cummings JC. Generic programming techniques for parallelizing and extending procedural finite element programs. *Engineering with Computers* 2008; **24**:1–16.
32. Gueziec A, Hummel R. Exploiting triangulated surface extraction using tetrahedral decomposition. *IEEE Transactions on Visualization and Computer Graphics* 1995; **1**:328–342.
33. Höllig K, Reif U, Wipper J. Weighted extended B-spline approximation of dirichlet problems. *SIAM Journal on Numerical Analysis* 2001; **39**:442–462.
34. Strickwerda JC. *Finite Difference Schemes and Partial Differential Equations*, 2nd ed. SIAM: Philadelphia, 2004.
35. Long Q, Bornemann PB, Cirak F. Shear-flexible subdivision shells. *International Journal for Numerical Methods in Engineering* 2012; **90**:1549–1577.
36. Simo JC, Fox DD. On a stress resultant geometrically exact shell model. Part I: formulation and optimal parameterization. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**:267–304.
37. Holzapfel GA. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley & Sons Ltd.: Chichester, UK, 2000.

38. Lewin GC, Haj-Hariri H. Modelling thrust generation of a two-dimensional heaving airfoil in a viscous flow. *Journal of Fluid Mechanics* 2003; **492**:339–362.
39. Wang ZJ. Vortex shedding and frequency selection in flapping flight. *Journal of Fluid Mechanics* 2000; **410**:323–341.
40. Fung YC. *Introduction to the Theory of Aeroelasticity*. Dover: New York, 2002.
41. Henderson RD. Nonlinear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics* 1997; **352**:65–112.
42. Shiels D, Leonard A, Roshko R. Flow-induces vibration of a circular cylinder at limiting structural parameters. *Journal of Fluids and Structures* 2001; **15**:3–21.
43. Alben S, Shelley M, Zhang J. Drag reduction through self-similar bending of a flexible body. *Nature* 2002; **420**:479–481.
44. Zhu L, Peskin CS. Drag of a flexible fibre in a 2D moving visous fluid. *Computers & Fluids* 2007; **36**:398–406.
45. Yin B, Luo H. Effect of wing inertia on hovering performance of flexible flapping wings. *Physics of Fluids* 2010; **22**:111902.
46. Nolan GR. Aerodynamics of vortex lift in insect flight. *Ph.D. Thesis*, University of Cambridge, Department of Zoology, Cambridge, UK, 2004.
47. Hunt JCR, Wray A, Moin P. Eddies, streams, and convergence zones in turbulent flows. *Studying Turbulence Using Numerical Simulation Databases*, vol. 2, Stanford, 1988; 193–208.
48. Ellington CP. The aerodynamics of hovering insect flight. V. A vortex theory. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 1984; **305**:115–144.
49. Sistek J, Cirak F. Parallel performance of iterative solution techniques for solving the incompressible Navier–Stokes equation with pressure-correction 2013. University of Cambridge, Department of Engineering, Technical report.
50. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
51. Bornemann PB, Cirak F. A subdivision-based implementation of the hierarchical b-spline finite element method. *Computer Methods in Applied Mechanics and Engineering* 2013; **253**:584–598.
52. Kraft R. Adaptive and linearly independent multilevel b-splines. In *Surface Fitting and Multiresolution Methods*, Méhauté AL, Rabut C, Schumaker LL (eds). Vanderbilt University Press: Nashville, Tennessee, USA, 1997; 209–218.
53. Schillinger D, Rank E. An unfitted hp-adaptive finite element method based on hierarchical b-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:3358–3380.
54. Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:229–263.
55. LeTallec P, Mouro J. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3039–3067.