Swansea University
Prifysgol Abertawe

Cronfa
Setting Research Free

Cronfa - Swansea University Open Access Repository

This is an author produced version of a paper published in :
Computer Methods in Applied Mechanics and Engineering

Cronfa URL for this paper:
http://cronfa.swan.ac.uk/Record/cronfa31832

Paper:
Kadapa, C., Dettmer, W. & Peri, D. (2017). A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid–solid contact. *Computer Methods in Applied Mechanics and Engineering*
http://dx.doi.org/10.1016/j.cma.2017.01.024

This article is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Authors are personally responsible for adhering to publisher restrictions or conditions. When uploading content they are required to comply with their publisher agreement and the SHERPA RoMEO database to judge whether or not it is copyright safe to add this version of the paper to this repository.
http://www.swansea.ac.uk/iss/researchsupport/cronfa-support/

Accepted Manuscript

A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid–solid contact

C. Kadapa, W.G. Dettmer, D. Perić

PII: S0045-7825(16)31370-6
DOI: http://dx.doi.org/10.1016/j.cma.2017.01.024
Reference: CMA 11309

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date : 18 October 2016
Revised date : 20 January 2017
Accepted date : 23 January 2017

Please cite this article as: C. Kadapa, et al., A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid–solid contact, *Comput. Methods Appl. Mech. Engrg.* (2017), http://dx.doi.org/10.1016/j.cma.2017.01.024.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid-solid contact

C. Kadapa*, W. G. Dettmer, D. Perić

Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Fabian Way, Swansea, SA1 8EN, Wales, UK.

Abstract

An accurate, efficient and robust numerical scheme is presented for the simulation of the interaction between flexibly-supported rigid bodies and incompressible fluid flow with topology changes and solid-solid contact. The solution of the incompressible Navier-Stokes equations is approximated by employing a stabilised formulation on Cartesian grids discretised with hierarchical b-splines. The solid is modelled as a rigid body and represented by linear segments along its boundary. Kinematic conditions along the fluid-rigid body interface are enforced weakly using Nitsche's method, while ghost penalty operators are employed to avoid excessive ill-conditioning of the system matrix arising from small cut cells. A staggered scheme is used for resolving the coupled fluid-rigid body interaction. The contact between moving or moving and fixed solid bodies is modelled with Lagrange multipliers. The excellent performance and wide range of applicability of the proposed scheme are demonstrated in a number of benchmark tests as well as industrially relevant model problems. The examples cover the galloping phenomena, particulate flow, hydraulic check valves and a model turbine.

Keywords: Fluid-rigid body interaction; Hierarchical b-splines; Immersed boundary methods; Staggered scheme; Particulate flows; Check valve.

1. Introduction

Fluid-structure interaction (FSI) is frequently encountered in science and engineering. Developing robust and accurate numerical schemes for the simulation of FSI phenomena has been in the focus of research in computational engineering for several decades. Despite some considerable developments in this field, see [1–3], a substantial amount of research is still required in order to extend the methodology to the range and scale of problems encountered in industrial practice. This is due to the numerous challenges that the developers of FSI solvers face, including: (a) large structural deformations, (b) topological changes of the

*Corresponding author
Email address: c.kadapa@swansea.ac.uk (C. Kadapa)

ACCEPTED MANUSCRIPT

fluid domain, (c) added-mass effects and (d) computational efficiency. Therefore the range of applicability of existing numerical schemes and commercial software tools is generally limited. In fact, many important real world engineering problems, for instance, various types of valves, pumps, turbines and blood flow through heart and arteries, require that all these challenges are addressed successfully.

A number of researchers have developed numerical schemes for FSI based on the now well established arbitrary Lagrangian-Eulerian (ALE) formulation using body-fitted meshes, see [4–11] and references therein. Despite the significant research effort invested, such schemes have certain inherent disadvantages that limit their applicability to complex FSI problems. Namely, FSI schemes based on ALE require sophisticated mesh-moving and/or remeshing algorithms in order to capture large structural deformations. These algorithms are not only complex to implement but also introduce additional numerical errors during the data transfer from one mesh to the other, when remeshing is needed during the solution process. In situations involving complex geometries and frequent topological changes, typically encountered in valves, pumps, mixers etc, use of body-fitted ALE bases FSI schemes, although possible in some cases using complex mesh generation strategies (see [12–14]), becomes impractical for a majority of such problems.

In order to overcome the difficulties encountered by ALE formulations in problems with large structural deformations and topological changes, numerical methods based on immersed or unfitted strategies have been explored. Immersed methods offer important advantages over body-fitted ALE schemes, namely, (a) the fluid grid does not have to align with the boundary of the solid and (b) the formulation naturally allows for large structural deformations and topological changes. Several different immersed boundary methods have been proposed, based on different strategies for the imposition of the interface conditions between the fluid and solid phases, such as: immersed boundary method (IBM) of Peskin [15], immersed interface method by LeVeque et al. [16, 17], immersed structural potential method (ISPM) by Gil et al. [18], immersed finite element method by [19, 20], immersed b-spline methods by Rüberg and Cirak [21, 22] and immersogeometric methods [23–27]. The latter relate to the innovative field of isogeometric modelling which originates from the work by Hughes and co-workers [28]. However, many of these schemes lack local refinement capabilities, a feature that is essential in order to reduce the computational cost of the simulation. In addition, in the original IBM and many of its variants, the interface conditions are enforced weakly using virtual springs which limits the time steps to very small values in order to maintain stability.

With the aim of addressing the above concerns, our recent research effort has been directed at the development of a computational framework based on an immersed method, which is robust and efficient, and suitable for the simulation of complex industrial FSI problems. Kadapa et al. [29] present a fully-coupled numerical scheme for the interaction of thin flexible structures with viscous incompressible fluid flow based on hierarchical b-splines and a fictitious domain method (FDM) employing Lagrange multipliers. The present work presents a new numerical framework motivated by the recent developments in unfitted methods, also known as CutFEM, by Burman et al. [30–33]. A preliminary study of this framework is

ACCEPTED MANUSCRIPT

presented in Dettmer et al. [34], which focuses on the accuracy and robustness of the scheme in the context of the Laplace equation and the steady incompressible Navier-Stokes equations on fixed domains immersed in Cartesian b-spline grids. The methodology is essentially based on the deactivation of the degrees of freedom which do not possess any support in the physical domain. The integration of the cut cells is restricted to the active part of the cells and the immersed boundary conditions are imposed by applying Nitsche's method. It is known and demonstrated clearly in [34] that the presence of small cut cells leads to excessive system matrix condition numbers. So-called ghost penalty terms, originally developed by Hansbo and Burman in [30–32] and references therein, are employed to alleviate this effect. Thus, the present article presents the extension of the methodology proposed in [34] to the simulation of fluid-rigid body interaction.

For the temporal discretisation, we employ the generalised-α method for first and for second order problems as proposed by Jansen et al. in [35] and Chung and Hulbert in [36], respectively. The staggered scheme presented by Dettmer and Perić in [9] is used to resolve the interaction between the fluid flow and the flexibly-supported rigid bodies. Thus, the methodology including the weakly coupled FSI solution scheme is based throughout on the second order accurate time integration procedures. Normal contact between moving rigid bodies or moving bodies and the fixed fluid boundary is accounted for by employing Lagrange multipliers in the solid solver. This technique is standard in implicit computational contact mechanics and is described, for instance, in [37].

Summarising, the key ingredients of the present work are:

- hierarchical b-spline grid for the discretisation of the fluid flow,
- SUPG/PSPG formulation for the incompressible Navier-Stokes,
- Nitsche's method for imposing boundary as well as interface conditions,
- ghost penalty terms for the stabilisation of cut cells,
- generalised-α method for time integration,
- staggered solution scheme to solve the coupling between solid and fluid,
- Lagrange multipliers for modelling solid-solid contact.

The remainder of this article is organised as follows. In Section 2, we briefly introduce the b-spline grids with hierarchical refinement. In Section 3, we describe the aspects of the adopted immersed methodology. The governing equations relevant to the present work are given in Section 4. The adopted time integration schemes are described in Section 5. In Section 6 we present the staggered solution scheme for coupled fluid-structure interaction and comment on the modelling of solid-solid contact. The detailed finite element formulation for the fluid flow is given in Section 7. The performance of the proposed scheme is demonstrated using several numerical examples in Section 8. The summary and the conclusions are provided in Section 9.

ACCEPTED MANUSCRIPT

2. B-Spline grids with hierarchical refinement

Hierarchical b-splines have some attractive mathematical properties and their tensor-product structure is very convenient for computer programming. As such, they offer many advantages over standard Lagrange polynomials when used for the discretisation of the Cartesian grid in immersed boundary methods. In particular, the hierarchical refinement property allows for local grid refinement, which is very useful for problems exhibiting high gradients in solution variables, typically encountered around interfaces and intricate boundary shapes. This hierarchical refinement capability of the b-splines helps to improve the computational efficiency.

The theory and programming aspects of hierarchical b-splines are well established. In this section we present only basic properties of b-splines and their hierarchical refinement. For detailed information, we refer to Piegl and Tiller [38] and Rogers [39].

2.1. Univariate b-splines

B-splines are piecewise-continuous polynomial functions. For a given knot vector $\boldsymbol{\Xi} = \{\xi_0, \ldots, \xi_{n+a+1}\}$ and degree of polynomial a, the univariate b-spline functions are evaluated from the following recurrence relations,

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$N_{i,a}(\xi) = \frac{\xi - \xi_i}{\xi_{i+a} - \xi_i} N_{i,a-1}(\xi) + \frac{\xi_{i+a+1} - \xi}{\xi_{i+a+1} - \xi_{i+1}} N_{i+1,a-1}(\xi) \, . \tag{2}$$

Fig. 1 illustrates univariate uniform b-spline basis functions of different orders. A b-spline function $N_{i,a}$ of degree a is nonzero in knot span $[\xi_i, \xi_{i+a+1}]$ and is zero otherwise; and is at least $\mathcal{C}^{a-1}$ continuous in that knot span. Uniform b-splines are considered throughout this paper, i.e. when there is no local refinement, $\Delta\xi = \xi_{i+1} - \xi_i$ is constant throughout the domain.

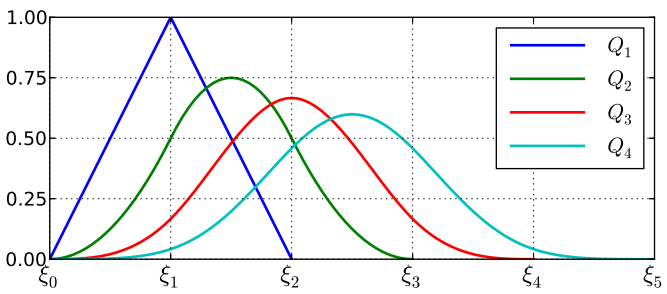


Figure 1: Univariate b-spline basis functions for different orders. Q_1, Q_2, Q_3 and Q_4 refer to linear, quadratic, cubic and quartic b-splines, respectively.

ACCEPTED MANUSCRIPT

2.2. B-splines in higher dimensions

One of the main motivations behind using b-splines for spatial discretisation is the ease with which they can be extended to higher dimensions. This is achieved by using the tensor product nature of b-splines. With $\mathbf{N}_\xi$, $\mathbf{N}_\eta$ and $\mathbf{N}_\zeta$ as the univariate b-spline functions in ξ, η, and ζ parametric directions, respectively, the multivariate b-spline basis functions in two- and three-dimensions are given as,

$$\mathbf{N}(\xi,\eta) = \mathbf{N}_\xi \otimes \mathbf{N}_\eta \qquad \text{in} \quad 2\text{D} \tag{3}$$

$$\mathbf{N}(\xi,\eta,\zeta) = \mathbf{N}_\xi \otimes \mathbf{N}_\eta \otimes \mathbf{N}_\zeta \qquad \text{in} \quad 3\text{D} \ . \tag{4}$$

2.3. Hierarchical refinement

The local refinement strategy of b-splines is based on their subdivision property, also known widely as *two-scale* relation. According to this property, a single b-spline function can be written as a sum of scaled and translated copies of itself. For a general b-spline function N_a, the *two-scale* relation is written as,

$$N_a(\xi) = \sum_{i=0}^{a+1} \alpha_i N_a(2\xi - i) \ , \tag{5}$$

where α_i are functions of Binomial coefficients, given as,

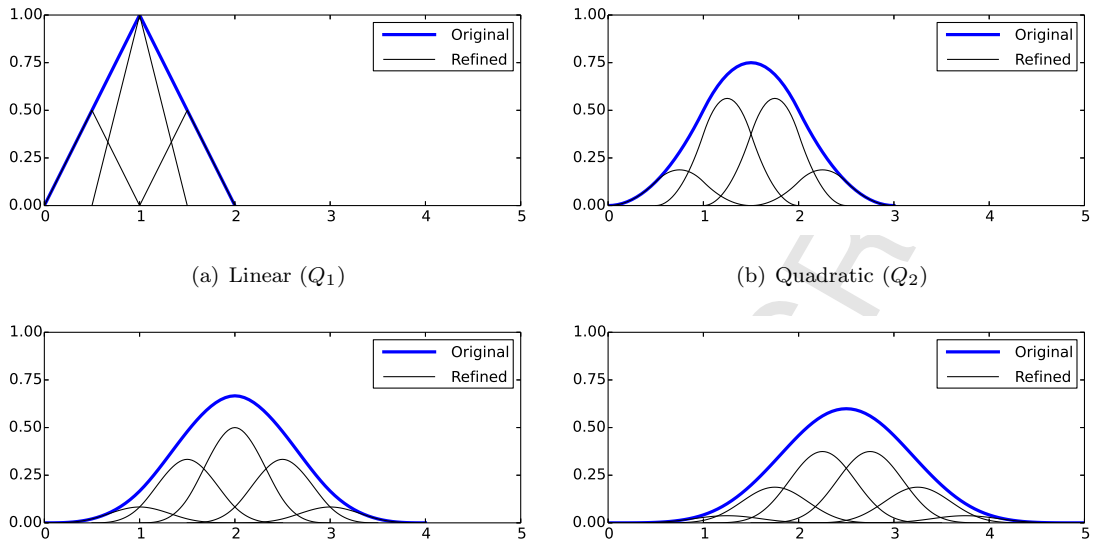
$$\alpha_i = \frac{1}{2^a} \begin{pmatrix} a+1 \\ i \end{pmatrix} \ . \tag{6}$$

In the context of hierarchical refinement, this property can be restated as: a b-spline function on a knot vector with knot span $\Delta\xi$ can be evaluated as a linear combination of b-spline functions defined on a knot vector with knot span $\Delta\xi/2$. That is, b-spline basis functions $\mathbf{N}_k(\xi)$ at level k can be written as a linear combination of b-spline basis functions $\mathbf{N}_{k+1}(\xi)$ at level k+1. This is expressed as

$$\mathbf{N}_k(\xi) = \mathbf{S}\, \mathbf{N}_{k+1}(\xi) \ , \tag{7}$$

where $\mathbf{S}$ is the subdivision matrix which contains the coefficients α_i from Eq. (5). Fig. 2 illustrates the schematic representation of *two-scale* relation for linear(Q_1), quadratic(Q_2), cubic(Q_3) and quartic(Q_4) b-splines.

ACCEPTED MANUSCRIPT



(a) Linear (Q_1) (b) Quadratic (Q_2)

(c) Cubic (Q_3) (d) Quartic (Q_4)

Figure 2: Two-scale relation of the b-spline functions.

The hierarchical refinement of b-splines can be implemented efficiently by using *Tree* data structures, see [40, 41], and the concept of *templates* in the programming language C++, see [42]. The algorithms pertaining to hierarchical refinement of b-splines are presented in Bornemann and Cirak [43] and Schillinger et al. [44]. In this work we have followed [44].

3. The immersed boundary method

The following subsections describe the key aspects of the immersed methodology adopted in this work.

3.1. B-spline grid with hierarchical refinement

The Cartesian grids employed in this work allow for targeted local refinement. In areas with complex local geometries or small physical features, the same refinement procedure can be applied to successive levels of b-spline basis functions, thereby allowing for arbitrarily small resolutions. If refinement is applied during run time, the mathematical properties of the b-spline based discretisation allow for the exact data transfer from the coarse to the refined mesh. The computational cost for mesh generation, local refinement and data transfer is negligible. This is a powerful setting for an immersed method and allows to employ efficient discretisations without any unnecessary degrees of freedom.

3.2. Cut finite element method

The degrees of freedom which are supported entirely outside of the fluid domain are deactivated and not included in the solution procedure. For problems with large inactive areas of the background mesh,

ACCEPTED MANUSCRIPT

this adds significantly to the efficiency of the methodology. Those degrees of freedom which possess support partly inside and partly outside of the fluid domain remain active and are solved for. This is schematically displayed in Figure 3. Only the active part of the cut finite elements is integrated. In the presence of small cut cells, some degrees of freedom possess very little support inside the active domain. It is well known and, for instance, demonstrated clearly in [34, 45] that this leads to system matrices with excessive condition numbers. While this may not be a problem for direct linear equation solvers, it can be detrimental to the performance of iterative solvers. Therefore, it is common to apply some form of cut cell stabilisation. In this work, we apply the ghost penalty terms developed by Burman, Hansbo and co-workers, see [30–33] and references therein. These penalty terms are used to enforce a suitable amount of continuity of the solution across the inter-element boundaries between the cut elements and the interior of the active domain. For a comprehensive study of the performance of this methodology in the context of steady-state problems we refer to [34]. The ghost penalty terms are presented in detail in Section 7.

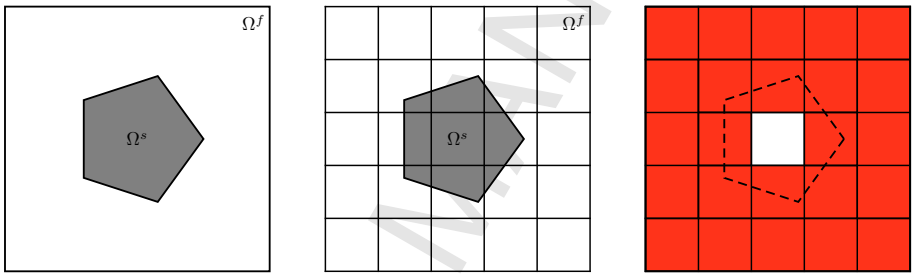


Figure 3: Immersed methodology: (a) problem geometry with fluid domain Ω^f and rigid body Ω^s, (b) discretisation with 5×5 elements, (c) elements belonging to domain Ω^f.

3.3. Modelling of immersed solids

The geometry of immersed solids can be represented using parametric equations or level sets for simple geometries. However, the use of parametric equations is not a viable option for problems involving complex geometries and intricate shapes frequently encountered in real-world industrial problems. Even though level set data can be generated for complex geometries it requires sophisticated algorithms and, moreover, such procedures introduce additional numerical errors in geometric representations. As the main motivation of the present work is to develop a numerical scheme that can be easily applied to complex geometries encountered in industrial problems, we use a discrete representation of the geometry that can be generated by using the standard meshing tools available in the engineering community. In 2D, the boundary of the immersed solid is represented as a set of straight edges and in 3D the surface of the immersed solid is represented as a set of 3-noded triangles. The methodology can be extended to NURBS-based descriptions of the immersed boundary which may be obtained directly from computer-aided design tools, similar to the methodology

ACCEPTED MANUSCRIPT

proposed by Hsu et al. [26].

3.4. Integration of cut cells

The inaccurate integration of cut cells results in suboptimal convergence rates, as demonstrated in [34]. For fluid-structure interaction problems, it also leads to inaccurate interface forces and hence the poor quality of results. A widely used method for the integration of cut cells is sub-triangulation or tessellation, which is achieved by subdividing the active part of the cell into triangles in 2D (and tetrahedrons in 3D) and subsequently applying a standard quadrature rule to each sub-triangle. The foremost advantage of using such a technique is that the cut cells can be integrated exactly. This technique has been followed in [22, 46, 47]. However, the disadvantage is that it is difficult to generate sub-triangulation if the immersed boundaries are curved. Moreover, the procedure is relatively complicated for 3D problems. For example, for immersed surfaces discretised with triangles, one has to use complex constrained Delaunay tetrahedralization algorithms. An alternative to this approach is to use the level-set representation of geometry, as in Burman et al. [46] and Rüberg and Cirak [22], and to use this information for sub-triangulation of cut-cells. However, acquiring a level-set representation of geometry for real world engineering problems itself is a difficult task and also generating subtriangulations from level-set data introduces discretisation errors. Alternatively, adaptive procedures based on the hierarchical application of Gauss-quadrature may be used for the integration of the cut cells, see for instance [44]. An efficient adaptive integration methodology based on this approach is presented in [34].

In the two-dimensional examples in Section 8 the immersed boundary is represented by straight line segments and we employ sub-triangulation for the integration of all cut cells. The number of Gauss points in a sub-triangle depends on the order of the b-spline basis functions.

4. Governing equations

4.1. Incompressible Navier-Stokes

The Navier-Stokes equations for modelling laminar, viscous and incompressible fluid are given as,

$$\rho^f \frac{\partial \mathbf{v}^f}{\partial t} + \rho^f (\mathbf{v}^f \cdot \nabla)\mathbf{v}^f - \mu^f \Delta \mathbf{v}^f + \nabla p = \mathbf{g}^f \qquad \text{in} \quad \Omega^f \tag{8a}$$

$$\nabla \cdot \mathbf{v}^f = 0 \qquad \text{in} \quad \Omega^f \tag{8b}$$

$$\mathbf{v}^f = \bar{\mathbf{v}}^f \qquad \text{on} \quad \Gamma^f_{\mathrm{D}} \tag{8c}$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f = \bar{\mathbf{t}}^f \qquad \text{on} \quad \Gamma^f_{\mathrm{N}}\,, \tag{8d}$$

where $\mathbf{v}^f$ is the velocity of the fluid, p is the fluid pressure, ρ^f is the density of the fluid, μ^f is the viscosity of the fluid, $\mathbf{g}^f$ is the body force, $\boldsymbol{\sigma}^f = \mu \nabla \mathbf{v}^f - p\,\mathbf{I}$, is the stress tensor, $\mathbf{n}^f$ is the unit outward normal on the boundary Γ^f of the fluid domain Ω^f, $\bar{\mathbf{v}}^f$ is the prescribed velocity on the Dirichlet boundary Γ^f_{D} and $\bar{\mathbf{t}}^f$ is the prescribed traction on the Neumann boundary Γ^f_{N}. Here, $\Gamma^f_{\mathrm{D}} \bigcup \Gamma^f_{\mathrm{N}} = \Gamma^f$ and $\Gamma^f_{\mathrm{D}} \bigcap \Gamma^f_{\mathrm{N}} = \emptyset$.

ACCEPTED MANUSCRIPT

4.2. Rigid-body dynamics

The equations governing the dynamics of rigid bodies considered in this work can be summarized as

$$\mathbf{M}\,\mathbf{a}^s + \mathbf{C}\,\mathbf{v}^s + \mathbf{K}\,\mathbf{d}^s = \mathbf{F}^s\,, \tag{9}$$

where $\mathbf{d}^s$, $\mathbf{v}^s = \mathrm{d}\mathbf{d}^s/\mathrm{d}t$ and $\mathbf{a}^s = \mathrm{d}^2\mathbf{d}^s/\mathrm{d}t^2$ are the displacement, velocity and acceleration vectors of the rigid body, respectively, $\mathbf{F}^s$ is the force acting on the rigid body and $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are mass, damping and stiffness matrices, respectively.

4.3. Interface constraints

For accurate simulation of FSI phenomena two important conditions have to be satisfied at the interface Γ^{f-s} between fluid and solid. The first is the kinematic constraint which requires that the fluid-solid interface moves at the same velocity as the boundary of the solid, while the second condition enforces the equilibrium of stresses along the interface. The two conditions can be written as

$$\mathbf{v}^f = \mathbf{v}^s \quad \text{on} \quad \Gamma^{f-s} \tag{10a}$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f + \boldsymbol{\sigma}^s \cdot \mathbf{n}^s = \mathbf{0} \quad \text{on} \quad \Gamma^{f-s} \tag{10b}$$

with $\mathbf{n}^s = -\mathbf{n}^f$. In the context of fluid-rigid body interaction Eq. (10b) can be expressed as

$$\mathbf{F}^f \;+\; \mathbf{F}^s \;=\; \mathbf{0}\,, \tag{11}$$

where,

$$\mathbf{F}^f \;=\; \int_{\Gamma^{f-s}} \boldsymbol{\sigma}^f \cdot \mathbf{n}^f \; d\Gamma^{f-s}\,. \tag{12}$$

If the rigid body possesses rotational degrees of freedom, Eqs. (9), (10a) and (11) need to be extended appropriately in a standard manner. For the two dimensional problems considered in Section 8 the incorporation of the rotational degree of freedom is straightforward.

5. Integration in time

In the present work, we use the second-order accurate and unconditionally stable generalised-α schemes for time integration of both the fluid and solid subsystems. For the detailed discussion of these schemes, we refer to the original articles by Jansen et al. [35] and Chung and Hulbert [36].

5.1. Generalised-α method for the fluid

Defining the acceleration $\mathbf{a}^f = \mathrm{d}\mathbf{v}^f/\mathrm{d}t$ and letting $\mathbf{v}^f_n$ and $\mathbf{a}^f_n$ denote, respectively, the fluid velocity and acceleration at time instant t_n, the system of equations for the generalised-α method for the fluid as

ACCEPTED MANUSCRIPT

proposed in [35] may be presented by

$$\mathbf{v}^f_{n+1} = \mathbf{v}^f_n + \Delta t \left((1-\gamma^f)\, \mathbf{a}^f_n + \gamma^f\, \mathbf{a}^f_{n+1} \right) \tag{13}$$

$$\mathbf{v}^f_{n+\alpha^f_f} = (1-\alpha^f_f)\, \mathbf{v}^f_n + \alpha^f_f\, \mathbf{v}^f_{n+1} \tag{14}$$

$$\mathbf{a}^f_{n+\alpha^f_m} = (1-\alpha^f_m)\, \mathbf{a}^f_n + \alpha^f_m\, \mathbf{a}^f_{n+1}\ . \tag{15}$$

Once the velocity $\mathbf{v}^f_{n+1}$ at time t_{n+1} is obtained the acceleration $\mathbf{a}^f_{n+1}$ can be computed from Eq. (13) as

$$\mathbf{a}^f_{n+1} = \frac{1}{\gamma^f\, \Delta t}\, (\mathbf{v}^f_{n+1} - \mathbf{v}^f_n) + \frac{\gamma^f - 1}{\gamma^f}\, \mathbf{a}^f_n\ . \tag{16}$$

The parameters are defined as

$$\alpha^f_m = \frac{1}{2}\, \frac{3-\rho^f_\infty}{1+\rho^f_\infty}, \qquad \alpha^f_f = \frac{1}{1+\rho^f_\infty}, \qquad \gamma^f = \frac{1}{2} + \alpha^f_m - \alpha^f_f\ , \tag{17}$$

where $\rho^f_\infty \in [0,1]$ is the user-defined spectral radius for an infinite time step and allows to choose the amount of numerical high-frequency damping in the scheme.

5.2. Generalised-α method for the solid

Similarly, with $\mathbf{d}^s_n$, $\mathbf{v}^s_n$ and $\mathbf{a}^s_n$ denoting, respectively, the displacement, velocity and acceleration at time t_n, the system of equations for the generalised-α method for the solid is given in [36] as

$$\mathbf{d}^s_{n+1} = \mathbf{d}^s_n + \Delta t \mathbf{v}^s_n + \Delta t^2 \left((\frac{1}{2} - \beta^s)\mathbf{a}^s_n + \beta^s \mathbf{a}^s_{n+1} \right) \tag{18}$$

$$\mathbf{v}^s_{n+1} = \mathbf{v}^s_n + \Delta t \left((1-\gamma^s)\mathbf{a}^s_n + \gamma^s \mathbf{a}^s_{n+1} \right) \tag{19}$$

$$\mathbf{d}^s_{n+\alpha^s_f} = (1-\alpha^s_f)\, \mathbf{d}^s_n + \alpha^s_f\, \mathbf{d}^s_{n+1} \tag{20}$$

$$\mathbf{v}^s_{n+\alpha^s_f} = (1-\alpha^s_f)\, \mathbf{v}^s_n + \alpha^s_f\, \mathbf{v}^s_{n+1} \tag{21}$$

$$\mathbf{a}^s_{n+\alpha^s_m} = (1-\alpha^s_m)\, \mathbf{a}^s_n + \alpha^s_m\, \mathbf{a}^s_{n+1} \tag{22}$$

$$\mathbf{F}^s_{n+\alpha^s_f} = (1-\alpha^s_f)\, \mathbf{F}^s_n + \alpha^s_f\, \mathbf{F}^s_{n+1}\ . \tag{23}$$

Choosing the velocity as the primary variable, Eqs. (18) and (19) can be rearranged as

$$\mathbf{v}^s_{n+1} = \frac{\gamma}{\beta \Delta t}(\mathbf{d}^s_{n+1} - \mathbf{d}^s_n) + \left(1 - \frac{\gamma}{\beta}\right)\, \mathbf{v}^s_n + \left(1 - \frac{\gamma}{2\beta}\right) \Delta t\, \mathbf{a}^s_n \tag{24}$$

$$\mathbf{a}^s_{n+1} = \frac{1}{\beta \Delta t^2}(\mathbf{d}^s_{n+1} - \mathbf{d}^s_n) - \frac{1}{\beta \Delta t}\, \mathbf{v}^s_n - \left(\frac{1}{2\beta} - 1\right)\, \mathbf{a}^s_n\ . \tag{25}$$

Chung and Hulbert [36] have proven that, in order for the scheme to be unconditionally stable and second-order accurate, the parameters require

$$\alpha^s_m = \frac{2-\rho^s_\infty}{\rho^s_\infty + 1}; \qquad \alpha^s_f = \frac{1}{\rho^s_\infty + 1} \tag{26}$$

$$\gamma^s = \frac{1}{2} + \alpha^s_m - \alpha^s_f; \qquad \beta^s = \frac{1}{4}(1 + \alpha^s_m - \alpha^s_f)^2\ , \tag{27}$$

ACCEPTED MANUSCRIPT

where $\rho^s_\infty \in [0,1]$ is the spectral radius, similar to ρ^f_∞, and allows to control the amount of numerical high-frequency damping in the scheme.

6. Resolution of fluid-structure interaction and solid-solid contact

The combined governing equations of the FSI problem, Eqs. (8), (9) and (10), can be solved by using either a strongly-coupled scheme, see [6–8, 29], or a weakly-coupled staggered scheme as proposed in [9, 48–53]. In this work we employ the force-predictor based staggered scheme proposed by Dettmer and Perić [9] which is proven to be second-order accurate. For the purpose of clarity and completeness, the pseudocode for the staggered scheme is shown in Algorithm 1. In the remainder of this article the user-defined parameter β, with $0 < \beta \leq 1$, is referred to as the *relaxation factor*. Problems which feature strong added mass effects require a small value of β. For the detailed discussion of this scheme, we refer to [9].

In the presence of contact between two moving rigid bodies or between one moving body and the fixed fluid boundary, Lagrange multipliers are incorporated into the solid solver in Step 2 of Algorithm 1 to avoid penetration. This is a standard technique in implicit computational contact mechanics and, for instance, described in [37]. In the context of the b-spline background mesh, numerical experimentation shows that, independently of the b-spline order, singularities in the fluid solver due to dynamic topology changes can be avoided by ensuring a minimum gap width in the contact zone which corresponds to two or three element diameters. Based on targeted hierarchical refinement of the background mesh, arbitrarily small gap widths can be achieved in an efficient manner.

The adopted second order accurate staggered time stepping procedure may cause artificial rebound effects when solid-solid contact first occurs. To alleviate this, the force predictor in Step 1 of Algorithm 1 is modified to $\mathbf{F}^{s^P}_{n+1} = \mathbf{F}_n$, if the rigid body has been in contact with another solid at t_n. As shown in Sections 8.6 and 8.7 in the simulation of hydraulic valves, this measure is sufficient to avoid any oscillatory behaviour when contact is activated or released.

For every time step:

1. predict force on the solid: $\mathbf{F}^{s^P}_{n+1} = 2\,\mathbf{F}_n - \mathbf{F}_{n-1}$
2. solve the solid problem for $\mathbf{d}^s_{n+1}$ and $\mathbf{v}^s_{n+1}$ using $\mathbf{F}^{s^P}_{n+1}$
3. reposition immersed solid, activate/deactivate fluid degrees of freedom, update cut cell data
4. solve the fluid problem to obtain force $\mathbf{F}^f_{n+1}$
5. average the interface force: $\mathbf{F}_{n+1} = -\beta\,\mathbf{F}^f_{n+1} + (1-\beta)\,\mathbf{F}^{s^P}_{n+1}$
6. proceed to next time step

Algorithm 1: Staggered scheme by Dettmer and Perić [9] for the present immersed methodology

ACCEPTED MANUSCRIPT

7. Formulation

In the context of the staggered scheme described in Section 6, the solution of Eq. (9) for rigid body dynamics is straightforward whereas the solution of fluid system requires special consideration. Hence, this section is restricted to the solution of the fluid problem.

7.1. Variational formulation for the fluid

The variational problem for the incompressible Navier-Stokes equations given by Eqs. (8) is stated as: Find the fluid velocity $\mathbf{v}^f \in \mathcal{S}_{\mathbf{v}}$ and the pressure $p \in \mathcal{S}_p$ such that for all weighting functions $\mathbf{w}^f \in \mathcal{V}_{\mathbf{v}}$ and $q \in \mathcal{V}_p$,

$$\begin{aligned} B^f_{\text{Gal}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) + B^f_{\text{Stab}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) + B^f_{\text{Nitsche}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) \\ + B^f_{\text{GP}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) = F^f_{\text{Gal}}(\{\mathbf{w}^f, q\}) \,, \end{aligned} \tag{28}$$

where $B^f_{\text{Gal}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\})$ and $F^f_{\text{Gal}}(\{\mathbf{w}^f, q\})$ consist of terms corresponding to the standard Galerkin formulation. $B^f_{\text{Stab}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\})$, $B^f_{\text{Nitsche}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\})$, and $B^f_{\text{GP}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\})$, respectively, are the terms corresponding to the SUPG/PSPG stabilisation, the Nitsche's method and the ghost penalty terms for cut cell stabilisation.

7.2. Standard Galerkin terms

The terms corresponding to the standard Galerkin formulation are given as

$$\begin{aligned} B^f_{\text{Gal}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) = \int_{\Omega^f} \mathbf{w}^f \cdot \rho^f \left(\frac{\partial \mathbf{v}^f}{\partial t} + \mathbf{v}^f \cdot \nabla \mathbf{v}^f \right) d\Omega^f \\ + \int_{\Omega^f} \mu \, \nabla \mathbf{w}^f : \nabla \mathbf{v}^f \, d\Omega^f - \int_{\Omega^f} \nabla \cdot \mathbf{w}^f \, p \, d\Omega^f + \int_{\Omega^f} q \, \nabla \cdot \mathbf{v}^f \, d\Omega^f \end{aligned} \tag{29}$$

$$F^f_{\text{Gal}}(\{\mathbf{w}^f, q\}) = \int_{\Omega^f} \mathbf{w}^f \cdot \mathbf{g}^f \, d\Omega^f + \int_{\Gamma^f_{\text{N}}} \mathbf{w}^f \cdot \bar{\mathbf{t}}^f \, d\Gamma \,. \tag{30}$$

7.3. SUPG/PSPG stabilisation

The Galerkin formulation of the incompressible Navier-Stokes equations, when used on its own, requires that the combination of spaces used for the discretisation of the velocity and the pressure fields satisfy the so-called LBB or inf-sup condition, see Brezzi and Fortin [54]. In the Lagrange family of elements, only a few combinations of basis functions have so far been proven to satisfy the inf-sup condition, for example, Taylor-Hood elements, see [55, 56] for further details. Inf-sup stable velocity-pressure combinations for Stokes flow in the context of isogeometric analysis have been studied in [57]. Rüberg and Cirak [21, 22] have applied inf-sup stable velocity-pressure b-spline combinations to fluid-structure interaction problems, while inf-sup stable displacement-pressure NURBS discretisations have been employed for nearly incompressible linear and non-linear elasticity by Kadapa et al. [58]. In both cases two different meshes are required, one for the

ACCEPTED MANUSCRIPT

velocities (or displacements) and another for the pressure, together with sophisticated data structures to compute basis functions on different meshes (see [21]). The complexity of this approach increases further if hierarchical refinement is used. Moreover, all inf-sup stable Galerkin methods require sophisticated linear solvers for the saddle-point systems. In order to avoid these difficulties, we use SUPG/PSPG stabilisation (see [59–61]), which allows for the use of the same discretisation for the velocity and the pressure. This technique also provides stability to the velocity field in convection dominated areas of the fluid domain.

The terms corresponding to SUPG/PSPG formulation are given as

$$B^f_{\text{Stab}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) = \sum_{e=1}^{nel} \int_{\Omega^{f^e}} \frac{1}{\rho^f} [\tau_{\text{SUPG}}\, \rho^f\, \mathbf{v}^f \cdot \nabla \mathbf{w}^f + \tau_{\text{PSPG}}\, \nabla q] \cdot \mathbf{r}_{\text{M}}\, d\Omega^f + \sum_{e=1}^{nel} \int_{\Omega^{f^e}} \tau_{\text{LSIC}}\, \rho^f\, (\nabla \cdot \mathbf{w}^f)\, (\nabla \cdot \mathbf{v}^f)\, d\Omega^f \ , \tag{31}$$

where $\mathbf{r}_{\text{M}}$ is the residual of the momentum equation and is given as

$$\mathbf{r}_{\text{M}} = \rho^f \frac{\partial \mathbf{v}^f}{\partial t} + \rho^f (\mathbf{v}^f \cdot \nabla \mathbf{v}^f) - \mu^f \Delta \mathbf{v}^f + \nabla p - \mathbf{g}^f \ . \tag{32}$$

The stabilisation parameters τ_{SUPG}, τ_{PSPG} and τ_{LSIC} are defined as

$$\tau_{\text{SUPG}} = \left(\mathbf{v}^f \cdot \mathbf{G} \mathbf{v}^f + C_I\, \nu^2\, \mathbf{G} : \mathbf{G}\right)^{-1/2} \tag{33}$$

$$\tau_{\text{PSPG}} = \tau_{\text{SUPG}} \tag{34}$$

$$\tau_{\text{LSIC}} = \left(\text{tr}\mathbf{G}\ \tau_{\text{SUPG}}\right)^{-1} \ , \tag{35}$$

where C_I is a positive constant independent of the mesh size. The quantity $\mathbf{G}$ is the element contravariant metric tensor. In the context of Cartesian b-spline grids with square elements, $\mathbf{G}$ can be written as

$$\mathbf{G} \ = \ \frac{4}{h^2}\, \mathbf{I} \, , \tag{36}$$

where $\mathbf{I}$ is the identity tensor and h is called as the characteristic element size. The exact values of C_I and h can be computed by solving local eigenvalue problems, see [62]. However, this procedure becomes quite expensive for dynamic FSI simulations. Therefore, in this work, we compute h from the area/volume of the corresponding element. For the evaluation of the stabilisation parameters in cut cells, h is evaluated, in the two dimensional setting, as

$$h = \sqrt{A'} \, , \tag{37}$$

where A' is the intersection of the element area and the physical domain Ω^f. In all examples in Section 8 we use $C_I = 4$.

7.4. Nitsche's method

The accurate imposition of the interface conditions is the critical aspect of finite element formulations based on the immersed boundary approach. However, the fact that the fluid grid does not align with the

ACCEPTED MANUSCRIPT

boundary of the immersed solid makes the task of imposing the interface conditions challenging. Recent developments [34, 63–67] have proven that Nitsche's method represents a robust and accurate technique for the imposition of boundary and interface conditions for finite element formulations based on b-splines. The terms corresponding to the Nitsche's method are given as

$$B^f_{\text{Nitsche}}(\{\mathbf{w}^f, q\}, \{\mathbf{v}^f, p\}) = \gamma_{\text{N}_1} \int_{\Gamma_\text{D}} \mathbf{w}^f \cdot (\mathbf{v}^f - \hat{\mathbf{v}})\, d\Gamma_\text{D} - \int_{\Gamma_\text{D}} \mathbf{w}^f \cdot (\boldsymbol{\sigma}(\{\mathbf{v}^f, p\}) \cdot \mathbf{n}^f)\, d\Gamma_\text{D} - \gamma_{\text{N}_2} \int_{\Gamma_\text{D}} (\boldsymbol{\sigma}(\{\mathbf{w}^f, q\}) \cdot \mathbf{n}^f) \cdot (\mathbf{v}^f - \hat{\mathbf{v}})\, d\Gamma_\text{D} \,, \tag{38}$$

where $\gamma_{\text{N}_1} \geq 0$ is a penalty parameter and γ_{N_2} allows to choose between the symmetric ($\gamma_{\text{N}_2} = 1$) and the unsymmetric ($\gamma_{\text{N}_2} = -1$) variants of Nitsche's method. Nitsche's method is applied at the interface between the fluid and moving rigid bodies as well as at all fixed Dirichlet boundaries of the fluid domain, thus, $\Gamma_\text{D} = \Gamma^f_\text{D} \bigcup \Gamma^{f-s}$. The vector $\hat{\mathbf{v}}$ is the imposed velocity which is given as $\hat{\mathbf{v}} = \bar{\mathbf{v}}^f$ on Γ^f_D and $\hat{\mathbf{v}} = \mathbf{v}^s$ on Γ^{f-s}. Following [34], the penalty-free ($\gamma_{\text{N}_1} = 0$) and unsymmetric version of Nitsche's method is used for imposing all Dirichlet boundary conditions in the examples in Section 8.

7.5. Cut cell stabilisation

If the active part of at least one cut element is small, then some basis functions possess little support inside the physical domain. This leads to large system matrix condition numbers, which makes the methodology unsuitable for the employment of iterative linear solvers and may cause solution inaccuracies in the context of direct solvers. The issue can be alleviated by applying the so-called ghost penalty terms along the inter-element boundaries of cut cells as shown schematically in Figure 4. These ghost penalty terms enforce a suitable amount of continuity across the boundaries between cut cells and the active cells in the interior of the physical domain. For detailed information we refer to [30–33]. A comprehensive study of the performance of ghost penalty terms in the context of an immersed method based on b-spline grids is presented in [34].

Following [33], the ghost penalty term for a scalar valued problem is defined as

$$g_s(w, v) := \sum_{j=1}^{a} \sum_{F \in \mathcal{F}} h^{2(j-1)+s} \int_F [D^j w][D^j v]\, dF \,, \tag{39}$$

where $[D^j(\bullet)]$ denotes the jump of the normal derivative of $(\bullet)$ of order j on the face F. The order of the basis function is denoted by a and h is element length scale. The parameter s is introduced for notational convenience. In the context of b-spline grids of polynomial order a, all derivatives except the one of order a are continuous across the inter-element boundaries and Eq. (39) reduces to

$$g_s(w, v) := \sum_{F \in \mathcal{F}} h^{2(a-1)+s} \int_F [D^a w][D^a v]\, dF \,. \tag{40}$$

ACCEPTED MANUSCRIPT

For a vector valued function $\mathbf{v}$ of dimension d, the ghost penalty term can be defined by using the straightforward component-wise extension of Eq. (40) as

$$\mathcal{G}_s(\mathbf{w},\mathbf{v}) := \sum_{i=1}^{d} \sum_{F\in\mathcal{F}} h^{2(a-1)+s} \int_F [D^a w_i][D^a v_i]\, dF \ . \tag{41}$$

Hence, using the operators defined in Eqs. (40) and (41), the ghost penalty terms for the incompressible Navier-Stokes equations with velocity $\mathbf{v}^f$ and pressure p as independent variables are given as

$$\mathcal{B}^f_{\mathrm{GP}}(\{\mathbf{w}^f, q)\}, \{(\mathbf{v}^f, p)\}) = \gamma^u_{\mathrm{GP}}\ \mu\, \mathcal{G}_1(\mathbf{w}^f, \mathbf{v}^f) + \gamma^p_{\mathrm{GP}}\ \frac{1}{\mu} g_3(q,p)\ , \tag{42}$$

where γ^u_{GP} and γ^p_{GP} are the dimensionless ghost penalty parameters for velocity and pressure, respectively. They can typically be chosen within a wide range of values spanning several orders of magnitude, such that they are small enough not to jeopardise accuracy but sufficiently large to effectively avoid excessive matrix condition numbers. A detailed investigation of the performance of ghost penalty terms on b-spline grids is presented in [34]. In the present work, the parameters have simply been set to $\gamma^u_{\mathrm{GP}} = \gamma^p_{\mathrm{GP}} = 0.01$ in all examples in Section 8.

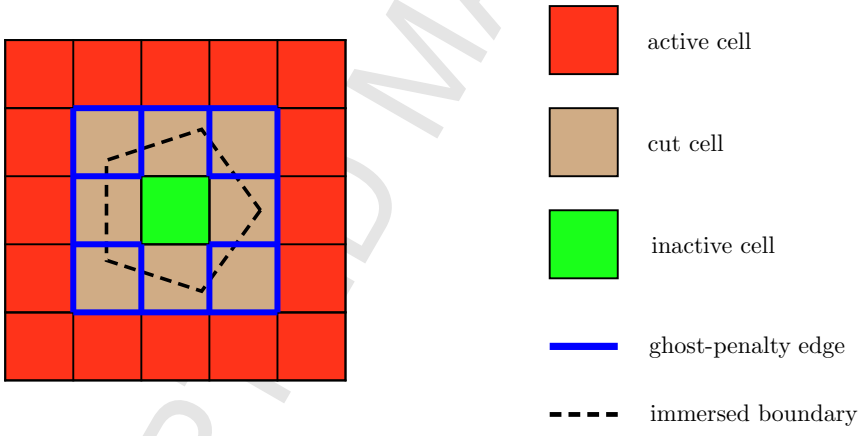


Figure 4: Active, inactive and cut cells for the scenario presented in Fig. 3. Ghost penalty operators are applied only on the edges of the background grid highlighted in blue color.

Remark: In [46, 47] ghost penalty terms are used effectively to stabilise the pressure field. For this purpose they are applied to all inter-element boundaries of the domain. A significant disadvantage of this type of pressure stabilisation is, however, that it substantially increases the fill-in of the system matrix by connecting degrees of freedom with neighbouring but non-overlapping support. In the context of higher order b-spline grids this additional fill-in is particularly undesirable. Therefore, the present work follows [34] and employs the PSPG/SUPG stabilisation technique.

ACCEPTED MANUSCRIPT

8. Numerical examples

The accuracy and robustness of the proposed scheme are assessed in this section by performing a number of numerical simulations. The following benchmark problems are studied: (1) flow around a fixed cylinder, (2) translational galloping of a square body, (3) rotational galloping of a rectangular body and (4) sedimentation of a circular cylinder. These are followed by the study of (5) the sedimentation of 25 particles, (6) the dynamic behaviour of a ball check valve with an elastic spring or (7) with a fixed retainer and (8) a model turbine.

All examples are based on the unsymmetric penalty-free Nitsche method and employ ghost penalty parameters $\gamma_{\mathrm{GP}_u} = \gamma_{\mathrm{GP}_p} = 0.01$.

8.1. Unsteady flow around a fixed circular cylinder for $Re = 100$ and $Re = 200$

The flow around a fixed cylinder is a classical benchmark example used to study the performance of a numerical scheme for incompressible fluids. The geometry and boundary conditions considered for this problem are shown in Fig. 5(a). The b-spline mesh with level-3 hierarchical refinement near the cylinder shown in Fig. 5(b) is used for the simulations. The cylinder surface is represented by 80 straight line segments. We consider the unsteady flow for $Re = 100$ and $Re = 200$. For the generalised-α scheme a spectral radius of $\rho_\infty = 0.2$ is used. Computed values of drag coefficient C_D, lift coefficient C_L and Strouhal number St are presented in Table 1. The use of Q_2 b-splines improves the accuracy of the results when compared with Q_1 b-splines and the results obtained with the Q_2 b-splines match well with the reference values. The evolution of C_D and C_L for $Re = 100$ and $Re = 200$ is shown in Fig. 6 and Fig. 7, respectively, with Q_2 b-splines and $\Delta t = 0.1$.

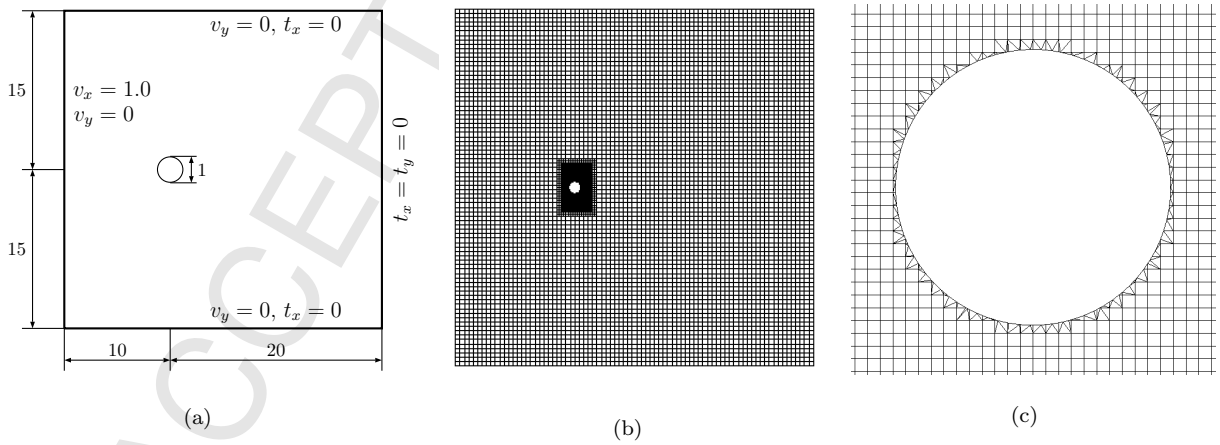


Figure 5: Flow around a fixed circular cylinder: (a) geometry and boundary conditions, (b) b-spline mesh and (c) subtriangulation of cut elements.

ACCEPTED MANUSCRIPT

	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
Braza et al. [68]	1.36	±0.250	-	1.40	±0.750	-
Liu et al. [69]	1.35	±0.339	0.165	1.31	±0.690	0.192
Calhoun [70]	1.33	±0.298	0.175	1.17	±0.668	0.202
Russell and Wang [71]	1.38	±0.300	0.169	1.29	±0.500	0.195
Le et al. [72]	1.37	±0.323	0.160	1.34	±0.430	0.187
Kadapa et al. [29]	1.39	±0.339	0.166	1.42	±0.711	0.194
Present - Level-2, $Q_1, \Delta t = 0.1$	1.27	±0.270	0.158	1.35	±0.662	0.183
Present - Level-2, $Q_1, \Delta t = 0.05$	1.27	±0.270	0.158	1.35	±0.662	0.186
Present - Level-3, $Q_1, \Delta t = 0.1$	1.28	±0.270	0.158	1.31	±0.704	0.192
Present - Level-3, $Q_1, \Delta t = 0.05$	1.28	±0.270	0.162	1.31	±0.701	0.188
Present - Level-2, $Q_2, \Delta t = 0.1$	1.34	±0.329	0.167	1.34	±0.714	0.195
Present - Level-2, $Q_2, \Delta t = 0.05$	1.35	±0.328	0.167	1.34	±0.717	0.195
Present - Level-3, $Q_2, \Delta t = 0.1$	1.34	±0.333	0.169	1.34	±0.716	0.196
Present - Level-3, $Q_2, \Delta t = 0.05$	1.34	±0.330	0.169	1.35	±0.717	0.196

Table 1: Flow around a fixed circular cylinder: C_D, C_L and St for $Re = 100$ and $Re = 200$.

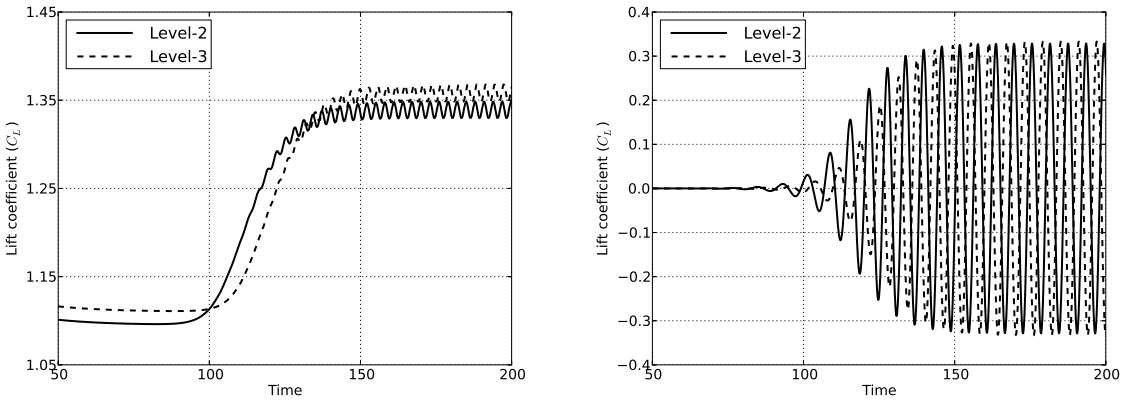


Figure 6: Flow around a fixed circular cylinder: evolution of C_D and C_L for $Re = 100$ with Q_2 b-splines and $\Delta t = 0.1$.

ACCEPTED MANUSCRIPT

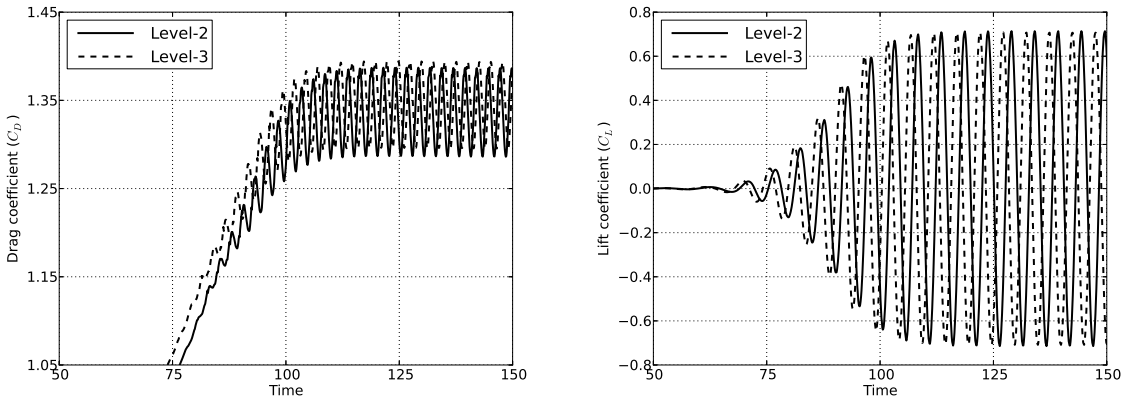
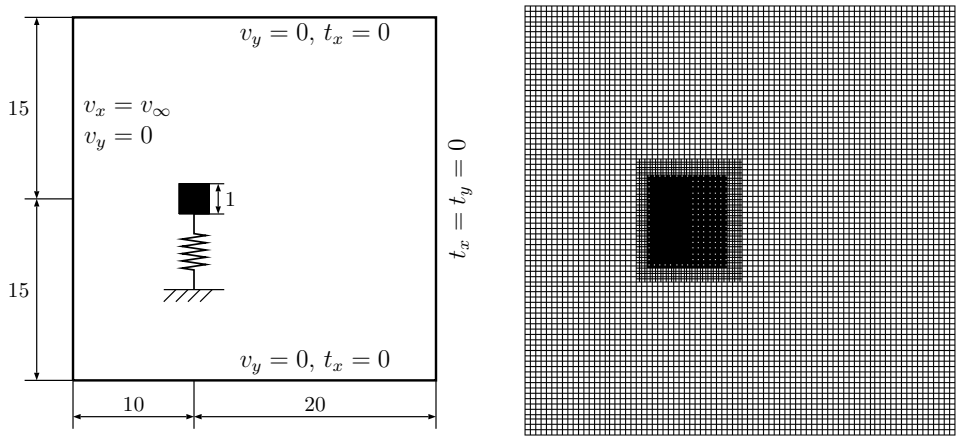


Figure 7: Flow around a fixed circular cylinder: evolution of C_D and C_L for $Re = 200$ with Q_2 b-splines and $\Delta t = 0.1$.

8.2. Translational galloping of a square body

Galloping is an aerodynamically unstable phenomenon involving high-amplitude oscillations of rigid-bodies exposed to fluid flow. For the theoretical background on galloping we refer to the text books by Blevins [73] and Den Hartog [74]. The geometry and boundary conditions of the problem considered are shown in Fig. 8(a). A rigid square body, of side length $D = 1$, supported on a spring-mass-damper system, starts oscillating transversely when exposed to fluid flow. The properties of the spring-mass-damper system, $K_{yy} = 3.08425, C_{yy} = 0.0581195$ and $M_{yy} = 20.0$, are the same as those used in [8, 75]. The natural frequency for this system is $f_n = 0.0625$. The density and viscosity of the fluid are $\rho^f = 1.0$ and $\mu^f = 0.01$, respectively. The inflow velocity is $v_\infty = 2.5$, so that the Reynolds number is $Re = \rho D v_\infty / \mu = 250$, which is one of the cases studied in [8, 75]. The hierarchical mesh shown in Fig. 8(b) with Q_1 and Q_2 b-splines is used for the analysis. The parameters for the generalised-α method and for the staggered scheme are set to $\rho^f_\infty = \rho^s_\infty = 0.2$ and $\beta = 0.5$, respectively. The simulations are carried out for two different time step sizes, $\Delta t = 0.1$ and $\Delta t = 0.05$. The displacement amplitude and frequency of the oscillation of the rigid body are given in Table 2. The results obtained with the present scheme agree well with the reference values reported in [8, 75]. Fig. 9 shows the evolution of the displacement for the level-2 mesh with $\Delta t = 0.1$ and Fig. 10 for the level-3 mesh with $\Delta t = 0.05$. Contour plots of vorticity and pressure at two different time instants are shown in Figs. 11 and 12.

ACCEPTED MANUSCRIPT



(a) (b)

Figure 8: Translational galloping: (a) geometry and boundary conditions and (b) hierarchical b-spline mesh.

	DOF	$\max(d_y^s)$	f_o
Robertson et al. [75]	-	1.150	0.938 f_n
Dettmer and Perić [8]	-	1.117	0.943 f_n
Present - Level-2, Q_1, $\Delta t = 0.1$	33195	1.080	$0.068 = 1.088\ f_n$
Present - Level-2, Q_2, $\Delta t = 0.1$	33417	1.050	$0.069 = 1.104\ f_n$
Present - Level-3, Q_1, $\Delta t = 0.1$	47460	1.081	$0.060 = 0.960\ f_n$
Present - Level-3, Q_2, $\Delta t = 0.1$	47475	1.046	$0.060 = 0.960\ f_n$
Present - Level-3, Q_1, $\Delta t = 0.05$	47460	1.197	$0.059 = 0.944\ f_n$
Present - Level-3, Q_2, $\Delta t = 0.05$	47475	1.202	$0.059 = 0.944\ f_n$

Table 2: Translational galloping: maximum value of displacement d_y^s and frequency of oscillations f_o.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

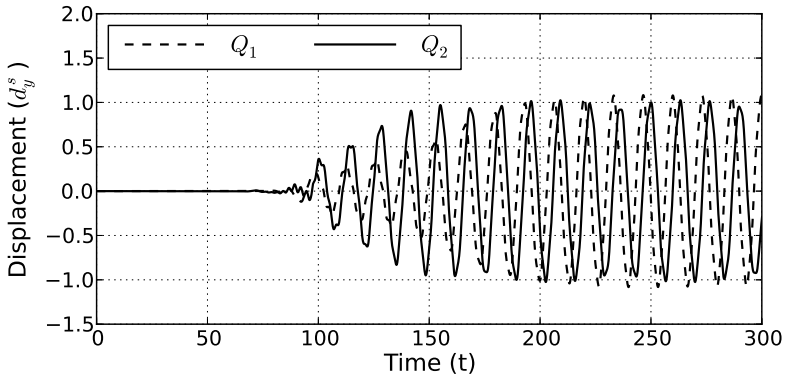


Figure 9: Translational galloping: evolution of displacement d_y for level-2 mesh with $\Delta t = 0.1$.

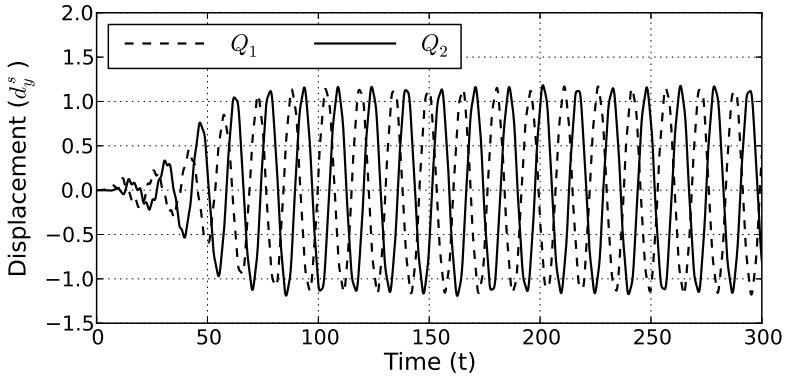


Figure 10: Translational galloping: evolution of displacement d_y for level-3 mesh with $\Delta t = 0.05$.

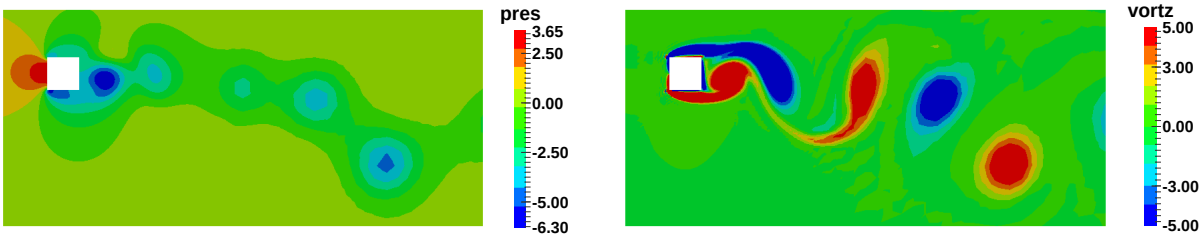


Figure 11: Translational galloping: contour plots of pressure and vorticity at $t = 201.2$ using level-3 Q_2 mesh.

ACCEPTED MANUSCRIPT

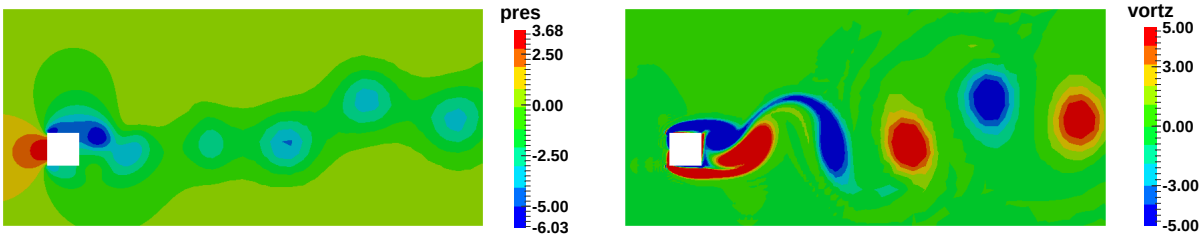


Figure 12: Translational galloping: contour plots of pressure and vorticity at $t = 209.4$ using level-3 Q_2 mesh.

8.3. Rotational galloping of a rectangular body

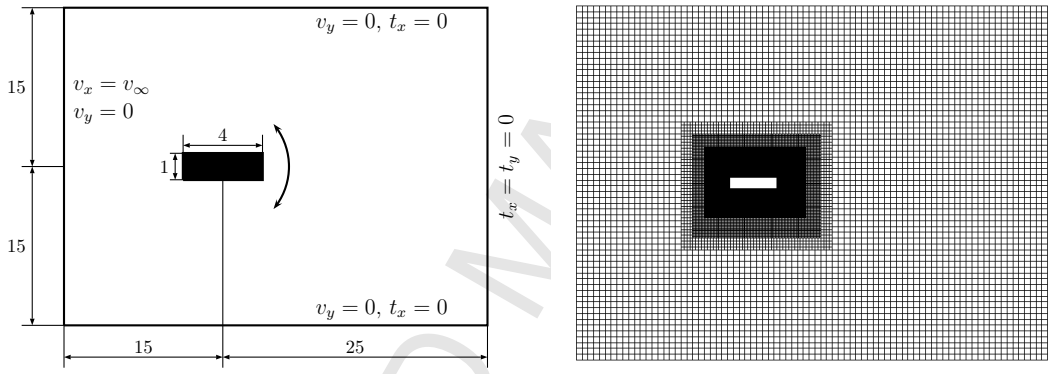
In this example we simulate the rotational galloping of a rectangular body studied in [8, 75, 76]. The geometry and boundary conditions of the problem are shown in Fig. 13(a). The thickness of the rectangular block is $D = 1$ and its length is $4\,D$. The body is constrained in x- and y-direction and its rotational degree of freedom is associated with a spring-mass-damper system. The properties of spring-mass-damper system are: moment of inertia $I_{\theta\theta} = 400$, damping $C_{\theta\theta} = 78.54$ and stiffness $K_{\theta\theta} = 61.685$. The density of the fluid is $\rho^f = 1.0$ and the viscosity is $\mu^f = 0.01$. The inlet velocity is set to $v_\infty = 2.5$ resulting in the Reynolds $Re = \rho D v_\infty/\mu = 250$. Simulations are carried out on the hierarchical b-spline mesh shown in Fig. 13(b) with level-3 and level-4 refinements using linear (Q_1) and quadratic (Q_2) b-splines. The spectral radii for the generalised-α schemes are set to $\rho^f_\infty = \rho^s_\infty = 0.8$. The relaxation parameter for the staggered scheme is chosen as $\beta = 0.9$. Simulations are performed using a constant time step size $\Delta t = 0.05$. The rectangular block is initially placed at an angle of 0.1 degrees in order to expedite the onset of oscillations.

The evolution of the angle of rotation of the rigid body, obtained with level-3 and level-4 meshes, is shown in Figs. 14 and 15, respectively. The amplitude, $\max(\theta)$, and frequency of the oscillations, f_o, obtained from the simulations, are tabulated in Table 3, along with the reference values. The results obtained with the present scheme are in excellent agreement with those published in Dettmer and Perić [8] and Robertson et al. [75]. Contour plots of pressure and vorticity at two different time instants are presented in Figs. 16 and 17.

ACCEPTED MANUSCRIPT

	DOF	max(θ)	f_o
Robertson et al. [75]	-	0.2620	0.762 f_n
Dettmer and Perić [8]	-	0.2670	0.780 f_n
Present - Level-3, Q_1	61560	0.2233	$0.052 = 0.832$ f_n
Present - Level-3, Q_2	109713	0.2262	$0.050 = 0.800$ f_n
Present - Level-4, Q_1	61491	0.2833	$0.050 = 0.800$ f_n
Present - Level-4, Q_2	109395	0.2688	$0.048 = 0.768$ f_n

Table 3: Rotational galloping: maximum value of angle of rotation (θ) and frequency of oscillations (f_o).



(a) (b)

Figure 13: Rotational galloping: (a) geometry and boundary conditions and (b) hierarchical b-spline mesh.

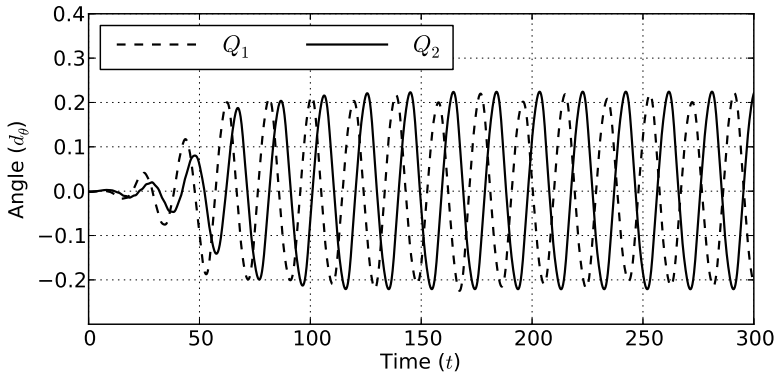


Figure 14: Rotational galloping: evolution of rotation angle θ with level-3 mesh.

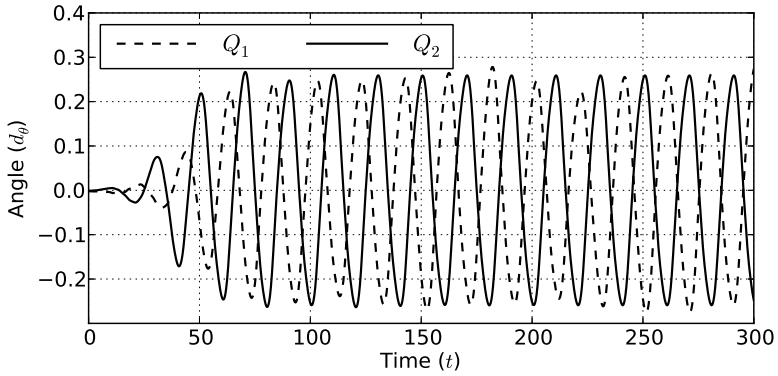


Figure 15: Rotational galloping: evolution of rotation angle θ with level-4 mesh.

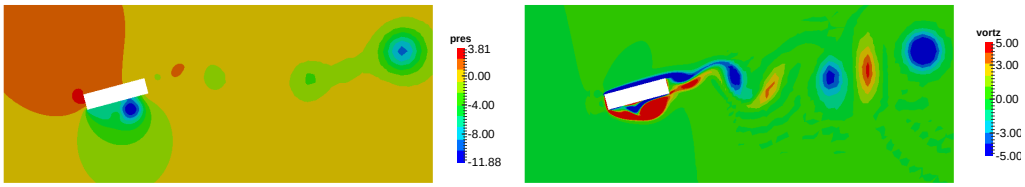


Figure 16: Rotational galloping: contour plots of pressure and vorticity at $t = 150.7$ using level-4 Q_2 mesh.

ACCEPTED MANUSCRIPT

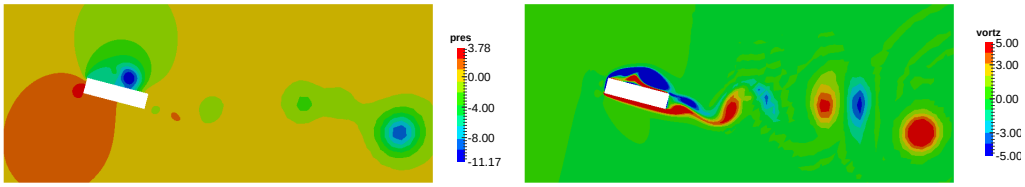


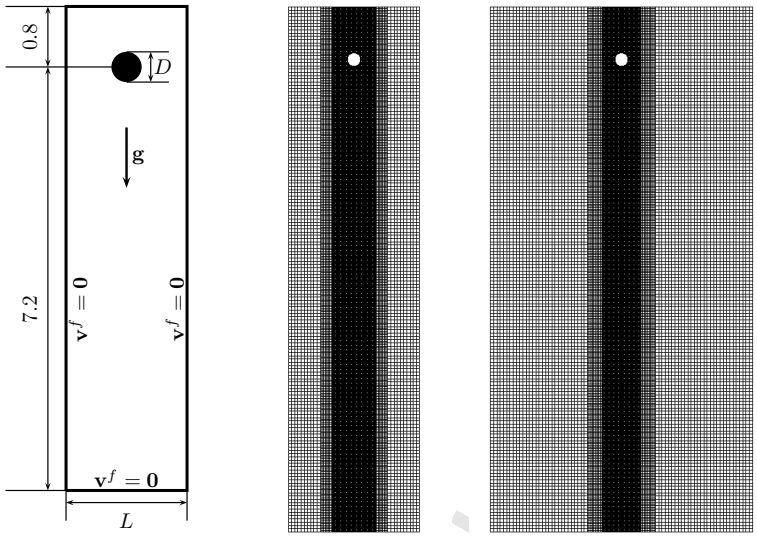
Figure 17: Rotational galloping: contour plots of pressure and vorticity at $t = 160.7$ using level-4 Q_2 mesh.

8.4. Particulate flow - single particle

Fictitious domain methods have been studied extensively for the modelling of particulate flows, see for instance Glowinski et al. [77–79]. Particulate flows have also been simulated in [7, 11] using the ALE framework with adaptive refinement. In this example we simulate the sedimentation of a single cylindrical particle which is falling freely under the action of gravity in an incompressible Newtonian fluid. The particle accelerates and eventually reaches a terminal velocity. The parameters are the same as those used in [11]. The geometry and boundary conditions of the problem are as shown in Fig. 18(a). The density and viscosity of the fluid are $\rho^f = 0.998$ g/cm^3 and $\mu = 0.0101$ dyn s/cm^{2}s, respectively. The diameter of the particle is $D = 0.2$ cm and its density is $\rho^s = 1.002\,\rho^f$. The value of the gravitational acceleration is g=981 cm/s^2. It is well known that the width of the channel influences the terminal velocity of the particle. Hence, simulations are carried out on two domains of different widths, $L = 10\,D$ and $L = 20\,D$, with the hierarchical meshes shown in Figs. 18(b) and 18(c). The parameters for the generalised-α method and the staggered scheme are $\rho^s_\infty = \rho^f_\infty = 0.0$ and $\beta = 0.1$, respectively. The time step size is set to $\Delta t = 0.02$.

The evolution of the vertical velocity of the particle is shown in Fig. 19. The terminal velocity attained by the particle in the narrow channel is obtained as 0.40 cm/s which agrees excellently with the result of 0.39 cm/s obtained in [11] for the same problem, but with a methodology based on an ALE formulation and remeshing. From two different semi-empirical formulae presented in [11] and references therein, the settling velocity of the considered particle in a wide channel is given as 0.42 cm/s or 0.48 cm/s. Our result of 0.44 cm/s obtained for the wide channel lies well within this range and is close to the settling velocity of 0.45 cm/s which is obtained in [11] for $L = 50\,D$. Contour plots of the velocity magnitude, pressure and vorticity at $t = 10$s are shown in Fig. 20 for $L = 20\,D$ using Q_2 b-splines.

ACCEPTED MANUSCRIPT



(a) (b) (c)

Figure 18: Sedimentation of single particle: (a) geometry and boundary conditions, (b) hierarchical b-spline mesh for $L = 10D$, and (c) hierarchical b-spline mesh for $L = 20D$.

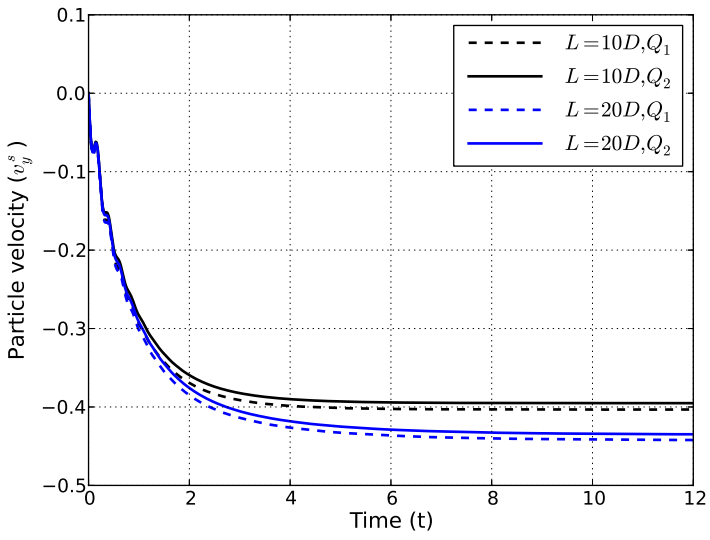


Figure 19: Sedimentation of single particle: time history of particle velocity.

ACCEPTED MANUSCRIPT

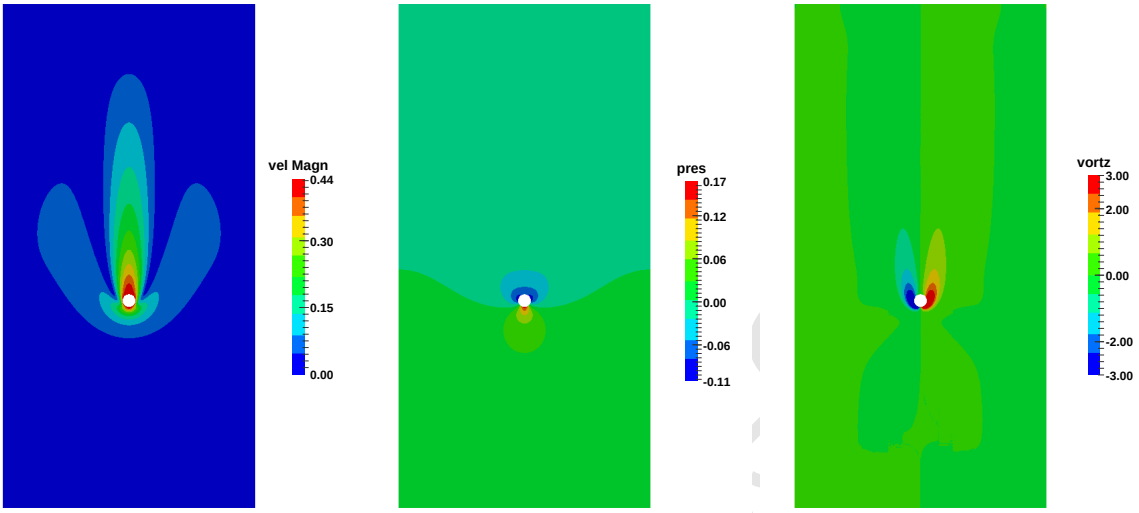


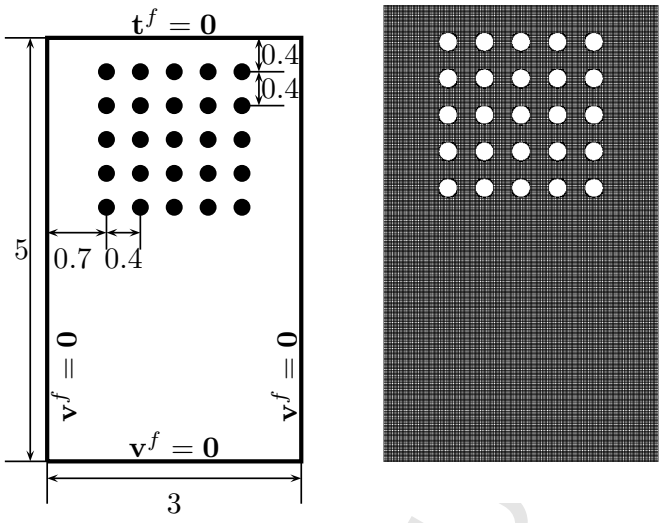
Figure 20: Sedimentation of single particle: contour plots of velocity magnitude, pressure and vorticity at $t = 10$s for $L = 20D$ with Q_2 b-splines.

8.5. Particulate flow - multiple particles

In this example we simulate the sedimentation of 25 cylindrical particles initially positioned as shown in Fig. 21(a). The uniform mesh of 180×300 elements displayed in Fig. 21(b) is used with Q_1 b-splines. The physical and simulation parameters are the same as those considered for the single particle in Section 8.4. A constant time step size $\Delta t = 0.005$ is used. Contact between particles and between particles and the grid boundary are modelled using Lagrange multipliers. The minimum gap width is set to two times the element size, which is equal to 1/6th of the particle diameter. Snap shots of displaced particles at several time instants during the sedimentation process are shown in Fig. 22.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

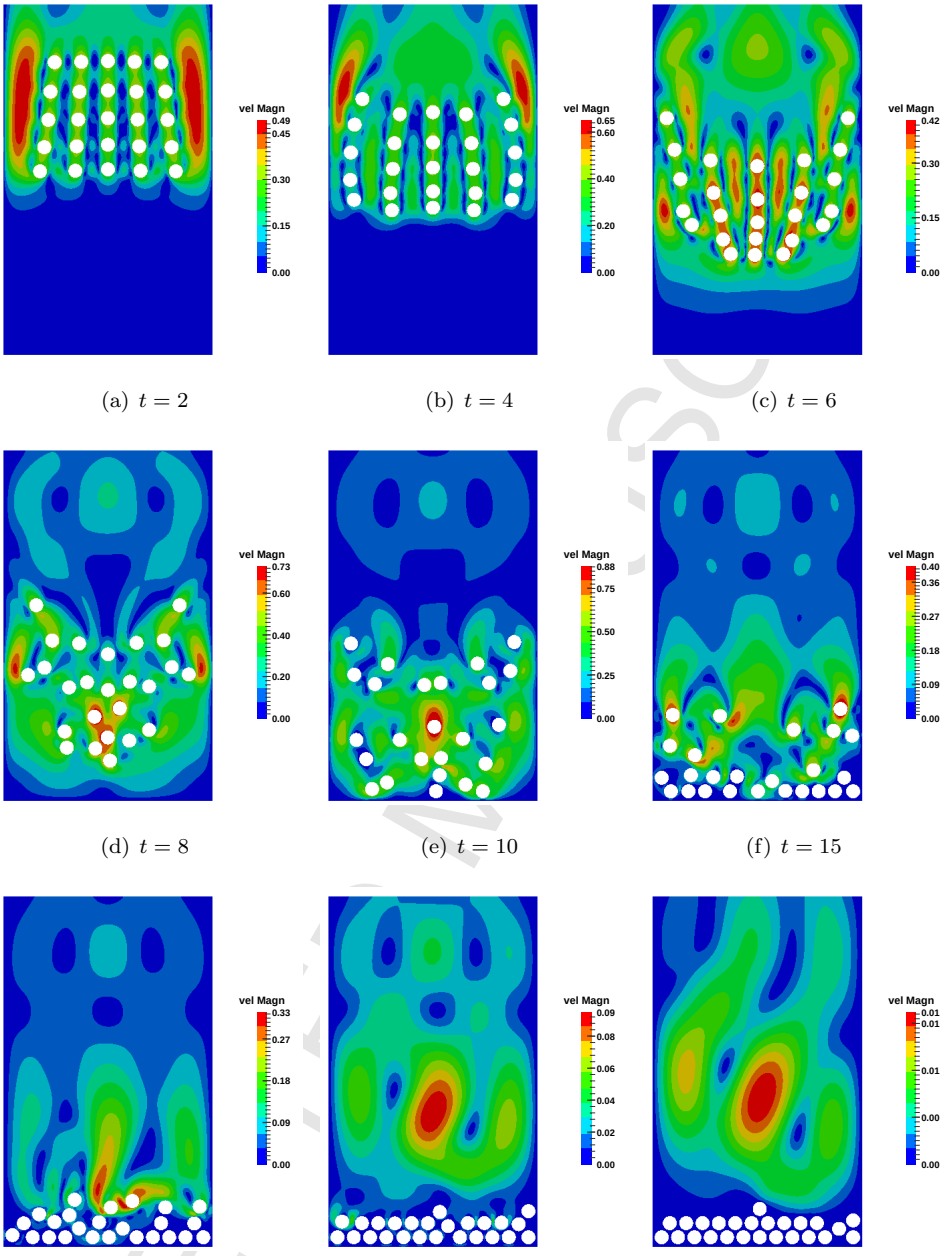


(a) (b)

Figure 21: Multiple particles: (a) geometry and boundary conditions and (b) mesh with particles at $t = 0$.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT



(a) $t = 2$ (b) $t = 4$ (c) $t = 6$

(d) $t = 8$ (e) $t = 10$ (f) $t = 15$

(g) $t = 18$ (h) $t = 25$ (i) $t = 50$

Figure 22: Multiple particles: contour plots of velocity magnitude at different time instants.

8.6. Ball check valve with an elastic spring

Various configurations of ball check valves are used in the mechanical engineering industry. They are, for instance, employed in hydraulic pumps or in automobile engines, where they control the oil flow. In this subsection we consider the axisymmetric model shown in Fig. 23(a). Here, the ball is connected to an elastic spring with linear stiffness, such that a positive pressure differential between inlet and outlet is required to

ACCEPTED MANUSCRIPT

move the ball along the central axis and to open a gap for the fluid flow.

The density and viscosity of the fluid are $\rho^f = 800$ kg/m^3 and $\mu^f = 50$ cP, respectively. The spherical ball is made of steel with density $\rho^s = 8000$ kg/m^3. The stiffness of the spring is $K_{yy} = 100$ N/m. The simulations are performed in an axisymmetric setting and are based on the hierarchical discretisation shown in Fig. 23(b). The fluid domain is refined locally in the zone where the ball comes into contact with the valve seat, thus allowing for a minimum contact gap of approximately 1% of the ball diameter. We point out that the minimum gap can be reduced further by adding more levels of localised refinement and doing so would not result in a significant increase of the computational cost. In fact, in certain applications the modelling of the seepage flow is important since it substantially influences the opening pressure of the valve.

Simulations are based on Q_1 b-splines. The parameters for the generalised-α schemes and for the staggered scheme are set to $\rho^s_\infty = \rho^f_\infty = 0.0$ and $\beta = 0.8$, respectively. The pressure at the outlet is zero and two different cases are considered at the inlet: (a) constant inlet pressure and (b) sinusoidally varying inlet pressure.

8.6.1. Constant inlet pressure

Simulations are performed for three different values of p_{in} with a constant time step $\Delta t = 0.2$. The evolution of the valve displacement and of the flow rates obtained from the simulations are displayed in Fig. 24. These graphs quantify the increase of the valve displacement and the flow rate due to larger inlet pressures. Contour plots of the velocity magnitude at the end of each simulation are shown in Fig. 25. Pressure contour plots are displayed in Fig. 26.

ACCEPTED MANUSCRIPT

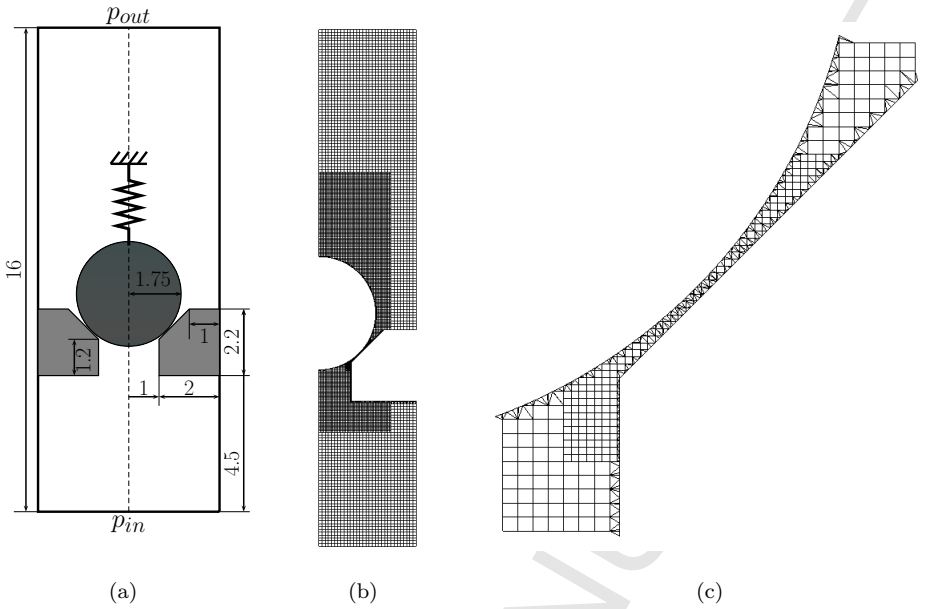


Figure 23: Ball check valve with spring: (a) geometry and boundary conditions (dimensions in [mm]), (b) hierarchical b-spline mesh and (c) discretisation around the opening.

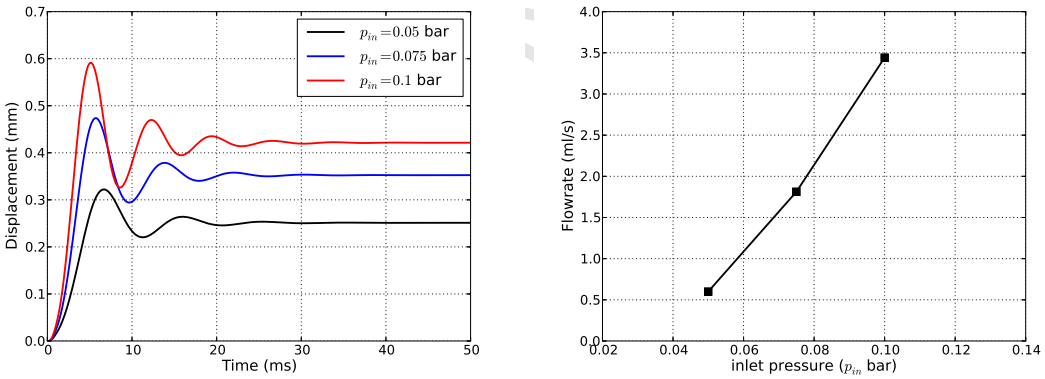
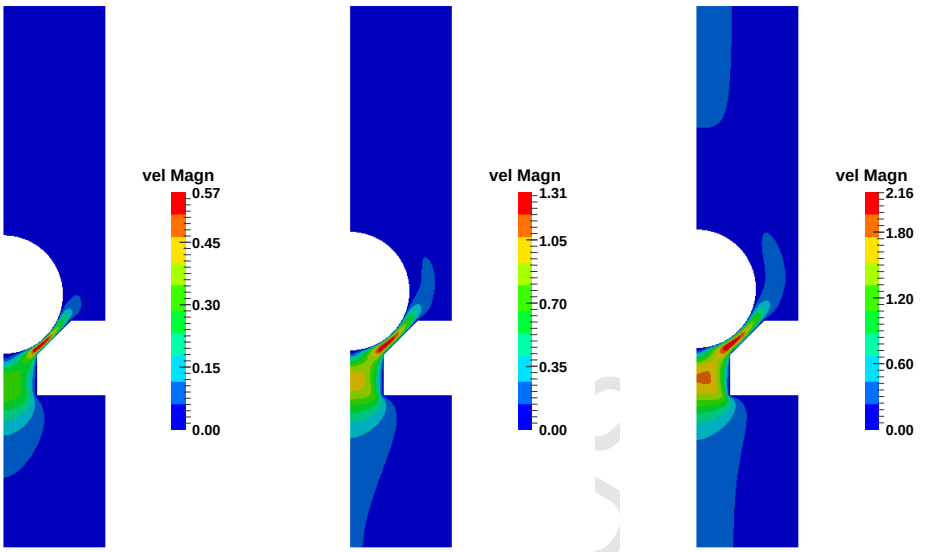


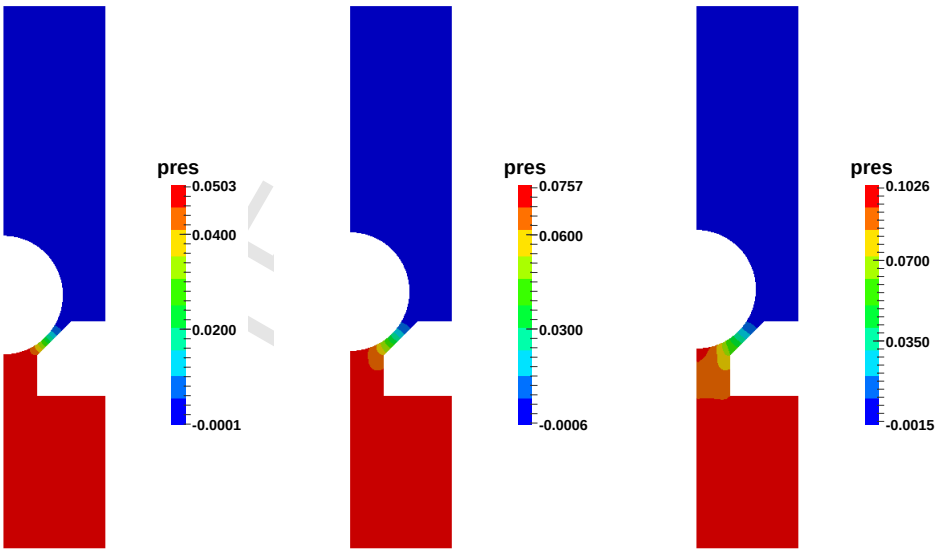
Figure 24: Ball check valve with spring: (a) evolution of ball displacement and (b) flow rates for different inlet pressure.

ACCEPTED MANUSCRIPT



(a) $p_{in} = 0.05$ bar (b) $p_{in} = 0.075$ bar (c) $p_{in} = 0.1$ bar

Figure 25: Ball check valve with spring: contour plots of velocity magnitude with constant inlet pressure.



(a) $p_{in} = 0.05$ bar (b) $p_{in} = 0.075$ bar (c) $p_{in} = 0.1$ bar

Figure 26: Ball check valve with spring: contour plots of pressure with constant inlet pressure.

8.6.2. Sinusoidal inlet pressure

We now assess the performance of the valve under sinusoidally varying inlet pressure, $p_{in} = p_A \sin(2\,\pi\,f t)$. For each simulation the time steps size is set to $\Delta t = 1/(100\,f)$. Fig. 27 shows graphs of the valve

ACCEPTED MANUSCRIPT

displacement for $p_A = 0.1$ bar and different values of f. From these graphs it can be observed that the maximum valve displacement decreases as the frequency of the inlet pressure is increased. The displacement of the valve is negligible for frequencies above $f = 500$ Hz. We observe that for frequencies of 100 Hz or more the backflow is of the same order of magnitude as the flow in the desired direction. This limits the application of this valve to the low frequency domain. Better performance at higher frequencies can be achieved by decreasing the ball inertia and increasing the valve stiffness or by applying a pre-load to the spring.

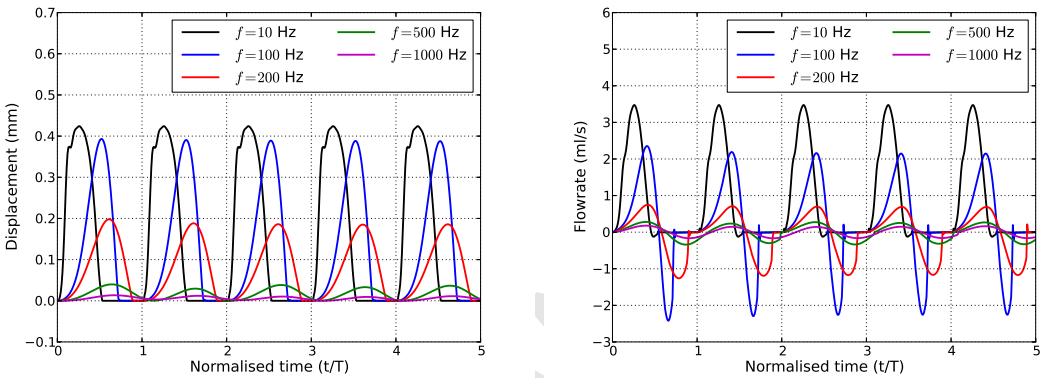


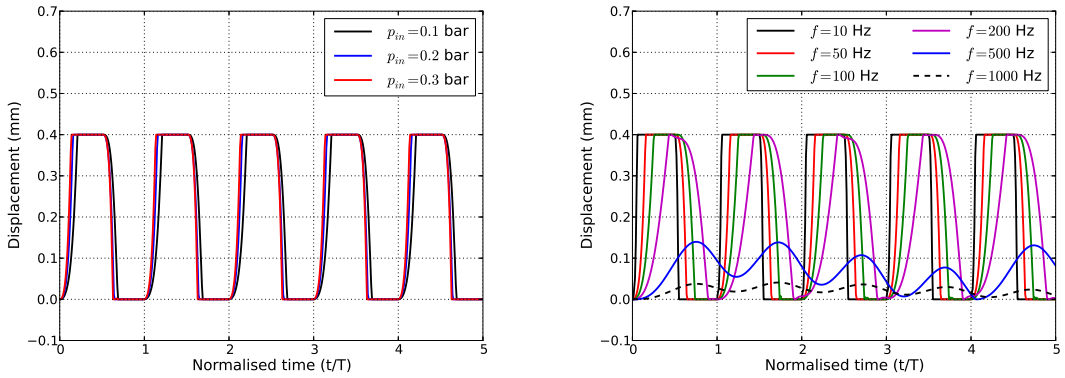
Figure 27: Ball check valve with spring: ball displacement and flow rates for $p_{in} = 0.1 \sin(2\,\pi\,f t)$ bar.

8.7. Ball check valve with a fixed retainer

In this example we consider the ball check valve described in Section 8.6 but without the elastic spring. Instead, the displacement of the ball is restricted to a maximum of 0.4 mm by mechanical contact with a fixed retainer. The presence of the retainer is accounted for by a Lagrange multiplier similarly to the contact between the ball and the valve seat. In the study performed here, the retainer is not 'seen' by the flow.

Fig. 28 shows the evolution of the displacement of the ball for different combinations of p_A and f. From these graphs we observe that for higher inlet pressures p_A and lower frequencies f, the valve switches more quickly between the closed and open positions. The impact velocities and contact forces at the valve seat and at the retainer are also computed by the adopted methodology. They are of great interest to the mechanical engineer as they serve as important design parameters in selecting suitable material for the valve components. As an example, input pressure, ball displacement, ball velocity, contact forces and flow rate for $p_A = 0.2$ bar and $f = 50$ Hz are displayed in Fig. 29.

ACCEPTED MANUSCRIPT



(a) (b)

Figure 28: Ball check valve without spring: ball displacement for (a) $f = 50$ Hz and different p_A and (b) $p_A = 0.2$ bar and different f.

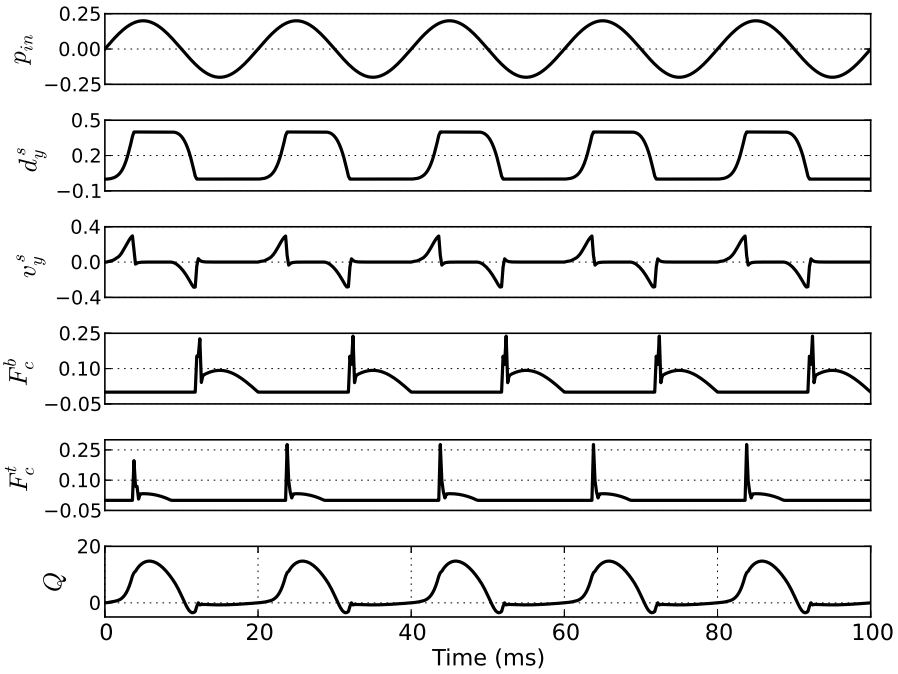


Figure 29: Ball check valve without spring: input pressure p_{in} [bar], ball displacement d_y^s [mm], ball velocity v_y^s [mm/ms], contact force for the bottom contact F_c^t [N], contact force for the top contact F_c^b [N] and flow rate Q [ml/s] for $p_A = 0.2$ bar and $f = 50$ Hz.

8.8. Model turbine

The model turbine studied in this subsection demonstrates the robustness of the scheme for problems involving full rotations of a component, which are a key feature of turbines, pumps and compressors. The

ACCEPTED MANUSCRIPT

geometry of the problem is shown in Fig. 30. The level-1 hierarchical mesh with Q_1 b-splines shown in Fig. 31 is used for the simulation. This discretisation results in approximately 90000 DOF. The density and viscosity of the fluid are $\rho^f = 1.0$ and $\mu^f = 0.02$, respectively. The rotary moment of inertia of the turbine wheel is $I_{\theta\theta} = 5.0$ and corresponds to a density of $\rho^s = 5.5$ for the wheel. A constant pressure value of 1500 is applied at the inlet and the outlet is assumed to be traction-free. The parameters for the generalised-α method and for the staggered scheme are set to $\rho^f_\infty = \rho^s_\infty = 0.0$ and $\beta = 0.2$, respectively. A constant time step of $\Delta t = 0.01$ is used. The evolution of the rotation angle and the angular velocity of the wheel is displayed in Fig. 32. Contour plots of velocity magnitude and pressure are shown in Figs. 33 and 34, respectively.

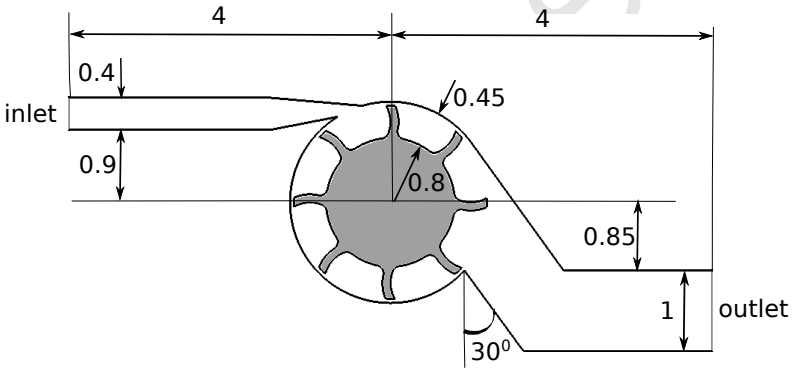


Figure 30: Model turbine: geometry.

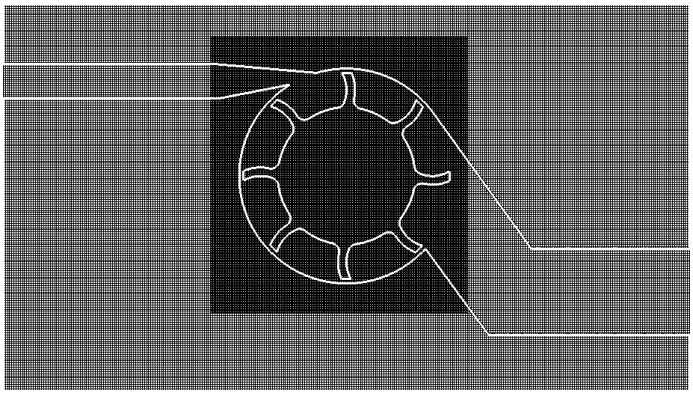


Figure 31: Model turbine: hierarchical b-spline mesh used for the simulation.

ACCEPTED MANUSCRIPT

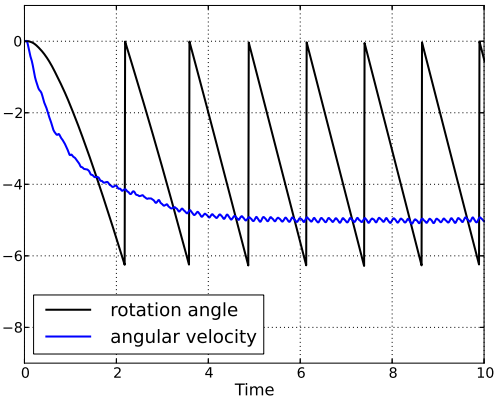


Figure 32: Model turbine: evolution of the rotation angle and the angular velocity. The rotation angle is reset to zero at the end of each full rotation for the purpose of clarity.

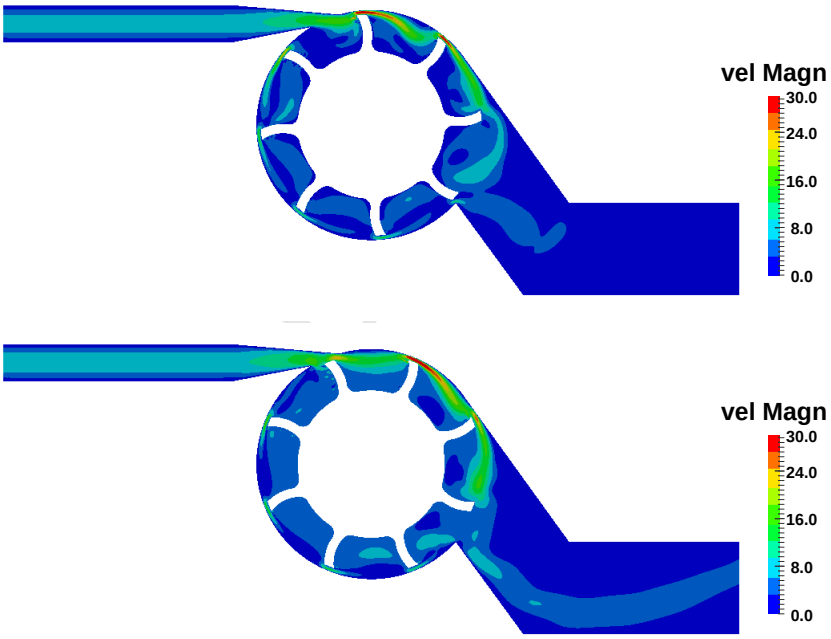


Figure 33: Model turbine: contour plots of velocity magnitude at $t = 1.0$ and $t = 5.02$.

ACCEPTED MANUSCRIPT

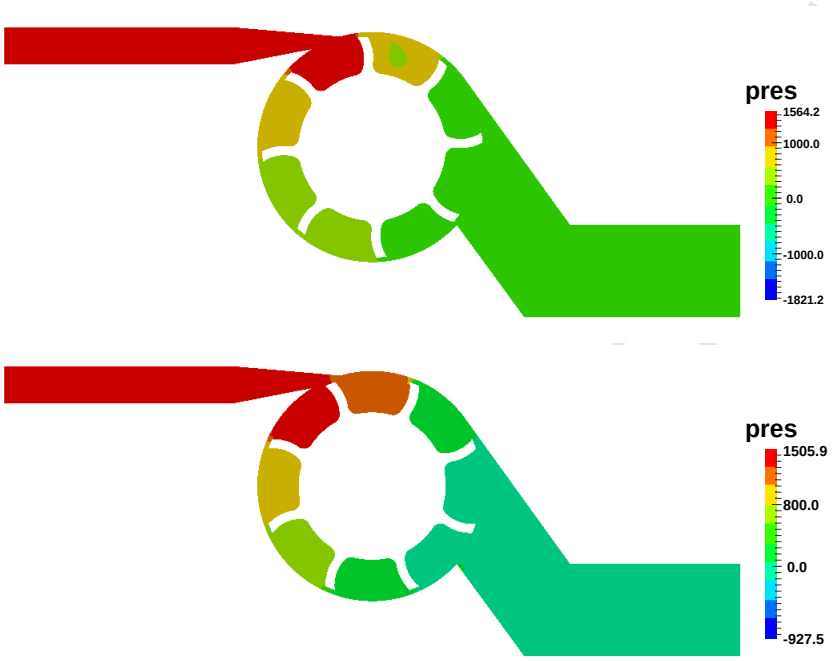


Figure 34: Model turbine: contour plots of pressure at $t = 1.0$ and $t = 5.02$.

9. Summary and conclusions

We have presented a numerical scheme for the simulation of fluid-structure interaction involving rigid solid bodies and laminar viscous incompressible fluid flow with topology changes and solid-solid contact.

Linear and quadratic b-spline grids with targeted hierarchical refinement have been used for the computationally efficient simulation of a number of numerical examples. The results obtained for the galloping and particle sedimentation benchmark problems are in excellent agreement with other published studies. The other numerical examples involve fluid-structure interaction and solid-solid contact. They include particulate flow, hydraulic check valves and a model turbine and thus clearly demonstrate the applicability of the proposed scheme to a wide range of industrially relevant problems. Key advantages of the present methodology lie in the robustness observed for large time steps and for the coupling of solid-solid contact with fluid-structure interaction.

Thus, the combination of the immersed finite element method proposed in [34] with the staggered scheme described in [9] and a standard Lagrange multiplier based solid-solid contact formulation has been proven to possess excellent potential for the modelling of coupled mechanical processes. Future work will be directed at its application to large-scale industrial simulations.

ACCEPTED MANUSCRIPT

Acknowledgements

This research project is funded by Schaeffler Technologies AG & Co. KG, Germany. This support is gratefully acknowledged.

References

[1] Y. Bazilevs, K. Takizawa, and Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. Wiley, 2013.

[2] H. Bungartz and M. Schäfer, editors. *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*. Lecture Notes in Computational Science and Engineering. Springer, 2006.

[3] H. Bungartz, M. Mehl, and M. Schäfer, editors. *Fluid Structure Interaction II: Modelling, Simulation, Optimization*. Lecture Notes in Computational Science and Engineering. Springer, 2010.

[4] *Encyclopedia of Computational Mechanics*, volume 1. Wiley.

[5] W. G. Dettmer and D. Perić. A computational framework for free surface fluid flows accounting for surface tension. *Computer Methods in Applied Mechanics and Engineering*, 195:3038–3071, 2006.

[6] W. G. Dettmer and D. Perić. A computational framework for fluid-structure interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195:5754–5779, 2006.

[7] W. G. Dettmer and D. Perić. A fully implicit computational strategy for strongly coupled fluid-solid interaction. *Archives of Computational Methods in Engineering*, 14:205–247, 2007.

[8] W. G. Dettmer and D. Perić. A computational framework for fluid-rigid body interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195:1633–1666, 2006.

[9] W. G. Dettmer and D. Perić. A new staggered scheme for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 93:1–22, 2013.

[10] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.

[11] P. H. Saksono, W. G. Dettmer, and D. Perić. An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 71:1009–1050, 2007.

[12] K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Spacetime interface-tracking with topology change (ST-TC). *Computational Mechanics*, 54:955–971, 2014.

[13] K. Takizawa, T. E. Tezduyar, S. Asada, and T. Kuraishi. Space-Time method for flow computations with slip interfaces and topology changes (ST-SI-TC). *Computers and Fluids*, 141:124–134, 2016.

[14] Y. Bazilevs, A. Korobenko, X. Deng, M. Kinzel, and J. O. Dabiri. FluidStructure Interaction Modeling of Vertical-Axis Wind Turbines. *Journal of Applied Mechanics*, 81, 2014.

[15] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[16] R. L. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1044, 1994.

[17] R. L. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18:709–735, 1997.

[18] A. J. Gil, A. A. Carreño, J. Bonet, and O. Hassan. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics*, 229:8613–8641, 2010.

[19] L. Zhang, A. Gerstenherger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193:2051–2067, 2004.

ACCEPTED MANUSCRIPT

[20] L. T. Zhang and M. Gay. Immersed finite element method for fluid-structure interactions. *Journal of Fluids and Structures*, 23:839–857, 2007.

[21] T. Rüberg and F. Cirak. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209-212:266–283, 2012.

[22] T. Rüberg and F. Cirak. A fixed-grid b-spline finite element technique for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 74:623–660, 2014.

[23] D. Kamensky, M. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid-structure interaction: application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[24] D. Kamensky, M. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluidstructure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 2016.

[25] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers and Fluids*, 141:135–154, 2016.

[26] M. Hsu, C. Wang, F. Xu, J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.

[27] V. Varduhn, M. Hsu, M. Reuss, and D. Schillinger. The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. *International Journal for Numerical Methods in Engineering*, 107:1054–1079, 2016.

[28] T. J. R. Hughes, Y. Bazilevs, and J. A. Cottrell. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[29] C. Kadapa, W. G. Dettmer, and D. Perić. A fictitious domain/distributed Lagrange multiplier based fluid-structure interaction scheme with hierarchical B-Spline grids. *Computer Methods in Applied Mechanics and Engineering*, 301:1–27, 2016.

[30] E. Burman and A. Ern. Continuous interior penalty hp-finite element methods for transport operators. *Mathematics of Computation*, 76:1119–1140, 2007.

[31] E. Burman. Ghost Penalty. *Comptes Rendus Mathematique*, 348:1217–1220, 2010.

[32] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 62:328–341, 2012.

[33] E. Burman and P. Hansbo. Fictitious domain methods using cut elements III. a stabilized Nitsche method for Stoke's problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:859–874, 2014.

[34] W. G. Dettmer, C. Kadapa, and D. Perić. A stabilised immersed boundary method on hierarchical b-spline grids. *Computer Methods in Applied Mechanics and Engineering*, 311:415–437, 2016.

[35] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[36] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60:353–371, 1993.

[37] P. Wriggers. *Computational Contact Mechanics*. Springer, Berlin Heidelberg, 2006.

[38] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*. Springer-Verlag, New York, 1997.

[39] D. F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.

[40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

[41] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers, San Francisco, 2006.

ACCEPTED MANUSCRIPT

[42] D. Vandevoorde and N. M. Josuttis. *C++ Templates: The Complete Guide*. Pearson Education, 1st edition, 2003.

[43] P. B. Bornemann and F. Cirak. A subdivision-based implementation of the hierarchical b-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013.

[44] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-252:116–150, 2012.

[45] F. de Prenter, C. V. Verhoosel, G. J. van Zwieten, and E. H. van Brummelen. Condition number analysis and preconditioning of the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 2016.

[46] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104:472–501, 2014.

[47] D. Schott and W. A. Wall. A new face-oriented stabilized XFEM approach for 2D and 3D incompressible NavierStokes equations. *Computer Methods in Applied Mechanics and Engineering*, 276:233–265, 2014.

[48] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227:7027–7051, 2008.

[49] J. W. Banks, W. D. Henshaw, and D. W. Schwendeman. An analysis of a new stable partitioned algorithm for FSI problems. Part I: Incompressible flow and elastic solids. *Journal of Computational Physics*, 269:108–137, 2014.

[50] J. W. Banks, W. D. Henshaw, and D. W. Schwendeman. An analysisof a new stable partitioned algorithm for FSI problems. Part II: Incompressible flow and structural shells. *Journal of Computational Physics*, 268:399–416, 2014.

[51] E. Burman and M. Fernández. Explicit strategies for incompressible fluid-structure interaction problems: Nitsche type mortaring versus Robin-Robin coupling. *International Journal for Numerical Methods in Engineering*, 97:739–758, 2014.

[52] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[53] R. K. Jaiman, N. R. Pillalamarri, and M. Z. Guan. A stable second-order partitioned iterative scheme for freely vibrating low-mass bluff bodies in a uniform flow. *Computer Methods in Applied Mechanics and Engineering*, 301:187–215, 2016.

[54] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.

[55] F. Brezzi and R. S. Falk. Stability of higher-order Hood-Taylor methods. *SIAM Journal on Numerical Analysis*, 28:581–590, 1991.

[56] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier Butterworth and Heinemann, Oxford, England, Sixth edition, 2005.

[57] A. Buffa, C. de Falco, and G. Sangalli. Isogeometric analysis: stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65:1407–1422, 2011.

[58] C. Kadapa, W. G. Dettmer, and D. Perić. Subdivision based mixed methods for isogeometric analysis of linear and nonlinear nearly incompressible materials. *Computer Methods in Applied Mechanics and Engineering*, 305:241–270, 2016.

[59] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, 1982.

[60] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[61] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.

ACCEPTED MANUSCRIPT

[62] I. Harari and T. J. R. Hughes. What are C and h?: Inequalities for the analysis and design of finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 97:157–192, 1992.

[63] S. Fernandes-Mendez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.

[64] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.

[65] A. Embar, J. Dolbow, and I. Harari. Imposing Dirichlet boundary conditions with Nitsches method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83:877–898, 2010.

[66] M. Reuss, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95:811–846, 2013.

[67] D. Schillinger, I. Harari, M. Hsu, D. Kamensky, S. K. F. Stoter, Y. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.

[68] M. Braza, P. Chassaing, and H. Ha Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165:79–130, 1986.

[69] C. Liu, X. Sheng, and C. H. Sung. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics*, 139:35–57, 1998.

[70] D. Calhoun. A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of Computational Physics*, 176:231–275, 2002.

[71] D. Russell and Z. J. Wang. A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics*, 191:177–205, 2003.

[72] D. V. Le, B. C. Khoo, and J. Péraire. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *Journal of Computational Physics*, 220:109–138, 2006.

[73] R. D. Blevins. *Flow Induced Vibration*. Van Nostrand Reinhold Company, New York, 1977.

[74] J. P. Den Hartog. *Mechanical Vibrations*. New York, Toronto, London. Mc Graw-Hill Book Company, 4th edition, 1956.

[75] I. Robertson, S. J. Sherwin, and P. W. Bearman. A numerical study of rotational and transverse galloping rectangular bodies. *Journal of Fluids and Structures*, 17:681–699, 2003.

[76] W. G. Dettmer. *Finite element modelling of fluid flow with moving free surfaces and interfaces including fluid-solid interaction*. PhD thesis, Swansea University, 2004.

[77] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux. A fictitious domain method with distributed Lagrange multipliers for the numerical simulation of particulate flow. *Contemporary Mathematics*, 218:121–137, 1998.

[78] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25:755–794, 1999.

[79] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. J. Périaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169:363–426, 2001.

ACCEPTED MANUSCRIPT

- Stabilised immersed framework for fluid-structure interaction.
- A second-order accurate staggered solution scheme.
- SUPG/PSPG stabilisation and ghost-penalty stabilisation.
- Penalty-free unsymmetric Nitsche method for imposing interface and boundary conditions.
- Applied to galloping, particulate flows, ball check-valve and a model turbine.

ACCEPTED MANUSCRIPT