



Subdivision-stabilised immersed b-spline finite elements for moving boundary flows

T. Rüberg, F. Cirak*

*Department of Engineering, University of Cambridge,
Trumpington Street, Cambridge CB2 1PZ, U.K.*

Abstract

An immersed finite element method is presented to compute flows with complex moving boundaries on a fixed Cartesian grid. The viscous, incompressible fluid flow equations are discretized with b-spline basis functions. The two-scale relation for b-splines is used to implement an elegant and efficient technique to satisfy the LBB condition. On non-grid-aligned fluid domains and at moving boundaries, the boundary conditions are enforced with a consistent penalty method as originally proposed by Nitsche. In addition, a special extrapolation technique is employed to prevent the loss of numerical stability in presence of arbitrarily small cut-cells. The versatility and accuracy of the proposed approach is demonstrated by means of convergence studies and comparison with previous experimental and computational investigations.

Keywords: Finite elements, Navier-Stokes, b-splines, subdivision refinement

*Corresponding author

Email address: f.cirak@eng.cam.ac.uk (F. Cirak)

Contents

1	Introduction	3
2	B-spline finite elements	5
2.1	One-dimensional b-spline basis functions and their two-scale relation	5
2.2	Tensor product b-spline basis functions	7
2.3	B-spline finite elements for axis-aligned cuboidal domains	8
3	Subdivision stabilised mixed formulation	12
3.1	Stokes system	12
3.2	Subdivision stabilisation	13
3.3	Numerical inf-sup test	15
3.4	Verification example: Stokes flow in a square	17
4	Geometrically complex domains	19
4.1	Weakly enforced Dirichlet boundary conditions	19
4.2	Immersed b-spline finite elements	21
4.2.1	Implicit boundary representation	21
4.2.2	Discrete system of equations and its solution	22
4.2.3	Cut-cell stabilisation	22
4.3	Verification examples	25
4.3.1	Study of penalty parameter and cut-cell sizes	25
4.3.2	Poisson equation in a sphere	27
4.3.3	Stationary Navier-Stokes equation in an irregular domain	28
5	Incompressible flow applications	30
5.1	Flow around a fixed cylinder	30
5.2	Moving cylinder	32
5.3	Flow around a sphere	33
5.4	Rotating ellipse	35
6	Conclusions	39

1. Introduction

Immersed, or embedded, boundary discretization methods in combination with fixed Cartesian grids are particularly appealing for fluid flow applications with moving boundaries. In such methods, the fluid grid is fixed throughout the problem domain and the boundary, mostly representing an immersed solid, is allowed to move freely through the fluid grid. The main difficulty in implementing immersed methods concerns the imposition of boundary conditions. Historically, immersed methods were first considered in the context of finite differences or finite volumes and were until recently only able to represent the boundary in a diffuse sense, see [1] for a review. To remedy the unwanted smearing of the boundary over few fluid cells, lately, a number of sharp interface techniques have been introduced, which combine ideas from original immersed methods with finite elements, see [2, 3, 4, 5]. In particular, the variational underpinnings of finite elements enable the formulation of consistent and stable immersed methods with an accuracy comparable to body-fitted unstructured meshes. At the same time, immersed methods practically eliminate the need to generate and maintain good quality meshes throughout a computation.

In immersed methods the physical domain boundary can intersect the Cartesian grid cells and the corresponding basis functions in an arbitrary way. This leads to cut-cells close to the physical domain boundaries which have to be suitably integrated. The consideration of Neumann boundary conditions is straightforward and concerns only the proper integration of non-homogeneous boundary terms. In contrast, it is considerably harder to impose Dirichlet boundary conditions given that the basis functions are non-interpolating at the physical domain boundaries. In order to enforce Dirichlet boundary conditions it is feasible to either modify the basis functions or to augment the governing equations with additional constraint equations. Representatives of the first group of methods include web-splines [6], i-splines [7] and max-ent interpolation [8]. In the second group, the constraint governing equations are solved with conventional techniques, such as the Lagrange multiplier method [9, 10], the penalty method or the Nitsche method [2, 11, 12]. Our method of choice for this work is the Nitsche method because it appears to provide the best trade-off between stability, accuracy and ease of implementation. It is known that the Nitsche method suffers from conditioning problems if small cut-cells are present, see, e.g., [13]. This can be remedied by eliminating the critical basis functions whose active support size inside the physical domain is too small. In the present work we follow the proposal of Höllig et al. [14] and link the critical basis functions to the ones inside the domain by means of Lagrange extrapolation.

B-spline basis functions can give more efficient and more accurate finite elements for a number of applications as demonstrated, for example, in [15, 16, 17, 18]. The better performance of b-splines as compared to conventional Lagrange basis functions is attributed to their higher-order continuity. Equally important is that the continuity of b-splines can be easily relaxed if necessary

by introducing repeated knots into their knot sequence. Furthermore, it is possible to increase the polynomial degree or to subdivide the knot sequence of a b-spline and at the same time to keep the represented geometry or solution field unmodified, see [19, 20, 21, 22]. In this work, we make use of the two-scale relation for b-splines which relates the b-spline basis functions on a coarse knot sequence with the ones of the same polynomial degree on a finer knot sequence. In algebraic terms, the coarse b-splines are obtained by multiplying the fine b-splines with a subdivision matrix. The two-scale relation can be taken as a convenient means for implementing incompressible fluid finite elements, which we refer to as *subdivision-stabilised* finite elements. Thereby, the pressure field is interpolated using coarse b-splines and the velocity field is interpolated using fine b-splines. The two-scale relation for b-splines allows the pressure and velocity interpolation to be performed using the fine b-splines and the resulting element matrices can subsequently be stabilised by multiplying with the subdivision matrix.

The outline of this paper is as follows. Section 2 begins with a review of b-splines and their refinability followed by the introduction of the b-spline finite elements in the case of axis-aligned cuboidal domains. In Section 3 the stable discretization of the Stokes equation using the two-scale relation for b-splines is introduced. The stability and convergence of the proposed approach is numerically confirmed with verification examples. This is followed in Section 4 by discussion of an immersed approach for discretization of geometrically complex domains. The essential components of this method are the weak enforcement of the Dirichlet boundary conditions with the Nitsche method and the Lagrange extrapolation of the solution field close to the boundaries to sidestep numerical instabilities due to cut-cells. Section 4 is complemented by several small verification examples which illustrate the methods relative insensitivity with respect to the size and number of cut-cells and the choice of the Nitsche parameter. Finally, in Section 5 the developed approach is applied to several stationary and non-stationary Navier-Stokes applications involving fixed and moving boundaries.

2. B-spline finite elements

In this Section, we first briefly review the b-spline basis functions and introduce their properties relevant for this paper. Subsequently, the set-up of the Cartesian grid used for finite element discretization is described. Further details on b-splines and their use as finite element basis functions can be found in standard literature [6, 15, 19, 20, 21].

2.1. One-dimensional b-spline basis functions and their two-scale relation

We consider the uniform b-spline basis functions over a parameter space with equally spaced knots $\xi_i = 0, 1, 2, 3, \dots$ defined by the recursive averaging formula

$$B_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$B_i^\mu(\xi) = \frac{\xi - \xi_i}{\xi_{i+\mu} - \xi_i} B_i^{\mu-1}(\xi) + \frac{\xi_{i+\mu+1} - \xi}{\xi_{i+\mu+1} - \xi_{i+1}} B_{i+1}^{\mu-1}(\xi),$$

where μ is the polynomial degree of the b-spline. Alternatively, b-spline basis functions can be defined using convolution relationships, which are better suited for inferring certain properties, such as smoothness. A b-spline basis function of degree μ is non-zero over an interval $\xi_i \leq \xi \leq \xi_{i+\mu+1}$. Moreover, the basis functions B_i^μ consist of piecewise polynomials of degree μ , which are joined $C^{\mu-1}$ -continuously over the knots. These smoothness properties do not hold in case of non-uniform splines with repeated knots, see, e.g., [19]. The uniform b-spline basis functions of degree μ computed with (1) have the same shape and differ only with respect to their position along the parameter space, this means that all basis functions can be presented as the translate of $B_0^\mu(\xi)$, i.e.

$$B_i^\mu(\xi) = B_0^\mu(\xi - i). \quad (2)$$

In Figure 1, the uniform b-splines with index zero, $i = 0$, and polynomial degrees from zero to three, $\mu = 0, 1, 2$ and 3 , are plotted. Clearly, the two lowest degree b-spline basis functions, B_0^0 and B_0^1 , are identical to the corresponding Lagrange basis functions. Furthermore, as can be inferred from Figure 1 according to the indexing scheme implied by (1) each b-spline is identified by the index of the knot at its left boundary.

A remarkable property of b-spline basis functions is their refinability, see [6, 22, 23], also known as the two-scale relation, which holds irrespective of the degree of the b-spline. In addition to the original coarse knot sequence, we consider a refined knot sequence $\tilde{\xi}_i = 0, 1/2, 1, 1 1/2, 2, \dots$, which is obtained by bisecting the original knot intervals. This implies the indexing scheme $\tilde{\xi}_{2i} = \xi_i$ so that the even numbered knots on the refined knot sequence are also present on the coarse knot sequence. The new b-splines \tilde{B}_i^μ over the fine knot sequence can be computed with the recursive averaging formula (1). It is straightforward to show that the following relation holds between the basis functions of the coarse and fine knot sequence

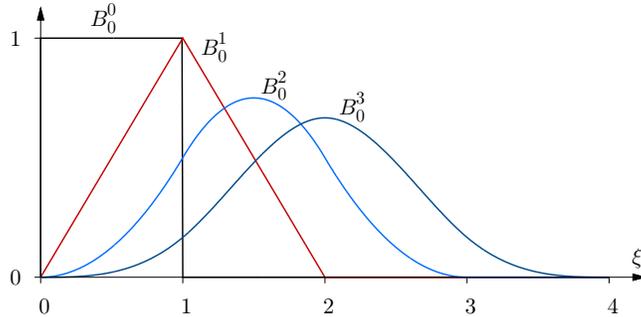


Figure 1: One-dimensional b-splines with index zero and polynomial degrees zero to three.

$$\tilde{B}_i^\mu(\xi) = B_0^\mu(2\xi - i), \quad (3)$$

where the factor 2 in the argument scales the support size of fine basis functions to half of the support size of coarse basis functions. It is also known that any b-spline $B_i^\mu(\xi)$ over the original coarse knot sequence can be represented as a linear combination of the b-splines $\tilde{B}_i^\mu(\xi)$ over the finer knot sequence using

$$B_i^\mu(\xi) = \frac{1}{2^\mu} \sum_{k=0}^{\mu+1} \binom{\mu+1}{k} \tilde{B}_{2i+k}^\mu(\xi) = \sum_k S_{ik}^\mu \tilde{B}_{2i+k}^\mu(\xi), \quad (4)$$

where the bracket expression in the first term represents the usual binomial coefficient and S_{ik}^μ is the subdivision matrix [22, 23], which has been defined for convenience. According to (4), each basis function $B_i^\mu(\xi)$ can be expressed as a linear combination of $\mu + 2$ basis functions $\tilde{B}_k^\mu(\xi)$. As illustrated in Figure 2a, the two-scale relation is, in particular, in case of linear b-splines apparent

$$B_i^1(\xi) = \frac{1}{2} \tilde{B}_{2i}^1(\xi) + \tilde{B}_{2i+1}^1(\xi) + \frac{1}{2} \tilde{B}_{2i+2}^1(\xi).$$

In addition, Figure 2b illustrates the two-scale relation in the case of quadratic b-splines, which reads

$$B_i^2(\xi) = \frac{1}{4} \tilde{B}_{2i}^2(\xi) + \frac{3}{4} \tilde{B}_{2i+1}^2(\xi) + \frac{3}{4} \tilde{B}_{2i+2}^2(\xi) + \frac{1}{4} \tilde{B}_{2i+3}^2(\xi).$$

The multiplication factors in these two examples are components of the subdivision matrix and do not depend on the knot index of the considered b-spline.

In passing, we note that the importance of refinability of b-spline basis functions cannot be overstated. Refinability is crucial to a number of b-spline techniques, such as subdivision curves and surfaces [16, 23, 24], hierarchical refinement [25, 26], t-splines [27] and wavelets [23].

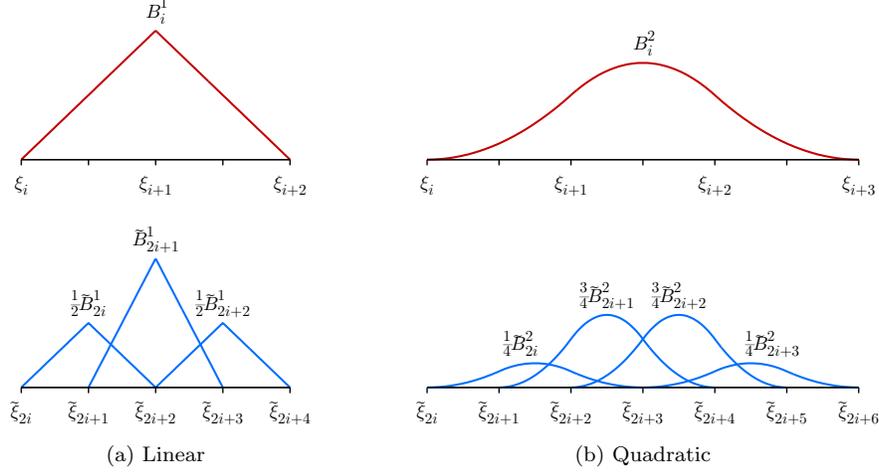


Figure 2: Two-scale relation for linear and quadratic b-splines.

2.2. Tensor product b-spline basis functions

We assume a computational domain Ω_{\square} which is a scaled d -dimensional axis-aligned hypercube with the lower left corner positioned at the coordinate origin, see Figure 3. Hence, the domain Ω_{\square} is either a rectangle or a rectangular cuboid in two or three space dimensions, respectively. The domain Ω_{\square} is partitioned with an orthogonal grid so that each grid point can be identified with a multi-index $\mathbf{i} = (i^1, \dots, i^d)$. The position vector of a grid point with the multi-index \mathbf{i} is denoted with

$$\mathbf{x}_{\mathbf{i}} = (x_{i^1}^1, \dots, x_{i^d}^d). \quad (5)$$

The chosen Cartesian grid implies a decomposition of the computational domain Ω_{\square} into $K = \max(i^1) \times \dots \times \max(i^d)$ identical cells $\omega_{\mathbf{k}}$

$$\Omega_{\square} = \bigcup_{\mathbf{k}=1}^K \omega_{\mathbf{k}}. \quad (6)$$

Each of the cells $\omega_{\mathbf{k}}$ is again labelled using a multi-index $\mathbf{k} = (k^1, \dots, k^d)$. The region of the cell $\omega_{\mathbf{k}}$ is the tensor-product of the associated grid point intervals

$$\omega_{\mathbf{k}} = [x_{k^1}^1, x_{k^1+1}^1] \times \dots \times [x_{k^d}^d, x_{k^d+1}^d]. \quad (7)$$

Hence, the length of the cell $\omega_{\mathbf{k}}$ in the coordinate direction x^{α} is given by

$$h_{\mathbf{k}}^{\alpha} = x_{k^{\alpha}+1}^{\alpha} - x_{k^{\alpha}}^{\alpha} \quad (8)$$

which enables the definition of a characteristic cell size

$$h_{\mathbf{k}} = (h_{k^1}^1 \times \dots \times h_{k^d}^d)^{1/d}. \quad (9)$$

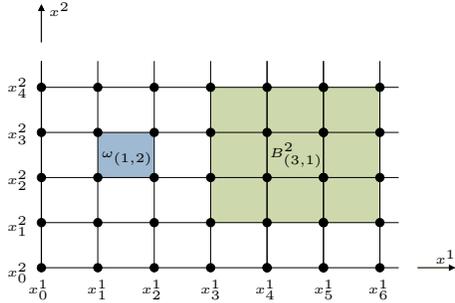


Figure 3: A cut-out from a two-dimensional computational domain Ω_{\square} with a specific cell $\omega_{(1,2)}$ and the support of a quadratic b-spline $B_{(3,1)}^2$. Each cell and b-spline is identified by the index of the grid point at its lower left corner.

It is assumed that the grids throughout this paper are *equidistant* so that there is the single grid parameter $h = h_{\mathbf{k}}$. This assumption has been adopted for computational efficiency and is not a principal restriction of the method.

The basis functions $B_{\mathbf{i}}^{\mu}$ on the Cartesian grid are defined as the d -dimensional tensor-products of the one-dimensional b-spline basis functions introduced in Section 2.1. Thus, the basis function of degree μ associated to the grid point with the multi-index \mathbf{i} is determined by

$$B_{\mathbf{i}}^{\mu}(\mathbf{x}) = B_{i^1}^{\mu}(x^1) \times \dots \times B_{i^d}^{\mu}(x^d). \quad (10)$$

Each of the one-dimensional uniform b-splines $B_{i^{\alpha}}^{\mu}$ can be expressed as the scaled translate of the one-dimensional basis function B_0^{μ} ,

$$B_{i^{\alpha}}^{\mu}(x^{\alpha}) = B_0^{\mu}((x^{\alpha}/h^{\alpha}) - i^{\alpha}), \quad (11)$$

assuming a linear mapping between the parameter space and the physical space of the form $\xi^{\alpha} = x^{\alpha}/h^{\alpha}$. This simple mapping is sufficient because of our restriction to orthogonal grids and uniform b-splines. In order to allow for non-orthogonal grids an isoparametric approach is feasible [15].

With regard to finite element interpolation, it is important to realise that the support of a basis function of degree μ covers $(\mu + 1)^d$ cells. Consequently, in the so-called ghost cells close to the lower left corner of the Cartesian grid, cf. Figure 3, not all the relevant basis functions forming a partition of unity are available. This means that only cells which are at least μ cells away from the lower left corner have the complete set of b-splines. In this context, the lower left corner of the grid is to be understood as the corner point where the indices i^1, \dots, i^d are all zero.

2.3. B-spline finite elements for axis-aligned cuboidal domains

As a representative boundary value problem, the Poisson equation on a rectangular axis-aligned domain is considered. We assume only Dirichlet boundaries

since their enforcement is a key aspect of any immersed finite element method. The Poisson problem is given by

$$\begin{aligned} -\nabla^2 u &= f && \text{in } \Omega \\ u &= \bar{u} && \text{on } \Gamma_D \equiv \Gamma, \end{aligned} \quad (12)$$

where \bar{u} is the prescribed solution over the entire boundary. The corresponding weak formulation can be stated as: Find $u \in \mathcal{V}_{\bar{u}}$ such that

$$a(u, v) = \ell(v) \quad \forall v \in \mathcal{V}_0 \quad (13)$$

with

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega, \quad \ell(v) = \int_{\Omega} f v \, d\Omega \quad (14)$$

and

$$\mathcal{V}_g = \{v \in \mathcal{H}^1(\Omega) | v = g \text{ on } \Gamma\}, \quad (15)$$

which is a subspace of the standard Sobolev space \mathcal{H}^1 restricted to functions which satisfy the boundary condition $v = g$.

In the present approach, the solution field is interpolated with tensor-product basis functions, i.e.

$$u^h(\mathbf{x}) = \sum_{\mathbf{i}} B_{\mathbf{i}}^{\mu}(\mathbf{x}) u_{\mathbf{i}}, \quad (16)$$

where the summation is over all grid nodes, i.e. $\sum_{\mathbf{i}}(\cdot) = \sum_{i_1} \cdots \sum_{i_d}(\cdot)$ and $u_{\mathbf{i}}$ is the scalar control coefficient of a b-spline basis function belonging to the grid point with the multi-index \mathbf{i} . The corresponding test functions are the same $B_{\mathbf{i}}^{\mu}$ as in any standard finite element method. Evidently, the computational domain Ω_{\square} , on which the tensor basis functions are defined, has to be larger than the physical domain Ω . In fact, Ω_{\square} has to be large enough so that in each cell, which is partly or fully contained in Ω , all the non-zero basis functions are present.

Special techniques are necessary for imposing Dirichlet boundary conditions because (i) the domain boundary Γ is not aligned with Cartesian grid lines, and (ii) the b-splines of degree $\mu \geq 2$ are non-interpolating, i.e. $u_{\mathbf{j}} \neq u^h(\mathbf{x}_{\mathbf{j}})$ for any $\mathbf{x}_{\mathbf{j}}$. Conceptually, the Dirichlet boundary conditions can be enforced by imposing in addition to the weak form (13) with $v \in \mathcal{H}^1$ a constraint equation of the form

$$\sum_{\mathbf{i}} B_{\mathbf{i}}^{\mu} u_{\mathbf{i}} - \bar{u} = 0 \quad \text{on } \Gamma_D. \quad (17)$$

Before addressing the general case in Section 4, to begin with, we consider axis-aligned cuboidal domains with boundaries conforming to the Cartesian grid lines. For such domains, it is feasible to develop modified b-splines close to the Dirichlet boundary and to determine the corresponding b-spline coefficients in a pre-processing step. Subsequently, the system of equations can be solved without having to explicitly enforce the constraint (17).

The modified tensor product b-splines use interpolating one-dimensional b-splines orthogonal to the boundary. There are a number of possibilities to

make a one-dimensional b-spline interpolating at specific locations, such as the begin and end of the curve [28]. For instance, in non-uniform b-splines, a basis function of degree μ is made interpolating at a particular knot by repeating that knot coordinate μ times in the knot sequence, see, e.g., [19]. To describe our approach, we consider the modification of the b-spline basis functions of degree μ in the x^1 coordinate direction so that they interpolate the left boundary at grid point $i^1 = \mu$. To simplify the notation, we write $x_{i^1=\mu}^1 = x_\Gamma^1$ and require $u^h(x_\Gamma^1) = \bar{u}(x_\Gamma^1)$. First, using the partition of unity property of the b-spline basis we write

$$\left(\frac{1 - B_1^\mu(x_\Gamma^1) - B_2^\mu(x_\Gamma^1) - \dots}{B_0^\mu(x_\Gamma^1)} \right) B_0^\mu(x^1) + B_1^\mu(x^1) + B_2^\mu(x^1) + \dots = 1. \quad (18)$$

From this, the following modified basis functions N_i^μ are defined

$$\begin{aligned} N_0^\mu(x^1) &= \frac{1}{B_0^\mu(x_\Gamma^1)} B_0^\mu(x^1) \\ N_i^\mu(x^1) &= \frac{-B_i^\mu(x_\Gamma^1)}{B_0^\mu(x_\Gamma^1)} B_0^\mu(x^1) + B_i^\mu(x^1) \quad i > 0. \end{aligned} \quad (19)$$

Obviously, $N_0^\mu(x_\Gamma^1) = 1$ and $N_{i>0}^\mu(x_\Gamma^1) = 0$, and due to the local support only the b-splines close to the boundary are affected. In Figure 4 the original and modified b-splines over one cell in case of a quadratic basis are plotted. The corresponding modified tensor-product b-splines are obtained by replacing the affected basis functions with the modified ones in (10). In two and three dimensions, the resulting b-spline basis is however not interpolating along the boundaries. In order to approximate Dirichlet boundary conditions, the coefficients of the modified basis functions can be determined with a least-squares fit

$$\min_{u_i^m} \int_{\Gamma_D} \left(\sum_i N_i^\mu u_i^m - \bar{u} \right)^2 d\Gamma. \quad (20)$$

Notice that the integral is over the boundary so that only the modified b-spline basis functions participate in the integration.

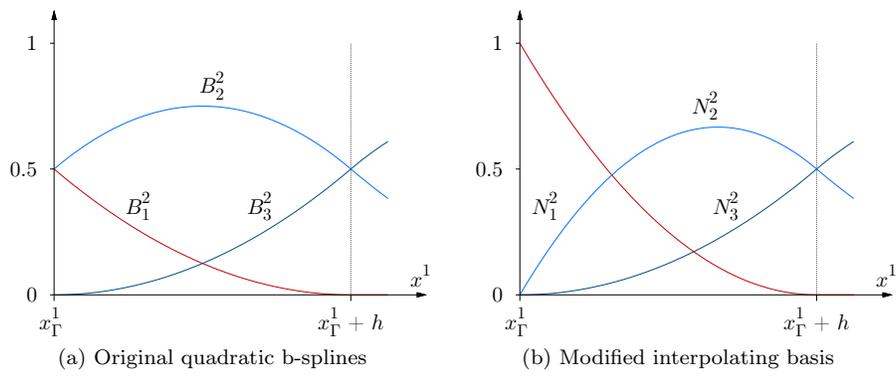


Figure 4: Basis functions over a grid cell next to the left boundary x_Γ^1 . Note in (b) the smooth transition to the right cell at $x_\Gamma^1 + h$.

3. Subdivision stabilised mixed formulation

Before moving on to the discretization of incompressible Navier-Stokes equations, we consider the stable finite element discretization of the Stokes equations. Here, we use the refinability or, in other terms, the two-scale relation for b-splines introduced in Section 2.1 for developing a family of stable mixed finite elements.

3.1. Stokes system

The Stokes system for modelling viscous, incompressible flow with the velocity and pressure fields (\mathbf{u}, p) reads

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) &= \mathbf{f}, & \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega \\ \mathbf{u} &= \bar{\mathbf{u}} & & & \text{on } \Gamma_D \\ \mathbf{t}(\mathbf{u}, p) &= \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n} = \bar{\mathbf{t}} & & & \text{on } \Gamma_N, \end{aligned} \quad (21)$$

where \mathbf{f} is the applied body force, $\bar{\mathbf{u}}$ is the prescribed velocity on the Dirichlet boundary Γ_D and $\bar{\mathbf{t}}$ is the prescribed traction on the Neumann boundary Γ_N , with the outward unit normal \mathbf{n} . Finally, $\boldsymbol{\sigma}(\mathbf{u}, p)$ represents the Newtonian fluid stress tensor

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) \quad \text{with} \quad \boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top) \quad (22)$$

with μ denoting the constant viscosity and $\boldsymbol{\varepsilon}$ denoting the strain rate tensor. The letter μ for the viscosity is not to be confused with the superscript μ indicating the degree of a b-spline.

The corresponding mixed variational problem reads: Find $(\mathbf{u}, p) \in \mathcal{V}_u \times \mathcal{Q}$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + b(\mathbf{u}, q) = \ell(\mathbf{v}) \quad \forall (\mathbf{v}, q) \in \mathcal{V}_0 \times \mathcal{Q} \quad (23)$$

with the bilinear and linear forms

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega, & b(\mathbf{v}, q) &= - \int_{\Omega} (\nabla \cdot \mathbf{v}) q \, d\Omega \\ \ell(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} \, d\Gamma. \end{aligned} \quad (24)$$

The definition of the relevant function spaces are

$$\begin{aligned} \mathcal{V}_g &= \{\mathbf{v} \in [\mathcal{H}^1(\Omega)]^d \mid \mathbf{v} = \mathbf{g} \text{ on } \Gamma_D\} \\ \mathcal{Q} &= \{q \in L_2(\Omega)\}. \end{aligned} \quad (25)$$

The space \mathcal{V}_g is a subspace of the standard Sobolev space \mathcal{H}^1 restricted to functions which satisfy the boundary condition $\mathbf{v} = \mathbf{g}$ on Γ_D . The space \mathcal{Q} is the standard space of square-integrable functions. Note that in case of a pure Dirichlet problem, i.e., $\Gamma \equiv \Gamma_D$, the pressure solution of (21) is only known up

to an arbitrary constant. In order to achieve solvability of the final system of equations, we simply fix a pressure degree of freedom.

For determining the finite element approximation, the pressure and velocity fields (\mathbf{u}, p) and the corresponding test fields in the weak form are discretized using the b-spline basis functions

$$\mathbf{u}^h(\mathbf{x}) = \sum_i B_i^{\mu_u}(\mathbf{x}) \mathbf{u}_i \quad \text{and} \quad p^h(\mathbf{x}) = \sum_i B_i^{\mu_p}(\mathbf{x}) p_i, \quad (26)$$

where μ_u and μ_p denote the polynomial degrees of the velocity and pressure b-spline basis functions, respectively. Inserting these approximations into the weak form (23) yields the discrete block system

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}. \quad (27)$$

As extensively studied, the approximation spaces for (\mathbf{u}^h, p^h) have to be chosen with great care to guarantee the inf-sup stability of the discrete system (27), see, e.g., [29] for details. Over the years, a great number of ingenious finite element techniques have been proposed to satisfy the inf-sup condition. Most of these methods are based on Lagrange basis functions, with the exception of few recent work on non-uniform b-splines [15, 30, 31].

Since the lowest degree b-splines with $\mu = 0$ and $\mu = 1$ are identical to Lagrange basis functions, their inf-sup stability (i.e. the lack thereof) can be inferred from corresponding studies. For example, the element types Q_1 - Q_0 and Q_1 - Q_1 are known to violate the inf-sup condition [29]. The corresponding b-spline combinations $(\mu_u = 1, \mu_p = 0)$ and $(\mu_u = 1, \mu_p = 1)$ will obviously fail as well. Furthermore, it is to be expected that equal-order b-spline approximations with $\mu_u = \mu_p = k$ for any $k \geq 1$ are not inf-sub stable similar to Q_k - Q_k approximations [29]. On the other hand, the Taylor-Hood element Q_2 - Q_1 and its generalisations Q_{k+1} - Q_k for $k \geq 1$ satisfy the inf-sup condition [32]. Unfortunately, we could not confirm this for the b-spline approximations with $\mu_u = k + 1$ and $\mu_p = k$. In our numerical studies the cases $k = 1$ and $k = 2$ exhibited numerical instabilities in the form typically reported for unstable combinations of Lagrange basis functions. For instance, for the case $\mu_u = 2$ and $\mu_p = 1$, which resembles the classic Taylor-Hood element, one can show that an assumed pressure distribution in form of a checkerboard pattern lies in the null space of \mathbf{B} .

3.2. Subdivision stabilisation

The two-scale relation for b-splines, introduced in Section 2.1, provides an alternative route for deriving and implementing efficient inf-sup stable elements. The two-scale relation facilitates the use of grids with two different resolutions for the velocity and pressure interpolations. Similar to the so-called Q_1 -iso- Q_2 element [33], for example, the pressure field can be interpolated on a coarse grid and the velocity field on a refined grid. As will be demonstrated below, a set of

new stable elements can be developed by using different combinations of b-spline degrees and grid resolutions by recourse to the two-scale relation. Importantly, the algorithmic complexity of having to operate with multiple grids is minimised by performing all the computations on the finest level and subsequently projecting them onto the coarse level during the assembly procedure.

Following the notation of Section 2.1, the basis functions on the fine grid are denoted with $\tilde{B}_{\mathbf{i}}^{\mu}$ and the basis functions on the coarse grid with $B_{\mathbf{i}}^{\mu}$. Thereby, \mathbf{i} is again a multi-index, $\mathbf{i} = (i^1, \dots, i^d)$, and the number of coarse grid b-splines is approximately $1/2^d$ times the number of fine grid b-splines. In the multi-dimensional case we write for the refinement relation

$$B_{\mathbf{i}}^{\mu}(\mathbf{x}) = \sum_{\mathbf{k}} S_{\mathbf{i}\mathbf{k}}^{\mu} \tilde{B}_{2\mathbf{i}+\mathbf{k}}^{\mu}(\mathbf{x}), \quad (28)$$

where the summation and the subdivision matrix $S_{\mathbf{i}\mathbf{k}}^{\mu}$ have to be understood in a tensor product sense, i.e.

$$B_{\mathbf{i}}^{\mu}(\mathbf{x}) = \left(\sum_{k^1} S_{i^1 k^1}^{\mu} \tilde{B}_{2i^1+k^1}^{\mu}(x^1) \right) \times \dots \times \left(\sum_{k^d} S_{i^d k^d}^{\mu} \tilde{B}_{2i^d+k^d}^{\mu}(x^d) \right). \quad (29)$$

With the preceding definitions, the velocity and pressure interpolations can be expressed as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{\mathbf{i}}^{N_f} \tilde{B}_{\mathbf{i}}^{\mu_u}(\mathbf{x}) \tilde{\mathbf{u}}_{\mathbf{i}} \quad \text{and} \quad p^h(\mathbf{x}) = \sum_{\mathbf{i}}^{N_c} B_{\mathbf{i}}^{\mu_p}(\mathbf{x}) p_{\mathbf{i}} \quad (30)$$

with N_f and N_c referring to the total number of fine grid and coarse grid degrees of freedom, respectively. For computational purposes it is appealing to express the pressure interpolation on the fine grid as well. To this end, we can make use of the mapping between the coarse and fine grid basis functions (28). In assembling the system matrix (27), computing the matrix block \mathbf{A} is straightforward since only the velocity grid needs to be taken into account. In contrast, for computing the matrix block \mathbf{B} , the velocity and pressure grids have to be simultaneously considered

$$\begin{aligned} b(\mathbf{u}^h, p^h) &= - \int_{\Omega} \sum_j^{N_c} \left(B_j^{\mu_p} p_j \right) \sum_{\mathbf{i}}^{N_f} \nabla \cdot \left(\tilde{B}_{\mathbf{i}}^{\mu_u} \tilde{\mathbf{u}}_{\mathbf{i}} \right) d\Omega \\ &= - \sum_{\mathbf{i}}^{N_f} \sum_j^{N_c} p_j \underbrace{\left[\int_{\Omega} B_j^{\mu_p} \left(\nabla \tilde{B}_{\mathbf{i}}^{\mu_u} \right) d\Omega \right]}_{\mathbf{B}_{\mathbf{i}j}} \cdot \tilde{\mathbf{u}}_{\mathbf{i}}. \end{aligned} \quad (31)$$

Using the refinement relation, the entries $\mathbf{B}_{\mathbf{i}j}$ of the matrix \mathbf{B} can also be expressed with

$$\mathbf{B}_{\mathbf{i}j} = \int_{\Omega} B_j^{\mu_p} \left(\nabla \tilde{B}_{\mathbf{i}}^{\mu_u} \right) d\Omega = \sum_{\mathbf{k}} S_{\mathbf{j}\mathbf{k}}^{\mu_p} \int_{\Omega} \tilde{B}_{2\mathbf{j}+\mathbf{k}}^{\mu_p} \left(\nabla \tilde{B}_{\mathbf{i}}^{\mu_u} \right) d\Omega. \quad (32)$$

Hence, the entries can first be computed using the fine grid basis functions without needing any information from the coarse grid. Multiplying the so obtained matrix during the assembly stage with the subdivision matrix yields the correct mixed-grid matrix. Therefore, in an existing b-spline finite element code, these proposed element types can be easily incorporated by simply adapting the assembly routine. Note that it is never necessary to establish the entire subdivision matrix S_{ik} because of its shift-invariant structure. As discussed in Section 2.1, each one-dimensional coarse grid b-spline function of degree μ can be represented by $\mu + 2$ fine-grid b-splines and, therefore, the subdivision matrix consists only out of $(\mu + 2)^d$ unique rows (again, d being the space dimension). Moreover, the shift-invariant structure of the Cartesian grid and the uniformity of the b-spline basis functions can be exploited to greatly streamline the computer implementation and computation of the system matrix.

3.3. Numerical inf-sup test

We establish the stability of the proposed two-grid b-spline finite elements using the numerical inf-sup test as proposed in [29]. Whereas the linear b-spline element with $\mu_u = \mu_p = 1$ is identical to the provably stable Q_1 -iso- Q_2 element [33], there are no analytic proofs available for other considered pairs of b-splines.

The discrete stability condition for mixed finite elements requires that the discrete spaces for velocity and pressure approximation fulfil the inequality

$$\sup_{\mathbf{v}_h \in \mathcal{V}_h} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{\mathcal{V}}} \geq \beta_h \|q_h\|_{\mathcal{Q}} \quad \forall q_h \in \mathcal{Q}_h, \quad (33)$$

with a positive constant β_h independent of the mesh size. This critical constant β_h can be identified as the smallest non-zero eigenvalue λ_{\min} of the following eigenvalue system

$$\mathbf{B}\mathbf{v}_i = \lambda_i \mathbf{Q}\mathbf{q}_i \quad \text{and} \quad \mathbf{B}^\top \mathbf{q}_i = \lambda_i \mathbf{V}^\eta \mathbf{v}_i. \quad (34)$$

It has to be ensured that the eigenvectors \mathbf{v}_i and \mathbf{q}_i are not in the null spaces of \mathbf{B} and \mathbf{B}^\top , respectively [29]. The norm matrices \mathbf{Q} and \mathbf{V}^η used in these equations have been defined as

$$\begin{aligned} \mathbf{Q}_{ij} &= \int_{\Omega} B_i B_j \, d\Omega \\ \mathbf{V}_{ij}^\eta &= \eta \int_{\Omega} \tilde{B}_i \tilde{B}_j \, d\Omega + \int_{\Omega} \nabla \tilde{B}_i \cdot \nabla \tilde{B}_j \, d\Omega. \end{aligned} \quad (35)$$

Here, the parameter η enables to switch between the H^1 -norm ($\eta = 1$) and the H^1 -seminorm ($\eta = 0$). Therefore, the discrete norms can be computed with

$$\|\mathbf{q}\|_0^2 = \mathbf{q}^\top \mathbf{Q}\mathbf{q}, \quad \|\mathbf{v}\|_1^2 = \mathbf{v}^\top \mathbf{V}^1 \mathbf{v}, \quad \text{and} \quad |\mathbf{v}|_1^2 = \mathbf{v}^\top \mathbf{V}^0 \mathbf{v}. \quad (36)$$

Elimination of the eigenvector \mathbf{v}_i from system (34) yields the equivalent eigenvalue problem

$$\mathbf{B}(\mathbf{V}^\eta)^{-1} \mathbf{B}^\top \mathbf{q}_i = \lambda_i^2 \mathbf{Q}\mathbf{q}_i. \quad (37)$$

Note that $\mu \mathbf{V}^0 = \mathbf{A}$, with μ being the fluid viscosity, so that for the seminorm case $\eta = 0$, the left hand side of (37) is the Schur complement of the system (27).

In the numerical tests reported below, we computed the eigenvalues of expression (37) for Dirichlet and Neumann problems on a unit square with varying grid sizes. In case of the Dirichlet problem, the pressure can only be determined up to a constant and, for this reason, there is exactly one zero eigenvalue. On the other hand, the matrix $\mathbf{V}^0 = \mu^{-1} \mathbf{A}$ is singular for the Neumann problem since the velocity in each coordinate direction can only be determined up to a constant. Therefore, the full H^1 -norm is used by employing the matrix \mathbf{V}^1 . In this case, the boundary conditions indirectly determine the pressure (cf. the Neumann boundary condition in (21)) and, hence, no zero eigenvalue occurs in (37).

Figure 5 shows the computed inf-sup constants as the smallest non-zero eigenvalue of (37). The considered polynomial degrees of the splines are $\mu_u = \mu_p = k$ and $\mu_u = \mu_p + 1 = k$ with $k = 1$ and $k = 2$. In all cases the velocity field is interpolated on the fine grid and the pressure field is interpolated on the coarse grid. In general, the inf-sup constants for the Neumann problem are larger than for the Dirichlet problem. Furthermore, all curves tend towards a positive constant value with increasing number of grid points N per direction. Hence, as stipulated by the numerical inf-sup test, there is a mesh-size independent constant β_h and the method can be considered stable.

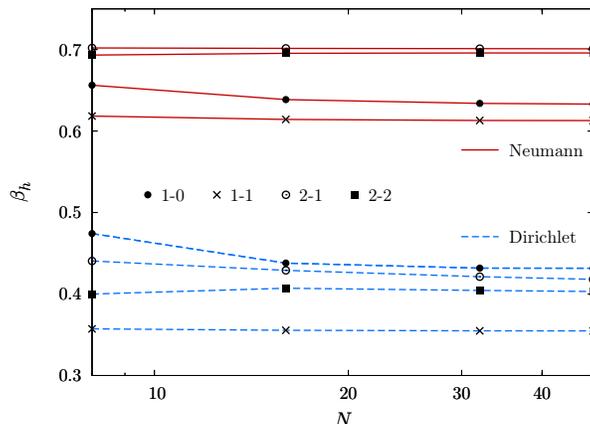


Figure 5: Computed inf-sup constant vs. number of grid points for Dirichlet and Neumann problems and different combinations of polynomial degrees $\mu_u - \mu_p$.

Even though an analytic proof of stability for the presented family of b-spline finite elements is presently not available our numerical tests provide strong evidence for their soundness. In this context it appears also to be relevant that we only work with Cartesian grids so that all our grids are, allowing for boundary conditions, identical to the ones used in the foregoing stability analysis. Moreover, no stability issues have been observed in any of the presented numerical examples.

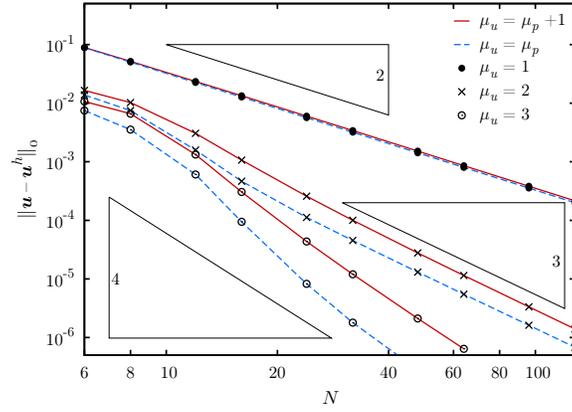
3.4. *Verification example: Stokes flow in a square*

As a small illustrative example, a Dirichlet problem on the unit square $\Omega = (0, 1) \times (0, 1)$ with the analytic solution

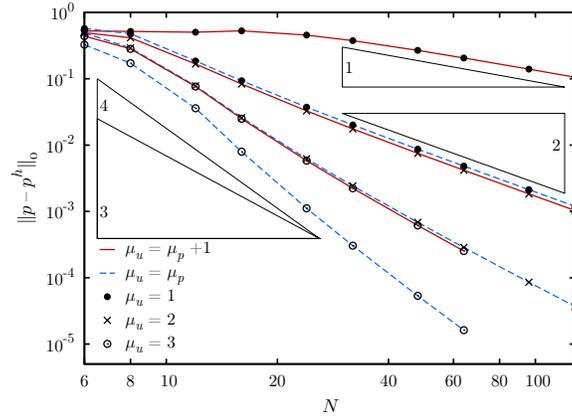
$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} \sin(2\pi x^1) \cos(2\pi x^2) \\ -\cos(2\pi x^1) \sin(2\pi x^2) \end{pmatrix}, \quad p(\mathbf{x}) = \sin(3\pi x^1) \cos(3\pi x^2) \quad (38)$$

and a fluid viscosity $\mu = 1$ is computed. The boundary conditions are applied using a least-squares fit as described in Section 2.3. The convergence behaviour is plotted in Figure 6. B-spline degrees of $\mu_u = 1$ to $\mu_u = 3$ are used together with pressure interpolations of equal order ($\mu_p = \mu_u$) or one order lower ($\mu_p = \mu_u - 1$). In all computations the pressure field is interpolated on the coarse grid and the velocity field on the fine grid. It is worthwhile emphasising that using the same grid for pressure and velocity interpolations leads in general to unstable elements. In Figure 6, one can observe the following optimal convergence orders

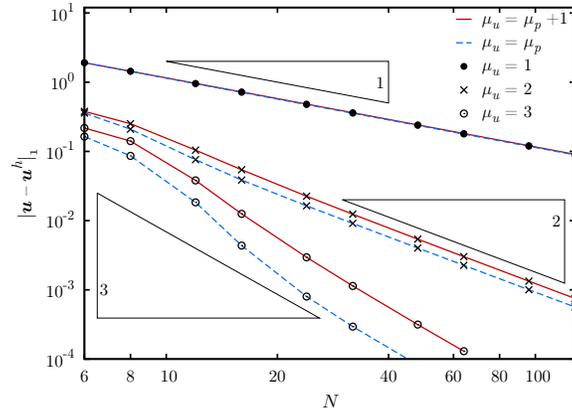
$$\|\mathbf{u} - \mathbf{u}^h\|_0 = Ch^{\mu_u+1}, \quad |\mathbf{u} - \mathbf{u}^h|_1 = Ch^{\mu_u}, \quad \text{and} \quad \|p - p^h\|_0 = Ch^{\mu_p+1}. \quad (39)$$



(a) L_2 -norm error of velocities



(b) L_2 -norm error of pressure



(c) H_1 -seminorm error of velocities

Figure 6: Convergence of the errors in velocity and pressure fields for different combinations of polynomial degrees $\mu_u - \mu_p$.

4. Geometrically complex domains

The b-spline finite element method introduced so far is only suitable for analysing rectangular axis-aligned domains. In this section, we introduce its extension towards the analysis of domains with arbitrary geometry and topology.

4.1. Weakly enforced Dirichlet boundary conditions

We start off by considering the problem of enforcing Dirichlet boundary conditions in the case where they are not satisfied by the chosen finite element basis. For this purpose, we decided to use a variationally consistent penalty method, which is also known as the *Nitsche method* [11]. In this context, there is a fine line between the various constraint enforcement techniques so that the Nitsche method can also be interpreted, for example, as a stabilised Lagrange multiplier method [34]. In recent years, the Nitsche method has been introduced to an increasing number of application areas, including interface problems [2], boundary enforcement in immersed and element-free methods [35, 36, 37], and fluid-structure interaction [12].

In the following we introduce the derivation of the Nitsche method for the Navier-Stokes system. This derivation can easily be specialised to other problems discussed in this paper, since the Navier-Stokes system represents the most generic considered problem. In its stress-divergence form, the Navier-Stokes equation can be expressed as

$$\begin{aligned} \rho \frac{D\mathbf{u}}{Dt} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) &= \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0 && \text{in } \Omega \\ \mathbf{u} &= \bar{\mathbf{u}} && \text{on } \Gamma_D \\ \mathbf{t}(\mathbf{u}, p) &= \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n} = \bar{\mathbf{t}} && \text{on } \Gamma_N \\ \mathbf{u}(\cdot, 0) &= \mathbf{u}_0 && \text{in } \Omega \end{aligned} \quad (40)$$

with the constant mass density ρ , the total time derivative

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \quad (41)$$

and the initial prescribed velocity \mathbf{u}_0 . The definition of the other variables appearing in the Navier-Stokes system are identical to the ones in the Stokes system (21).

Multiplication of the field equations (40) with the test functions, \mathbf{v} and q , and subsequent application of the divergence theorem yields the corresponding weak form

$$\begin{aligned} \int_{\Omega} \rho \frac{D\mathbf{u}}{Dt} \cdot \mathbf{v} d\Omega + \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}, p) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega \\ + \int_{\Omega} q(\nabla \cdot \mathbf{u}) d\Omega - \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} \mathbf{t}(\mathbf{u}, p) \cdot \mathbf{v} d\Gamma = 0. \end{aligned} \quad (42)$$

Here, it is important to emphasise that the field \mathbf{v} does not fulfil any constraints on the Dirichlet boundary. This is contrary to what is assumed in the

derivation of the weak form used in conventional finite elements. Therefore, the integral over the boundary Γ_D is still present in the above result. The crucial step is now to suitably augment the weak form so that the Dirichlet boundary constraint $\mathbf{u} - \bar{\mathbf{u}} = \mathbf{0}$ on Γ_D is enforced in a weak sense. In the Nitsche method this is accomplished by adding the following two boundary terms to the weak form (42)

$$\frac{\mu\gamma}{h} \int_{\Gamma_D} \mathbf{v} \cdot (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma \quad \text{and} \quad - \int_{\Gamma_D} \mathbf{t}(\mathbf{v}, q) \cdot (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma, \quad (43)$$

where μ is the fluid viscosity, γ is a yet to be specified scalar parameter and h is the grid size parameter defined in (9). Adding those two terms and reordering the weak form according to unknown and prescribed quantities gives

$$\begin{aligned} & \int_{\Omega} \rho \frac{D\mathbf{u}}{Dt} \cdot \mathbf{v} d\Omega + \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}, p) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega + \int_{\Omega} q(\nabla \cdot \mathbf{u}) d\Omega \\ & - \int_{\Gamma_D} \mathbf{t}(\mathbf{u}, p) \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} \mathbf{t}(\mathbf{v}, q) \cdot \mathbf{u} d\Gamma + \frac{\mu\gamma}{h} \int_{\Gamma_D} \mathbf{v} \cdot \mathbf{u} d\Gamma \\ & = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} \mathbf{t}(\mathbf{v}, q) \cdot \bar{\mathbf{u}} d\Gamma + \frac{\mu\gamma}{h} \int_{\Gamma_D} \mathbf{v} \cdot \bar{\mathbf{u}} d\Gamma \end{aligned} \quad (44)$$

It is instructive to briefly review the two expressions added to the weak form. The first expression in (43) is of the same form as a typical penalty term for enforcing Dirichlet constraints. The second expression in (43) is the conjugate of the last integral in the weak form (42) and ensures that the augmented weak form (44) is symmetric with respect to \mathbf{u} and \mathbf{v} in the Stokes limit (i.e. for $D\mathbf{u}/Dt = \mathbf{0}$). Both terms in (43) are identical to zero for the analytic solution \mathbf{u} . For this reason, the method is inherently consistent. Note that the conventional penalty method is only consistent in the limit $\gamma \rightarrow \infty$ [11].

The augmented weak form (44) can be expanded to its final form by introducing the equations for the total time derivative (41) and the fluid stress tensor (22). The variational formulation of the Navier-Stokes equation can now be stated as: Find $(\mathbf{u}, p) \in [\mathcal{H}^1(\Omega)]^d \times \mathcal{Q}$ with $\mathbf{u}(\cdot, 0) = \mathbf{u}_0$ such that

$$\begin{aligned} & \rho \left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \right)_{\Omega} + \rho ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v})_{\Omega} + a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + b(\mathbf{u}, q) \\ & + \hat{a}(\mathbf{u}, \mathbf{v}) + \hat{a}(\mathbf{v}, \mathbf{u}) + \hat{b}(\mathbf{u}, q) + \hat{b}(\mathbf{v}, p) + \frac{\mu\gamma}{h} \langle \mathbf{u}, \mathbf{v} \rangle_{\Gamma_D} \\ & = \ell(\mathbf{v}) + \hat{a}(\mathbf{v}, \bar{\mathbf{u}}) + \hat{b}(\bar{\mathbf{u}}, q) + \frac{\mu\gamma}{h} \langle \bar{\mathbf{u}}, \mathbf{v} \rangle_{\Gamma_D} \end{aligned} \quad (45)$$

for all $(\mathbf{v}, q) \in [\mathcal{H}^1(\Omega)]^d \times \mathcal{Q}$ at all times $t > 0$. Here, the relevant function spaces are the standard Sobolev space $\mathcal{H}^1(\Omega)$ and the space of square-integrable functions \mathcal{Q} previously defined in the context of the Stokes system (25). In addition to the bilinear forms $a(\cdot, \cdot)$, $b(\cdot, \cdot)$, and the linear form $\ell(\cdot)$ defined in (24), the following bilinear forms

$$\hat{a}(\mathbf{u}, \mathbf{v}) = -2 \langle \mu \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \mathbf{n}, \mathbf{v} \rangle_{\Gamma_D} \quad \text{and} \quad \hat{b}(\mathbf{u}, p) = \langle p \mathbf{n}, \mathbf{u} \rangle_{\Gamma_D} \quad (46)$$

and scalar products

$$(\mathbf{u}, \mathbf{v})_{\Omega} = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, d\Omega \quad \text{and} \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\Gamma_D} = \int_{\Gamma_D} \mathbf{u} \cdot \mathbf{v} \, d\Gamma \quad (47)$$

are used. The variational statement (45) with function spaces independent of the boundary conditions will provide in the following the basis for formulating an immersed boundary finite element method.

4.2. Immersed b-spline finite elements

The previous sections introduced the techniques required for an immersed b-spline finite element, namely the construction of a stable b-spline basis on Cartesian grids and the weak enforcement of Dirichlet boundary conditions. We now are ready to formulate the immersed b-spline method and to discuss aspects of its computer implementation.

4.2.1. Implicit boundary representation

It is assumed that the problem domain is given in the form of a signed distance function, i.e. by means of an implicitly defined surface. This choice is mainly motivated by future applications of the developed method, which involve large deformations and topology changes. In comparison to parametric surface representations, implicit representations are better suited to such applications. For more details see our previous work [38, 39, 40].

The signed distance function (or level set function) describing the problem domain is defined as

$$\phi(\mathbf{x}, \Gamma) = \begin{cases} \text{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ -\text{dist}(\mathbf{x}, \Gamma) & \text{otherwise,} \end{cases} \quad (48)$$

where Ω is the physical problem domain and Γ its boundary. Here, $\text{dist}(\mathbf{x}, \Gamma) = \min_{\mathbf{y} \in \Gamma} |\mathbf{x} - \mathbf{y}|$ is the shortest distance between the point \mathbf{x} and the surface Γ . For complex domains usually only the level set values at the Cartesian grid points are given. The level set values at arbitrary locations are determined by linear interpolation of the grid point values.

In order to evaluate the bilinear and linear forms appearing in the weak form (45), it is necessary to have a parametric representation of the boundary. To this end, the boundary $\phi(\mathbf{x}, \Gamma) = 0$ is approximated with a simplicial complex consisting of either line segments (in 2D) or triangles (in 3D). Furthermore, in evaluating the domain integrals it is necessary to integrate over cut-cells, which are usually a general polytope. Since numerical integration schemes for such shapes are not available, the cut-cells are subdivided into simplices. For three-dimensional cut-cells we use a slightly modified version of the marching tetrahedra algorithm presented in [41]. The modification decreases the number of tetrahedra created during the subdivision process. This algorithm produces at the same time the triangles which yield a parametric representation of the zero

level set, i.e. $\phi(\mathbf{x}, \Gamma) = 0$. The evaluation of the surface integrals appearing in the augmented weak form (45) is thus reduced to integration over flat triangles (in 3D) or line segments (in 2D).

4.2.2. Discrete system of equations and its solution

The spatial discretization of the variational statement (45) is achieved using the finite element approximations (26) for the velocity and pressure spaces. This leads directly to the semi-discrete system of equations

$$\begin{aligned} \mathbf{M}\dot{\mathbf{u}} + \mathbf{N}(\mathbf{u})\mathbf{u} + \mathbf{A}\mathbf{u} + \hat{\mathbf{A}}\mathbf{u} + \hat{\mathbf{A}}^\top \mathbf{u} + \mathbf{C}\mathbf{u} + \mathbf{B}^\top \mathbf{p} + \hat{\mathbf{B}}^\top \mathbf{p} &= \hat{\mathbf{f}} \\ \mathbf{B}\mathbf{u} + \hat{\mathbf{B}}\mathbf{u} &= \hat{\mathbf{g}}. \end{aligned} \quad (49)$$

Here, \mathbf{M} is the mass matrix and $\mathbf{N}(\mathbf{u})$ represents the discretized advection term. The matrices \mathbf{A} and \mathbf{B} are as in the discrete Stokes system (27). Note that the pressure discretization is carried out on the coarse grid as described in Section 3.2. The matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ represent the corresponding discretizations of the bilinear forms \hat{a} and \hat{b} given in (46) and \mathbf{C} is the discrete penalty term. Moreover, the right hand side vectors $\hat{\mathbf{f}}$ and $\hat{\mathbf{g}}$ contain the prescribed force terms and the terms on the right hand side of (45) related to the given Dirichlet datum $\bar{\mathbf{u}}$.

The treatment of the nonlinear advection term and the temporal discretization of system (49) can be carried out by standard means. In the following, an Euler backward time discretization and a fixed point iteration are used for this purpose (see, e.g., [42] for details). Nevertheless, other choices are also possible.

4.2.3. Cut-cell stabilisation

In most immersed or embedded boundary methods, numerical accuracy and stability is compromised if exceedingly small cut-cells are present at the domain boundaries. This can be explained with the small contribution of the corresponding basis functions to the discretized governing equations (49), which results in ill-conditioned system matrices. Beyond that, as will be demonstrated in Section 4.3.1, cut-cells have an adverse effect on the choice of the penalty parameter γ in the Nitsche method.

The problems resulting from cut-cells can be remedied using the *extended* b-splines introduced by Höllig et al. [6, 14]. They revolve around the idea of extrapolating the solution field from inside to the outside of the domain using sufficiently high order Lagrange interpolation functions. To this end, the critical basis functions are first identified by computing the intersection area of each basis function B_i^{μ} with the physical domain $s_i = |\text{supp}(B_i^{\mu}) \cap \Omega|$. The b-splines with an intersection size $s_i \geq \alpha h^d$ are denoted as *interior*, in our implementation we chose $\alpha = 1$. The remaining b-splines with $0 < s_i < \alpha h$ are denoted as *exterior*. B-splines which lie entirely outside the physical domain, i.e. $s_i = 0$, are not relevant and can be discarded. In the following, the set of interior and exterior b-splines are denoted with \mathbb{I} and \mathbb{J} , respectively. With this

classification, the interpolation equation (16) can be rewritten as

$$u^h(\mathbf{x}) = \sum_{\mathbf{i} \in \mathbb{I}} B_{\mathbf{i}}^{\mu}(\mathbf{x}) u_{\mathbf{i}} + \sum_{\mathbf{j} \in \mathbb{J}} B_{\mathbf{j}}^{\mu}(\mathbf{x}) u_{\mathbf{j}} \quad (50)$$

where here and in the remainder of this section the interpolation of a generic scalar field u is considered. The extension to the case of vector fields is, as usual, accomplished by individually interpolating each component.

As discussed, the exterior b-splines have only a small contribution and lead to ill-conditioned system matrices. This can be remedied by expressing the coefficients of an exterior b-spline as a linear combination of the coefficients of interior b-splines

$$u_{\mathbf{j}} = \sum_{\mathbf{i} \in \mathbb{I}_{\mathbf{j}}} E_{\mathbf{i}\mathbf{j}} u_{\mathbf{i}}. \quad (51)$$

The weights $E_{\mathbf{i}\mathbf{j}}$ and the subset of indices $\mathbb{I}_{\mathbf{j}} \subset \mathbb{I}$ represent the extrapolation of the solution field u^h using Lagrange basis functions and will be specified below [6, 14]. Formally, inserting (51) into (50) yields an interpolation equation, which depends only on the coefficients of the interior b-splines

$$u^h(\mathbf{x}) = \sum_{\mathbf{i} \in \mathbb{I}} \left[B_{\mathbf{i}}^{\mu}(\mathbf{x}) + \sum_{\mathbf{j} \in \mathbb{J}_{\mathbf{i}}} E_{\mathbf{i}\mathbf{j}} B_{\mathbf{j}}^{\mu}(\mathbf{x}) \right] u_{\mathbf{i}}. \quad (52)$$

The index set $\mathbb{J}_{\mathbf{i}}$ is the reciprocal set to $\mathbb{I}_{\mathbf{j}}$ and includes for an interior b-spline all the exterior b-spline coefficients it influences. The modified interpolation equation (52) enables to formulate the finite element equations using only the interior b-splines so that the resulting equations are well conditioned.

It remains to specify the index set $\mathbb{I}_{\mathbf{j}}$ and the weights $E_{\mathbf{i}\mathbf{j}}$ used in the extrapolation of the coefficients (51). If the polynomial accuracy of a b-spline approximation of degree μ is to be preserved, the index set $\mathbb{I}_{\mathbf{j}}$ must include $\mu + 1$ coefficients per Cartesian direction giving $(\mu + 1)^d$ coefficients in the multi-dimensional case. As an additional condition, the indices in $\mathbb{I}_{\mathbf{j}}$ should be chosen as close as possible to index \mathbf{j} . In Figure 7 the implemented algorithm for choosing the $\mathbb{I}_{\mathbf{j}}$ for a particular external quadratic b-spline is illustrated. Although there is some freedom in the choice of $\mathbb{I}_{\mathbf{j}}$, formally starting from an index ℓ the remaining indices are given in dependence of the polynomial degree and dimension

$$\mathbb{I}_{\mathbf{j}} = \ell + [0, \dots, \mu]^d. \quad (53)$$

With this index set at hand, the weights $E_{\mathbf{i}\mathbf{j}}$ are computed by evaluating the corresponding Lagrange basis functions of degree μ at $\mathbf{j} - \ell$ [6]. In case of an equidistant grid this leads to the following equation

$$E_{\mathbf{i}\mathbf{j}} = \prod_{\alpha=1}^d \left(\prod_{\substack{\nu=0 \\ \nu \neq i^{\alpha} - \ell^{\alpha}}}^{\mu} \frac{j^{\alpha} - \ell^{\alpha} - \nu}{i^{\alpha} - \ell^{\alpha} - \nu} \right) \quad (54)$$

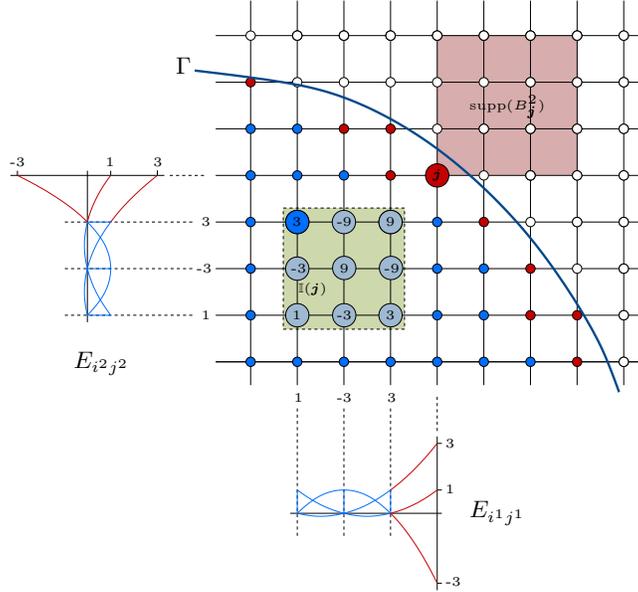


Figure 7: Computation of the weight coefficients E_{ij} for a quadratic exterior b-spline B_j^2 . The set of interior indices $\mathbb{I}(j) = (j^1 - 3, j^2 - 3) + [0, \dots, 2]^2$ is used for extrapolating the solution field. The weights are determined as the tensor product of the Lagrange extrapolation coefficients in the both coordinate directions, i.e. $\{E_{ij}\} = [1, -3, 3] \times [1, -3, 3]$.

wherein the outer product represents the tensor-product over the Cartesian directions α . The bracket expression is the Lagrange polynomial of degree μ in direction α corresponding to the index $i^\alpha - \ell^\alpha$ evaluated at the index $j^\alpha - \ell^\alpha$. Figure 7 provides a graphic illustration of this equation in case of a quadratic b-spline.

It is straightforward to use the new basis functions in the solution procedure by inserting the modified interpolation equation (52) into the weak form. As shown by Höllig et al. [6, 14], the modified basis functions inherit most of the b-spline properties and lead to interpolation estimates with the same convergence order as the original b-spline basis. In terms of implementation, the element vectors and matrices are first computed without considering the weights E_{ij} . Subsequently, only during the assembly procedure the element contributions need to be multiplied with E_{ij} to establish the global system vectors and matrices. Note that the flow of computations is identical to the subdivision stabilisation procedure introduced in Section 3. In practice, exploiting this procedural similarity during software implementation leads to fast and compact solvers. Furthermore, the similarities between the two procedures justify to regard the introduced cut-cell treatment as a preconditioning or stabilisation step.

4.3. Verification examples

In this section we assess the robustness and accuracy of the presented immersed method by means of several numerical examples. The solution of two Poisson problems and a stationary Navier-Stokes problem are considered.

4.3.1. Study of penalty parameter and cut-cell sizes

The primary purpose of this example is to study the influence of the penalty parameter and cut-cell size on the conditioning of the system matrix. The considered computational domain $\Omega_{\square} = (-1, 1) \times (-1, 1)$ contains a straight immersed Dirichlet boundary located at $x^1 = \delta$ and parallel to the x^2 -axis. The remaining boundaries are homogeneous Neumann boundaries. The location of the Dirichlet boundary is chosen so that $\delta > 0.75$. The physical domain is hence given by $\Omega = (-1, \delta) \times (-1, 1)$. This simple setup has been chosen in order to provide control of the relative area of cut-cells defined as $\varepsilon_{\mathbf{k}} = |\omega_{\mathbf{k}} \cap \Omega|/h$, where $\omega_{\mathbf{k}}$ denotes the domain of the cell with the index \mathbf{k} and the grid size h . In the chosen setup $\varepsilon = \varepsilon_{\mathbf{k}}$ is constant for all cut-cells.

In a first step, the positivity of the system matrix of the discretized Laplace problem with Dirichlet conditions along Γ_D is studied. In order to ensure positivity, the penalty parameter has to be larger than some critical value, i.e., $\gamma > \gamma_c$, see [34]. Figure 8a depicts for linear b-splines the dependency of the minimal eigenvalue of the system matrix on the penalty parameter for different selected cut cell sizes. It is clear that a positive minimal eigenvalue $\lambda_{\min} > 0$ is obtained for specific values $\gamma > \gamma_c$, where the threshold value γ_c increases with decreasing cut-cell size ε . Plotting the threshold value γ_c against the cut-cell size for b-spline degrees $\mu = 1, 2$ and 3 , yields the curves shown in Figure 8b. Note that even in the limit $\varepsilon \rightarrow 0$ a relatively small value of γ_c is sufficient to ensure positivity. This desirable property is due to the cut-cell stabilisation presented in Section 4.2.3. Without such a stabilisation the penalty parameter diverges (i.e. $\lim_{\varepsilon \rightarrow 0} \gamma_c = \infty$) leading to an ill-conditioned system matrix.

Moreover, the reported relatively small lower bounds for the penalty parameter are insensitive to grid size. This is illustrated in Figure 9, which shows the variation of the minimal eigenvalue λ_{\min} with cut-cell size ε for three different grid sizes $h = 2/N$ and a fixed penalty parameter $\gamma = 10$. For the considered linear and quadratic b-splines the minimal eigenvalue is practically independent of the cut-cell size.

Finally, we demonstrate that the cut-cell size has an impact on the finite element error in L_2 - and H^1 -norms but does not affect the convergence order. To this end, the convergence of a Poisson problem with the analytic solution

$$u(\mathbf{x}) = (x^1 - 4)^{-1} + x^1/4 \quad (55)$$

is considered. The errors in the L_2 - and H^1 -norms are plotted in Figure 10 for linear and quadratic b-splines. In both cases, the solution converges with optimal order and exhibits only for coarse meshes a slight dependence on the cut-cell size.

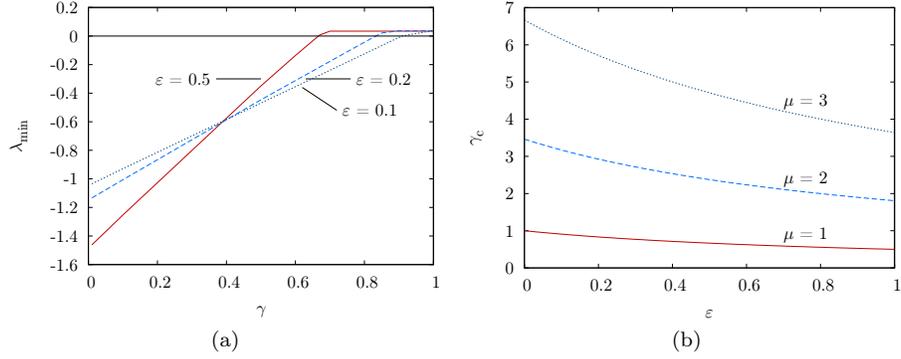


Figure 8: Positivity of the system matrix. (a) Minimal eigenvalue λ_{\min} for three different relative cut-cell areas ϵ and different penalty parameters γ . (b) Critical penalty parameter for linear ($\mu = 1$), quadratic ($\mu = 2$) and cubic ($\mu = 3$) b-splines for different relative cut-cell areas ϵ .

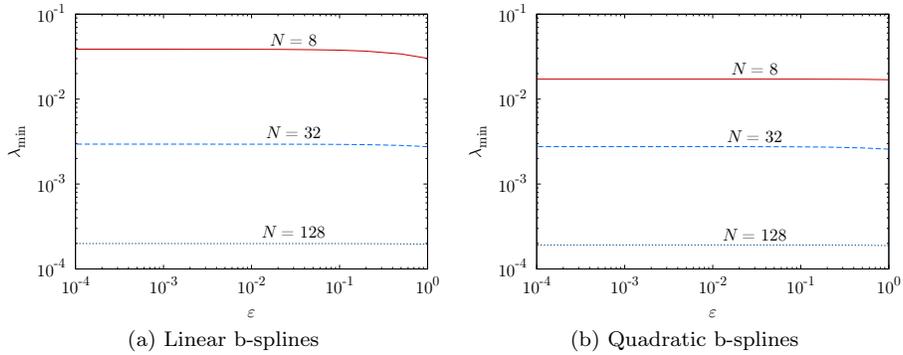


Figure 9: Minimal eigenvalue λ_{\min} vs. relative cut-cell area ϵ for $\gamma = 10$ and three different grids with $N = 8, 32,$ and 128 points.

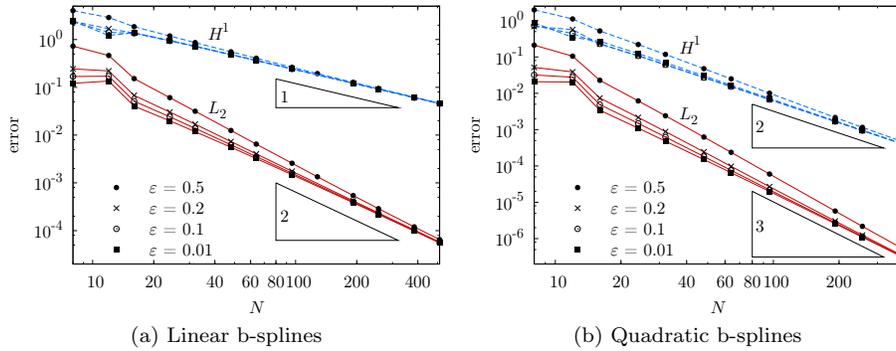


Figure 10: Convergence of the error in the L_2 -norm and the H^1 -seminorm in dependence of the cut-cell size ε .

4.3.2. Poisson equation in a sphere

In this three-dimensional example the solution of the Poisson equation in a unit sphere with $\Omega = \{\mathbf{x} \in \mathbb{R}^3 \mid |\mathbf{x}| < 1\}$ is considered. The loading and boundary conditions are chosen so that they satisfy the analytic solution

$$u(\mathbf{x}) = |\mathbf{x} - \mathbf{x}^*|^{-1} \quad \text{with} \quad \mathbf{x}^* = (1, 1, 1). \quad (56)$$

The plot in Figure 11a shows the computational domain $\Omega_{\square} = (-1.1, 1.1)^3$ coloured with the iso-contours of the level set function and the boundary of the physical domain. The iso-contours of the computed solution on the domain boundary are shown in Figure 11b. In the computation, a penalty parameter

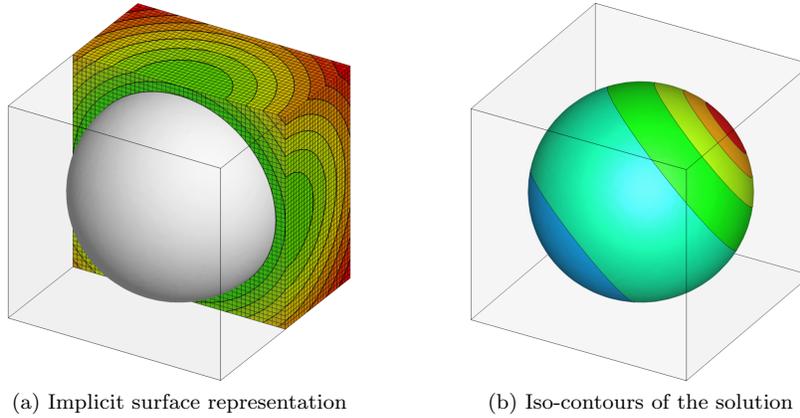


Figure 11: Poisson equation in a sphere discretized with a grid with 48^3 cells.

$\gamma = 20$ is used and the level set function representing the sphere on the fluid grid has been obtained from a surface mesh with 290.068 linear triangles using

the closest point transform algorithm described in [39]. Figure 12a shows the convergence in the L_2 -norm and the H^1 -seminorm against the grid parameter N . Although the absolute values of the errors are lower for quadratic than for linear b-splines, only second-order convergence in the L_2 -norm and first-order in the H^1 -seminorm can be observed. This reduced convergence order is due to the approximation of domain boundaries with linear triangles during the integration, see Section 4.2.1. It is not an inherent limitation of the method and can be remedied by interpolating the boundaries with higher order triangles.

In addition to the errors in the domain, the convergence of the integral surface flux F is considered, see Figure 12b. Again, the rate of convergence for linear and quadratic b-splines is similar but the absolute error is significantly smaller for the quadratic b-splines.

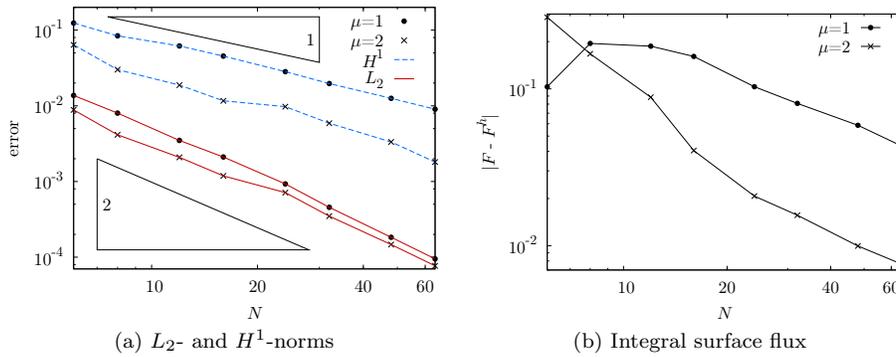


Figure 12: Poisson equation in a sphere. Convergence of errors for linear ($\mu = 1$) and quadratic ($\mu = 2$) b-splines.

4.3.3. Stationary Navier-Stokes equation in an irregular domain

To demonstrate the applicability of the proposed method to viscous incompressible fluid flow, we now consider the solution of the stationary Navier-Stokes equations. The corresponding weak form is obtained by neglecting the inertia terms in (45). Similar to the previous examples, the force term \mathbf{f} and Dirichlet datum $\bar{\mathbf{u}}$ are determined so that they correspond to the chosen solenoidal analytic solution with

$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} -\cos(5x^1)\sin(5x^2) \\ \sin(5x^1)\cos(5x^2) \end{pmatrix}, \quad p(\mathbf{x}) = -\frac{1}{4}(\cos(10x^1) + \cos(10x^2)) \quad (57)$$

and the viscosity $\mu = 0.001$. Figure 13 shows the geometry of the considered domain together with the computational domain $\Omega_{\square} = 0.9 \times 0.9$ and its discretization with $N = 32$ cells per direction. In total 4.148 line segments with a linear interpolation of $\bar{\mathbf{u}} = (\mathbf{u})|_{\Gamma}$ are used for the discretization of the boundary curve.

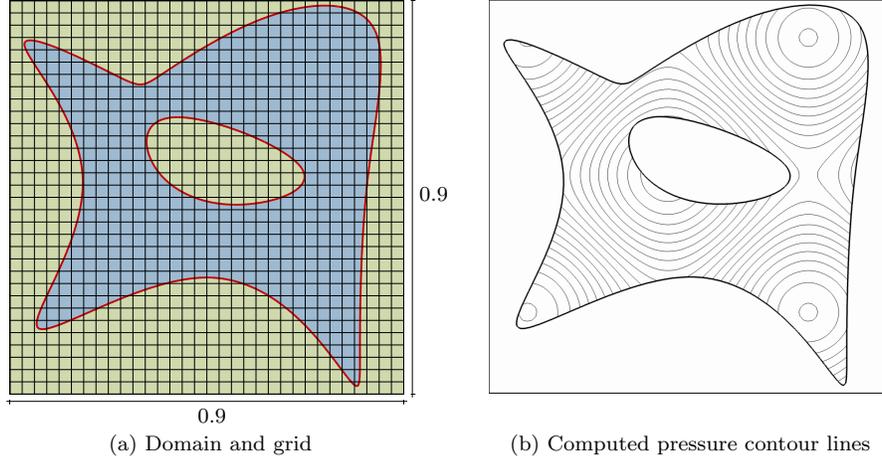


Figure 13: Embedded irregular domain problem with a grid consisting of $N = 32$ points per direction.

The convergence behaviour is depicted in Figure 14 where the b-spline combinations for velocity and pressure interpolation $(\mu_u, \mu_p) = (1, 0), (1, 1), (2, 1)$ and $(2, 2)$ are considered. For all but the lowest order combination, the convergence orders in the L_2 -norm of the velocity and pressure fields are of second order. This outcome clearly demonstrates that the performance of the proposed method is not affected by the non-linearity of the fluid problem.

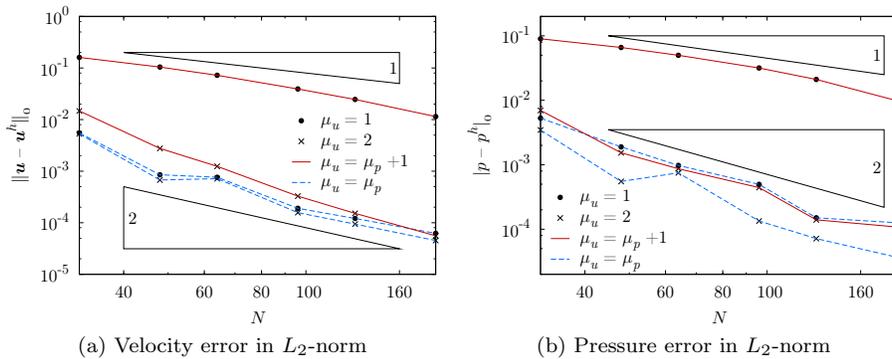


Figure 14: Immersed irregular domain. Convergence of the errors in velocity and pressure fields for different combinations of polynomial degrees $\mu_u - \mu_p$.

5. Incompressible flow applications

In this section, the presented method is applied to several incompressible viscous flow problems with complex boundary geometries. In all nonstationary cases, the time discretization is carried out using the Euler backward method. The nonlinearity of the convection term is resolved by means of a fixed point iteration. The force resultants are obtained by numerically integrating the boundary tractions, i.e. $\mathbf{F} = \int_{\Gamma} \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n} \, d\Gamma$. The fluid density is chosen as $\rho = 1$ and the fluid viscosity is determined so that a prescribed Reynolds number $Re = (\rho UL)/\mu$ is obtained. Here, U is the characteristic fluid speed (e.g. the freestream velocity) and L is the characteristic length of the solid.

5.1. Flow around a fixed cylinder

The two-dimensional flow around a stationary cylinder at different Reynolds numbers based on the cylinder's diameter is considered. The problem setup and the boundary conditions are described in Figure 15. The flow at three different Reynolds numbers $Re = 20$, $Re = 40$ and $Re = 80$, based on the cylinder's diameter, is investigated. In the $Re = 80$ case, the Reynolds number is large enough that periodic vortex shedding may occur. Hence, a transient analysis with a time step size of $\Delta t = 0.1$ is carried out. In the other two cases, a stationary analysis is sufficient because only a steady recirculation zone downstream of the cylinder is to be expected.

For the discretization of the velocity and pressure fields, the b-spline combinations $\mu_u = \mu_p = k$ with $k = 1$ and $k = 2$ are used. In the linear case a grid with 480×320 cells and in the quadratic case a grid with 360×240 cells is employed. Figure 15b shows a detail of the linear grid in the vicinity of the cylinder.

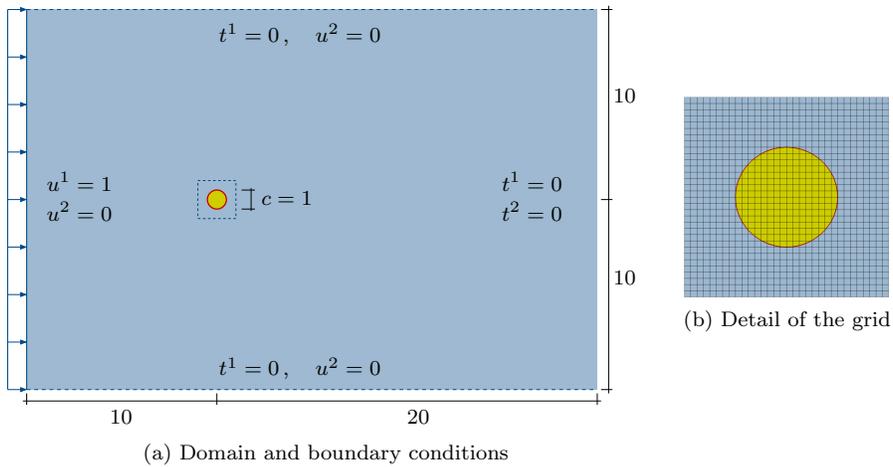


Figure 15: Problem description of flow around a fixed cylinder. The detail in (b) represents a box with $(-1, 1) \times (-1, 1)$ around the cylinder.

Figure 16 shows the computed streamlines and the contour lines for pressure and vorticity for the three different Reynolds numbers. For the two lower Reynolds numbers, $Re = 20$ and $Re = 40$, the stationary solution is plotted. The results for $Re = 80$ refer to time $t = 150$.

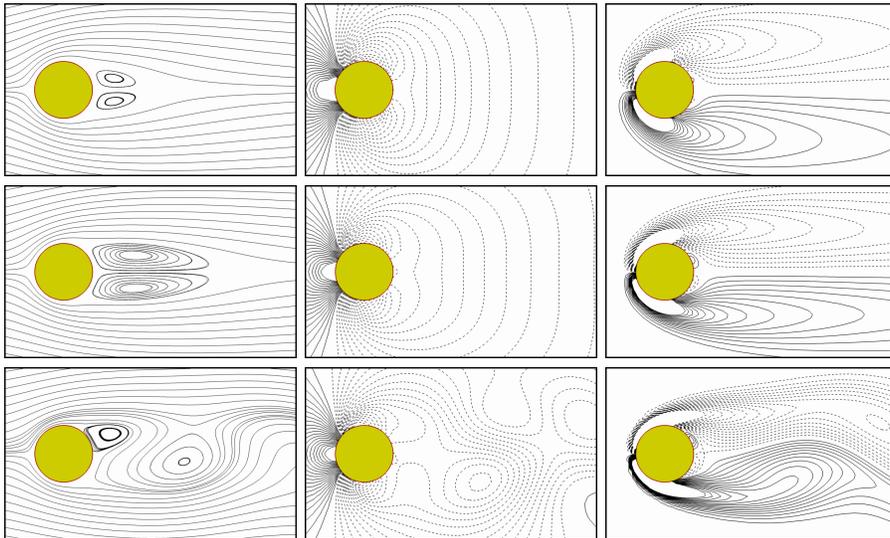


Figure 16: Flow around a fixed cylinder. Flow and pressure field in a box of size $(-1, 3) \times (-1.5, 1.5)$. From left to right: streamlines, pressure contour lines and vorticity contour lines. From top to bottom: $Re = 20$, 40 and 80. The pressure lines range from -0.5 to 0.5 with an interval of $\Delta p = 0.025$ and the vorticity lines range from -3 to 3 with an interval of $\Delta \omega = 0.2$. The lines of negative values are dashed and the zero contours are omitted.

In order to assess the quality of the numerical results several derived quantities are compared with published results. In all three cases, the drag coefficient defined with reference to the diameter, $C_D = 2F^1/(|\mathbf{u}_\infty|^2 c)$, is computed. In the stationary cases, the relative length of the recirculation downstream of the cylinder, L_r/c , and the angle of separation measured from the downstream stagnation point are of interest. On the other hand, the temporal variation of the drag and lift coefficients and the Strouhal number St are of particular interest for the transient case.

The temporal variation of the drag C_D and lift C_L coefficients are plotted in Figure 17 for the time interval $0 < t < 160$. Qualitatively the curves in Figure 17 agree well with those given in [43]. However, the amplitude of our computed lift coefficient is lower and the drag coefficient is also slightly different. These deviations are believed to be due to the applied boundary conditions which represent an approximation to the uniform far field flow. The domain size and boundary conditions we chose in our computations are different from [43]. Apart from that there is a good agreement between the derived quantities as shown

in Table 1 where the values are compared with the results of references [44] and [45].

Comparing the quality of the results for the different b-spline degrees, it is clear that the prediction of the separation angles is significantly better for the quadratic b-splines. Also, the drag coefficient in the transient case is more accurate with respect to the reference value. The other quantities have a similar precision in both cases. It has to be emphasised that the grid used for the linear b-splines is by a factor $3/2$ finer than for quadratic b-splines.

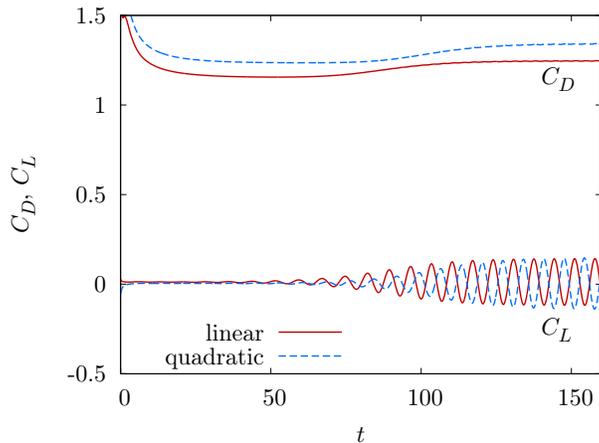


Figure 17: Flow around a fixed cylinder. Drag and lift coefficients vs. time for $Re = 80$.

Re	C_D	L_r/c	φ_s	St
20	2.11, 2.16 (2.03)	0.93, 0.93 (0.92)	38.7° , 41.9° (43.8°)	—
40	1.54, 1.60 (1.52)	2.23, 2.25 (2.27)	48.6° , 51.6° (53.8°)	—
80	1.24, 1.34 (1.38)	—	—	0.15, 0.15 (0.15)

Table 1: Flow around a fixed cylinder. Computed drag coefficient C_D , relative recirculation length L_r/c , separation angle φ_s and Strouhal number St . For each quantity, the first and second values refer to linear and quadratic b-splines, respectively, and the values in parentheses are from [44] and [45].

5.2. Moving cylinder

We now reverse the relative motion by pushing the cylinder with the diameter $c = 1$ through the fluid which is initially at rest. A constant horizontal acceleration of 0.2 is applied for $t < 5$ until the cylinder reaches its terminal velocity $\mathbf{u}_\infty = (1, 0)^T$. This velocity is then held constant for the rest of the computation. All outer boundaries are traction-free and a no-slip condition along the cylinder's surface is applied. Figure 18 shows the physical domain and

defines eight time instants used in the discussions to follow. The investigated Reynolds numbers are, in line with the previous example, $Re = 20, 40,$ and 80 . The spatial discretization consists of 640×160 cells with linear b-splines and the time step size is $\Delta t = 0.025$ for the cases $Re = 20$ and 40 , and $\Delta t = 0.0125$ for the case $Re = 80$.

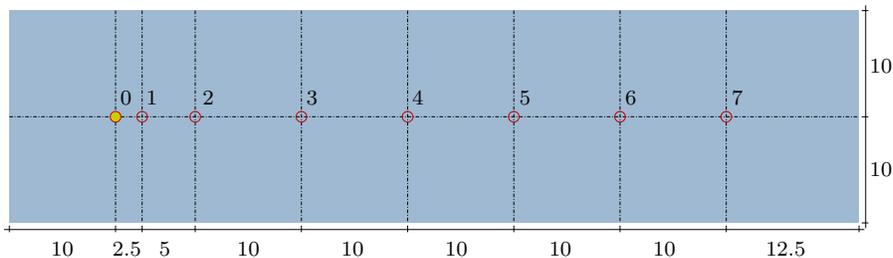


Figure 18: Problem description of the moving cylinder. The circles indicate the time instants referred to in Figure 19. The cylinder’s initial position 0 is at $t = 0$, position 1 is directly after the acceleration ceases at $t = 5$, and further positions, 2–7, are at $t \in \{10, 20, 30, 40, 50, 60\}$.

The snapshots of the streamlines, pressure contour lines and vorticity contour lines for the positions 1 to 7 (as defined in Figure 18) for $Re = 80$ are given in Figure 19. In the snapshots the build-up of vortices and their periodic shedding is clearly visible as the motion progresses. First at time instant four a slight deviation from symmetry starts to appear in the pressure and vorticity snapshots. This disturbance becomes more pronounced at the fifth time instant when the first closed vorticity contour lines are detached from the cylinder. This shedding of vortices continues until the last time instant which clearly displays the features of a developed von Kármán vortex street [46].

The development of lift and drag coefficients over time are given in Figure 20. The plot includes the drag coefficient for all three considered Reynolds numbers and the lift coefficient for $Re = 80$. The lift coefficients for $Re = 20$ and $Re = 40$ are identical to zero because of the symmetry of the flow field. The oscillation in the lift curve, which corresponds to the observed vortex shedding, is clearly visible. Quantitatively speaking, the final values of the drag coefficient C_D are all about 10% below the values obtained in Section 5.1 for the fixed cylinder. The reasons for these deviations are the coarser grid used in this set of computations and the different boundary conditions. Moreover, in the case of $Re = 80$ the flow field is still not fully developed to a quasi-static state as indicated by the slightly increasing drag curve (note the different time axes in Figures 17 and 20).

5.3. Flow around a sphere

As a three-dimensional example, we analyse the steady flow around a sphere. The physical domain has the dimensions $(-2, 10) \times (-4, 4)^2$ and contains a sphere with unit radius centred at the origin. The sphere surface is represented with 290.068 linear triangles. The prescribed boundary conditions are unit

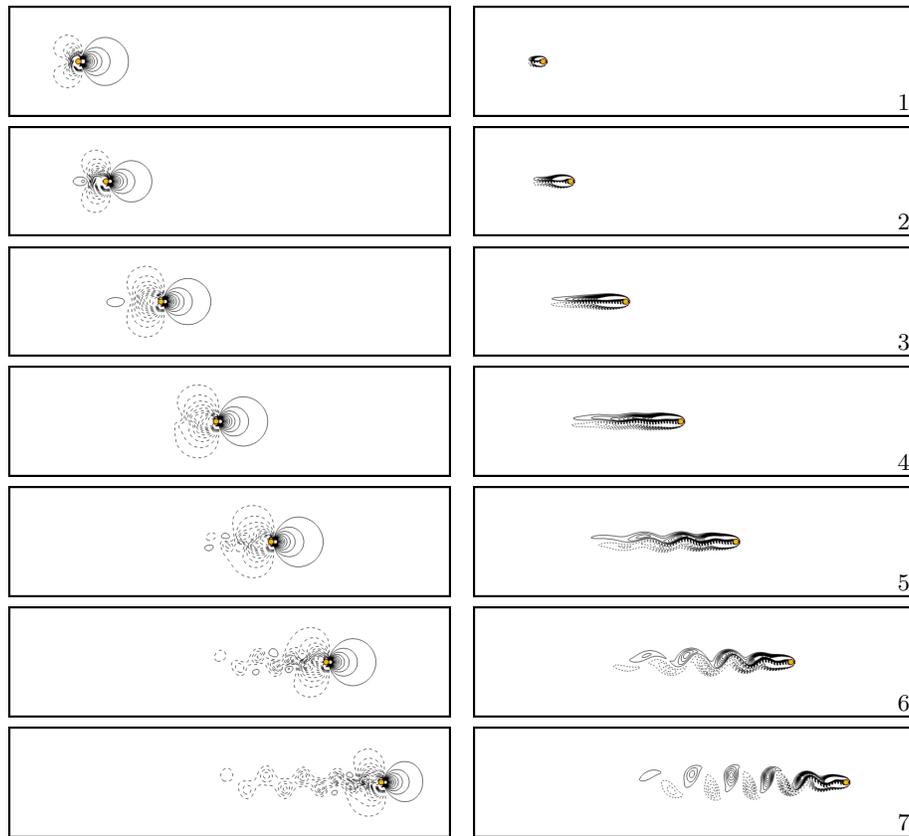


Figure 19: Moving cylinder at $Re = 80$. Top to bottom: the seven time instants as defined in Figure 18. Left: pressure contour lines. Right: vorticity contour lines.

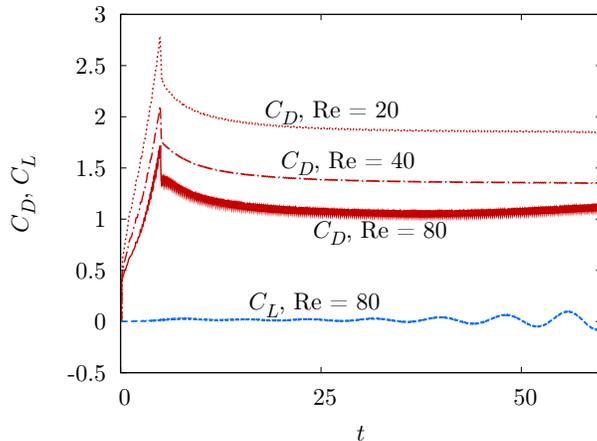


Figure 20: Moving cylinder. Drag and lift coefficients vs. time for three different Reynolds numbers. Note that for $Re = 20$ and $Re = 40$ the lift is zero.

inflow at $x^1 = -2$ and zero traction at all other boundaries. The finite element discretization of the domain consists of linear b-splines and 60×40^2 cells. The fluid viscosity is adjusted in order to achieve Reynolds numbers 40, 57, 80 and 113. Figure 21 shows the physical domain and the sphere together with some streamlines for $Re = 113$. One can clearly identify the recirculation zone downstream of the cylinder.

Figure 22 displays the computed relative recirculation lengths L_r/c . This plot also contains the line $L_r/c = 0.66 \log(Re) - 2.11$ which is taken from the experimental results in [47]. Despite the deviation from the reference results which are likely due to the coarse mesh and small size of the chosen physical domain, the qualitative behaviour is well captured.

5.4. Rotating ellipse

Finally, we study the flow induced by an elliptic body subject to eccentric rotation. The problem setup and the boundary conditions are described in Figure 23. It shows the ellipse of length $c = 2$ and 15% thickness in its initial position (horizontal) and the two extremal positions ($\alpha = \pm\pi/4$). The eccentricity of the rotation is $\delta = c/8$. The rotation itself is described by variation of the angle $\alpha(t) = \frac{\pi}{4} \sin(2\pi t)$. The Reynolds number (based on c and the maximal tip velocity of the ellipse) assumes the value $Re = 333.3$. The finite element discretization consists of linear b-splines and 900×450 cells, and the time step size is $\Delta t = 0.01$.

Figure 24 shows the temporal behaviour of the force components F^1 and F^2 for $0 < t < 4$ together with the angle $\alpha(t)$. The vorticity fields at the four time instants defined in Figure 24 are plotted in Figure 25. These snapshots show the vortex shedding at the right edge of the body as it occurs in each upward

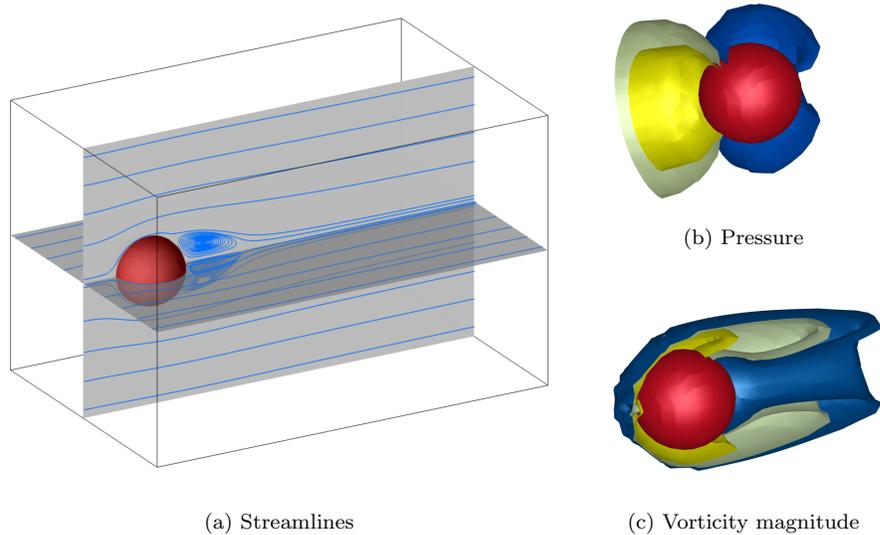


Figure 21: Flow around a sphere. Representative results for $Re = 113$.

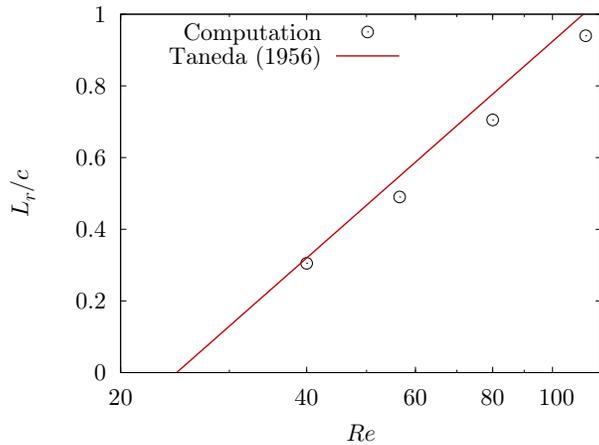


Figure 22: Flow around a sphere. Comparison of computed recirculation lengths with experimental results from [47].

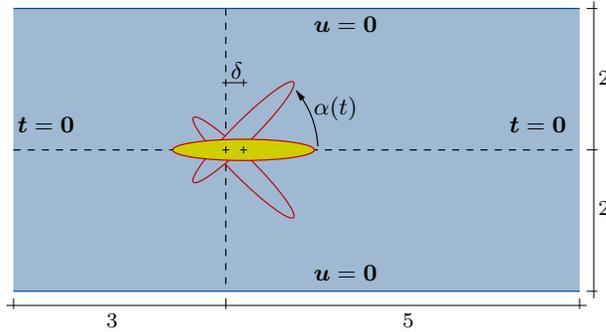


Figure 23: Problem description of the rotating ellipse.

and downward stroke. Also, minor vortices are released at the left edge which is subject to a smaller amplitude due to the eccentricity of the motion.

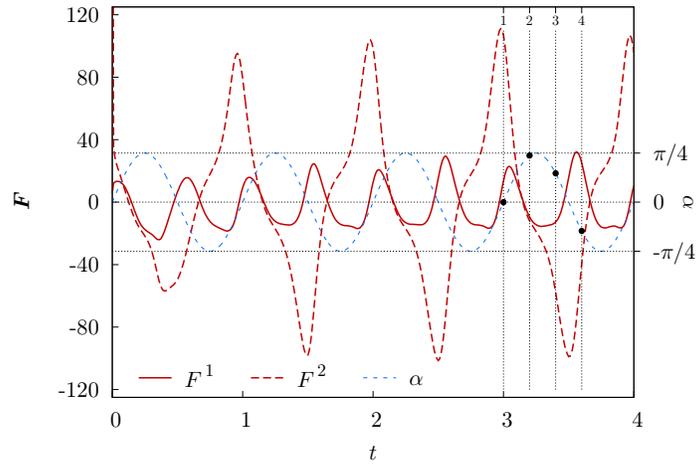


Figure 24: Rotating ellipse. Horizontal and vertical forces acting on the ellipse and the prescribed angle α vs. time. The indicated four time instants refer to the snapshots in Figure 25.

This example clearly demonstrates the advantage of an immersed method as compared to a conventional arbitrary Lagrangian-Eulerian (ALE) approach using body-fitted meshes. The distortion of a body-fitted mesh in the extreme positions of the ellipse would require dedicated mesh updating techniques; in the present approach such situations do not present a difficulty.

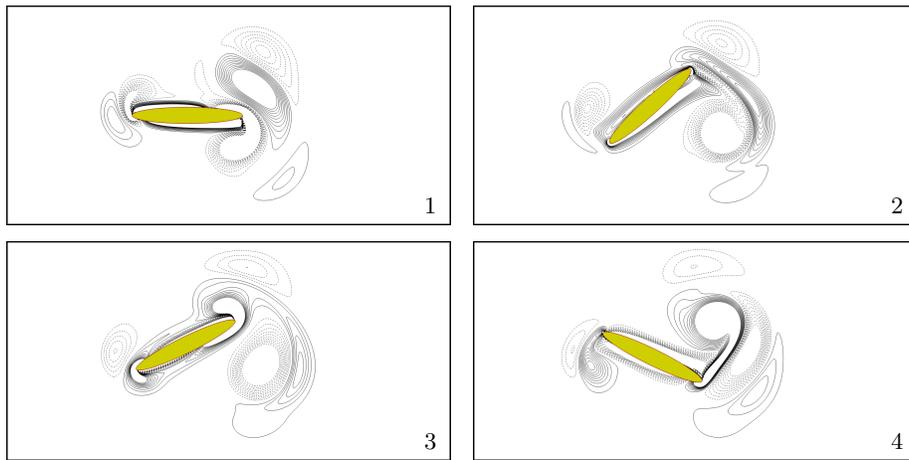


Figure 25: Rotating ellipse. Vorticity contour lines at the four time instants defined in Figure 24.

6. Conclusions

B-splines are particularly well suited for finite element applications that benefit from basis functions with higher degree and continuity. In this work, the refinability of b-splines or, in other terms, the two-scale relation for uniform b-splines is used to devise a new family of inf-sup stable finite elements for incompressible flows. Conveniently, the velocity and pressure fields are interpolated on the same grid using b-spline basis functions with same or different polynomial degrees. The element matrices and vectors are multiplied prior to the assembly with a subdivision matrix in order to stabilise the elements by projecting the pressure interpolation to the coarse grid. It is worth repeating that the proposed subdivision stabilisation procedure crucially relies on the refinability of b-splines and is not feasible, for example, using higher degree Lagrange basis functions. Although an analytic stability proof for the presented family of b-spline elements is not available, the performed numerical inf-sup test and the presented examples demonstrate their soundness. Further, an immersed type approach was developed to compute problems with arbitrary topology irrespective of b-spline basis functions restriction to grids with tensor-product structure. The proposed combination of the Nitsche method with the cut-cell stabilisation procedure for enforcing Dirichlet boundary conditions ensures that the penalty parameter remains always bounded. Indeed, all the presented examples were computed using one single penalty parameter.

In closing, we note some of the limitations of the presented method and directions for future development. In the introduced implementation, the domain boundaries are represented using a scalar signed distance function on the Cartesian grid. Although this implicit geometry representation leads to robust algorithms, especially in presence of topology and large geometry changes, it can only represent the domain boundaries with polygons in two dimensions and triangles in three dimensions. This impedes the convergence order of the method and leads to suboptimal results. Therefore, in applications with more stringent accuracy requirements it is advisable to represent the boundary using either NURBS or subdivision surfaces [15, 48]. This however raises the question of efficient integration of cut-cells, which can be, for instance, accomplished in line with the integration approach developed in [49]. In this context, the ability to consider NURBS boundaries can facilitate the isogeometric analysis of problems directly based on non-watertight CAD representations. Finally, for efficiency reasons it is desirable to have the possibility of local h -refinement in order, for example, to resolve boundary and shear layers more efficiently. To this end, the most promising h -refinement techniques are the t-splines [27] and hierarchical b-splines [6], which incidentally also rely on the two-scale relation for uniform b-splines.

Acknowledgement

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

References

- [1] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annual Review of Fluid Mechanics* 37 (2005) 239–261.
- [2] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche’s method, for elliptic interface problem, *Computer Methods in Applied Mechanics and Engineering* 191 (2002) 5537–5552.
- [3] A. Embar, J. Dolbow, I. Harari, Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements, *International Journal for Numerical Methods in Engineering* 83 (2010) 877–898.
- [4] A. Gerstenberger, W. A. Wall, An embedded Dirichlet formulation for 3D continua, *International Journal for Numerical Methods in Engineering* 82 (2010) 537–563.
- [5] T. Rüberg, F. Cirak, An immersed finite element method with integral equation correction, *International Journal for Numerical Methods in Engineering* 86 (2011) 93–114.
- [6] K. Höllig, *Finite Element Methods with B-splines*, SIAM Frontiers in Applied Mathematics, 2003.
- [7] R. A. K. Sanches, P. B. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains, *Computer Methods in Applied Mechanics and Engineering* 200 (2011) 1432–1445.
- [8] M. Arroyo, M. Ortiz, Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods, *International Journal for Numerical Methods in Engineering* 65 (2006) 2167–2202.
- [9] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, J. Periaux, 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 (2001) 363–426.
- [10] A. Gerstenberger, W. A. Wall, An eXtended Finite Element Method/Lagrange multiplier based approach for fluid-structure interaction, *Computer Methods in Applied Mechanics and Engineering* 197 (2008) 1699–1714.

- [11] J. Nitsche, Über ein Variationsverfahren zur Lösung von Dirichlet-Problemen bei der Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36 (1971) 9–15.
- [12] H. Wang, T. Belytschko, Fluid-structure interaction by the discontinuous-Galerkin method for large deformations, *International Journal for Numerical Methods in Engineering* 77 (2009) 30–49.
- [13] P. Hansbo, E. Burman, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, *Applied Numerical Mathematics* (in press).
- [14] K. Höllig, U. Reif, W. J., Weighted extended b-spline approximation of dirichlet problems, *SIAM Journal on Numerical Analysis* 39 (2001) 442–462.
- [15] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4135–4195.
- [16] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, *International Journal for Numerical Methods in Engineering* 47 (2000) 2039–2072.
- [17] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. R. Hughes, Isogeometric analysis of structural vibrations, *Computer Methods in Applied Mechanics and Engineering* 195 (2006) 5257–5296.
- [18] J. A. Evans, Y. Bazilevs, I. Babuska, T. J. R. Hughes, n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method, *Computer Methods in Applied Mechanics and Engineering* 198 (2009) 1726–1741.
- [19] L. Piegl, W. Tiller, *The NURBS book*, Monographs in visual communication, Springer, 1997.
- [20] C. de Boor, *A practical guide to splines*, Springer, 2001.
- [21] G. Farin, *Curves and Surfaces for CAGD*, Academic Press, 2002.
- [22] H. Prautzsch, W. Boehm, M. Paluszny, *Bezier and B-Spline Techniques*, Springer Verlag, 2002.
- [23] D. Zorin, P. Schröder, *Subdivision for Modeling and Animation*, SIGGRAPH 2000 Course Notes, 2000.
- [24] J. Peters, U. Reif, *Subdivision Surfaces*, Springer Series in Geometry and Computing, Springer, 2008.

- [25] D. R. Forsey, R. H. Bartels, Hierarchical b-spline refinement, *Computer Graphics* 22 (1988) 205–212.
- [26] D. Schillinger, E. Rank, 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* 200 (2011) 3358–3380.
- [27] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, in: *SIGGRAPH 2003 Conference Proceedings*, San Diego, CA, 477–484, 2003.
- [28] M. Sabin, *Analysis and Design of Univariate Subdivision Schemes*, Springer, 2010.
- [29] F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, 1991.
- [30] Y. Bazilevs, L. Beirao de Veiga, J. Cottrell, T. J. R. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, *Mathematical Methods and Models in Applied Sciences* 16 (2006) 1031–1090.
- [31] A. Buffa, C. de Falco, G. Sangalli, Isogeometric Analysis: Stable elements for the 2D Stokes equation, *International Journal for Numerical Methods in Fluids* 65 (2011) 1407–1422.
- [32] F. Brezzi, R. S. Falk, Stability of higher-order Hood-Taylor methods, *SIAM Journal on Numerical Analysis* 28 (1991) 581–590.
- [33] A. Ern, J.-L. Guermond, *Theory and Practice of Finite Elements*, Springer, 2003.
- [34] R. Stenberg, On some techniques of approximating boundary conditions in the finite element method, *Journal of Computational and Applied Mathematics* 63 (1995) 139–148.
- [35] J. Freund, R. Stenberg, On weakly imposed boundary conditions for second order problems, in: *Proceedings of the International Conference on Finite Elements in Fluids*, Venezia, 327–336, 1995.
- [36] R. Becker, Mesh adaptation for Dirichlet flow control via Nitsche’s method, *Communications in Numerical Methods in Engineering* 18 (2002) 669–680.
- [37] S. Fernandez-Mendez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, *Computer Methods in Applied Mechanics and Engineering* 193 (2004) 1257–1275.
- [38] F. Cirak, R. Radovitzky, Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets, *Computers & Structures* 83 (2005) 491–498.

- [39] F. Cirak, R. Deiterding, S. P. Mauch, Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations, *Computers & Structures* 85 (2007) 1049–1065.
- [40] K. Karagiozis, R. Kamakoti, F. Cirak, C. Pantano, A computational study of supersonic disk-gap-band parachutes using Large-Eddy Simulation coupled to a structural membrane, *Journal of Fluids and Structures* 27 (2011) 175–192.
- [41] A. Gueziec, R. Hummel, Exploiting triangulated surface extraction using tetrahedral decomposition, *IEEE Transactions on Visualization and Computer Graphics* 1 (1995) 328–342.
- [42] S. Turek, *Efficient solvers for incompressible flow problems*, Springer, 1999.
- [43] J.-E. Emblemsvåg, R. Suzuki, G. V. Candler, Cartesian grid method for moderate-Reynolds-number flows around complex moving objects, *AIAA Journal* 43 (2005) 76–86.
- [44] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *Journal of Computational Physics* 156 (1999) 209–240.
- [45] M.-H. Wu, C.-Y. Wen, R.-H. Yen, M.-C. Weng, A.-B. Wang, Experimental and numerical study of the separation angle for flow around a circular cylinder at low Reynolds number, *Journal of Fluid Mechanics* 515 (2004) 223–260.
- [46] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.
- [47] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, *Journal of the Physical Society of Japan* 11 (1956) 1104–1108.
- [48] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision, *Computer-Aided Design* 34 (2002) 137–148.
- [49] R. Sevilla, S. Fernandez-Mendez, A. Huerta, NURBS-enhanced finite element method (NEFEM), *International Journal for Numerical Methods in Engineering* 76 (2008) 56–83.