Accepted Manuscript

A stabilised immersed boundary method on hierarchical b-spline grids

W.G. Dettmer, C. Kadapa, D. Perić

PII: S0045-7825(16)30489-3
DOI: http://dx.doi.org/10.1016/j.cma.2016.08.027
Reference: CMA 11108

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

Received date: 1 June 2016
Revised date: 24 August 2016
Accepted date: 28 August 2016

Please cite this article as: W.G. Dettmer, C. Kadapa, D. Perić, A stabilised immersed boundary method on hierarchical
b-spline grids, *Comput. Methods Appl. Mech. Engrg.* (2016),
http://dx.doi.org/10.1016/j.cma.2016.08.027

A stabilised immersed boundary method on hierarchical b-spline grids

W. G. Dettmer, C. Kadapa, D. Perić

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

Abstract

In this work, an immersed boundary finite element method is proposed which is based on a hierarchically refined cartesian b-spline grid and employs the non-symmetric and penalty-free version of Nitsche's method to enforce the boundary conditions. The strategy allows for h- and p-refinement and employs a so-called ghost penalty term to stabilise the cut cells. An effective procedure based on hierarchical subdivision and sub-cell merging, which avoids excessive numbers of quadrature points, is used for the integration of the cut cells. A basic Laplace problem is used to demonstrate the effectiveness of the cut cell stabilisation and of the penalty-free Nitsche method as well as their impact on accuracy. The methodology is also applied to the incompressible Navier-Stokes equations, where the SUPG/PSPG stabilisation is employed. Simulations of the lid-driven cavity flow and the flow around a cylinder at low Reynolds number show the good performance of the methodology. Excessive ill-conditioning of the system matrix is robustly avoided without jeopardising the accuracy at the immersed boundaries or in the field.

Keywords:
immersed boundary method, hierarchical b-splines, Nitsche's method, ghost penalty, Poisson equation, Navier-Stokes equations

1. Introduction

For more than a decade, the focus of many researchers has been on the development of finite element methods which allow for the modelling of problems in fluid or solid mechanics, featuring complex geometries, without requiring extensive resources for mesh generation.

One strategy is based on using the same basis functions for design and for analysis, thereby eliminating the need for mesh generation. The concept of Isogeometric Analysis, first published in [1], is based on the employment of NURBS, which are commonly used in CAD models, as basis functions for finite element analysis. This approach has since been followed by a large community, resulting in a wealth of publications, see, for instance, [2, 3, 4, 5] and references therein. A related strategy, based on subdivision surfaces and initially developed for elastic shells, was proposed in [6].

The present work follows the alternative approach which has been derived from the field of immersed boundary methods. Here, a suitable model of the surface geometry is immersed in a background mesh, where the basis functions are defined. An appropriate strategy is used to project the boundary conditions onto the background mesh. Hence, the definition of basis functions fitted to the complex problem geometry is not required. Efficient algorithms can be used to achieve local refinement of the background mesh. The parts of the background mesh which lie outside of the physical domain are generally referred to as the 'fictitious' domain. Depending on the methodology, the basis functions in the fictitious domain may be active or inactive.

The present work is also motivated by developments in the field of computational fluid-structure interaction. Modelling strategies based on body fitted meshes and arbitrary Lagrangian-Eulerian formulations (see, for instance, [7, 8] and references therein) are limited not only by the complex geometry of real world problems, but also by topology changes of the fluid domain. However, the robust modelling of such topology changes is crucial for the simulation of many industrial processes. This includes most types of valves, mixers and extruders. For such problems, immersed boundary, or rather immersed interface, methods offer an attractive alternative, which has been investigated recently in, for instance, [9, 10, 11]. Earlier work in this area is presented in [12] and references therein. Partitioned strategies for the resolution of the strong coupling between the fluid and solid phases, described for instance in [13, 14] and references therein, are generally applicable in the context of body fitted as well as immersed models.

In general, an immersed boundary finite element method requires four key ingredients: (i) a background mesh with suitable basis functions, (ii) a strategy for the imposition of the boundary conditions, (iii) a stabilisation procedure for degrees of freedom with partial support (cut cell stabilisation) and (iv) a sufficiently accurate integration scheme for cut cells. Depending on the treatment of the boundary conditions, the cut cell stabilisation may

not be required (see, for instance, [11, 15]). The choices made in the present work are elaborated in the following paragraphs.

Background mesh: In the present work, a cartesian grid with b-spline basis functions is employed. It allows for efficient h- and p-adaptivity and has been used in the context of immersed boundary methods in, for instance, [9, 11, 15, 16, 17, 18, 19, 20]. The local refinement can be structured in an efficient hierarchical manner (see, for instance, [21, 22, 23] and references therein).

Boundary conditions: The imposition of the boundary conditions in the background mesh can be achieved by using Lagrange multipliers or penalty methods (see, for instance, [10, 11, 12]). More recent methodologies are based on Nitsche's method [9, 15, 17, 18, 20, 24, 25, 26, 27], resulting in a scheme that is consistent and does not require additional variables. Nitsche's method, as originally proposed in [28], allows for the weak imposition of Dirichlet boundary conditions by adding certain boundary terms, including a penalty term, to the weak form of the governing equations. In the original, symmetric form of Nitsche's method, the penalty term is required to ensure coercivity of the formulation and does not jeopardise consistency. A modified version, namely the non-symmetric and penalty-free Nitsche method, has been proven and demonstrated to perform well for the advection-diffusion equation [29] as well as for linear elasticity [30]. The absence of the penalty parameter is particularly attractive in the context of complex numerical systems, such as those resulting from immersed methods, where several numerical concepts interact with each other and the number of artificial parameters should be restricted to a minimum. In this work the non-symmetric and penalty-free Nitsche method is employed in the context of an immersed method and applied to the Poisson problem as well as to the incompressible Navier-Stokes equations.

Cut cell stabilisation: Many basis functions in the vicinity of the interface possess little support inside the physical domain. In the context of a finite element method this leads to ill conditioned system matrices. In fact, the matrix condition number often increases by many orders of magnitude. This is not generally a problem for direct solvers, however, it is detrimental for the convergence of iterative solvers or leads to excessive requirements on the pre-conditioner. Thus, it is advisable to employ an appropriate stabilisation technique. In [17, 22, 31], the degrees of freedom with partial support are eliminated by replacing them with appropriate extrapolations of the internal degrees of freedom. While this strategy is very effective, its robust

implementation is non-trivial. Strategies based on the local computation of the penalty parameter for Nitsche's method are presented in [15, 32, 33] and proven to provide a stabilising effect for cut cells. In [26, 27] a so-called ghost penalty term is proposed which enforces a suitable amount of smoothness of the solution across all edges or faces in the vicinity of the immersed boundary. The present work employs a basic extension of the ghost penalty term to higher order basis functions.

Integration of cut cells: The spatial integration is typically based on auxiliary subdivisions of the cut cells, which can be integrated based on standard Gauss quadrature (see, for instance, [10, 18, 21, 27]). The present work employs a hierarchical strategy and, beyond a user controlled depth, merges the sub-cells in the vicinity of the interface. It is demonstrated in Section 3.3 that this procedure renders accurate results without requiring excessive numbers of quadrature points.

In the present work, the proposed methodology is applied to the Poisson equation and to the incompressible Navier-Stokes equations. Since equal-order interpolations are used for the velocity and the pressure fields, the latter requires the employment of a suitable stabilisation technique. The strategy adopted in this work is based on standard SUPG/PSPG stabilisation (see, for instance, [34, 35, 36] and references therein). A selected Laplace problem, the flow in a lid-driven cavity and the flow around a circular cylinder are used to study the performance of the proposed methodology.

As described above, the present work is related to [15, 17, 18, 20], which also propose immersed methodologies based on Nitsche's method and b-spline grids. However, it differs from the above approaches as it employs the parameter-free version of Nitsche's method and the ghost penalty strategy for cut cell stabilisation. Further differences consist in the adopted cut cell integration scheme and in the employment of the SUPG/PSPG finite element formulation for the Navier-Stokes equations. The present work is also related to [26, 27] where the ghost penalty term is proposed, but considered only in the context of linear finite element discretisation and linear Poisson and Stoke's problems. Finally, the non-symmetric and penalty-free Nitsche method presented in [29, 30] is, in the present work, applied in the context of an immersed method.

Thus, the key contributions of the present work may be summarised as follows:

- ghost penalty based cut cell stabilisation applied to higher order b-

spline basis functions,

- non-symmetric and penalty-free Nitsche method applied in the context of an immersed method,
- non-symmetric and penalty-free Nitsche method applied to the incompressible Navier-Stokes equations,
- integration of cut cells with sub-cell merging,
- comprehensive study of accuracy and matrix condition numbers.

The remainder of this article is structured as follows: In Section 2, the b-spline grid, the non-symmetric and penalty-free Nitsche method, the cut cell stabilisation and the integration scheme for cut cells are described. In Sections 3 and 4, the proposed variational formulations and some numerical examples are presented for, respectively, the Poisson and the incompressible Navier-Stokes equations. Conclusions are drawn in Section 5.

2. Computational ingredients

2.1. Hierarchically Refined B-Spline Grids

This section provides a brief description of hierarchically refined b-spline grids. More information is presented in, for instance, [11, 17, 21, 22, 37, 38] and references therein.

b-splines are piecewise continuous polynomial functions. Consider the knot vector $\{\xi_0, \xi_1, \ldots, \xi_{n+a+1}\}$, where a is the degree of the polynomials and n is the number of basis functions defined in the interval $[\xi_0, \xi_{n+a+1}]$. The b-spline basis functions $N_{i,a}$ can then be defined recursively by

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

and

$$N_{i,a}(\xi) = \frac{\xi - \xi_i}{\xi_{i+a} - \xi_i} N_{i,a-1}(\xi) + \frac{\xi_{i+a+1} - \xi}{\xi_{i+a+1} - \xi_{i+1}} N_{i+1,a-1}(\xi) \quad \text{for } a > 0\,. \tag{2}$$

The basis function $N_{i,a}$ spans from knot ξ_i to knot ξ_{i+a+1} and is $\mathcal{C}^{a-1}$ continuous in $[\xi_i, \xi_{i+a+1}]$. Since this work is restricted to uniform b-splines the knots

for each basis function are uniformly spaced, *i. e.* $\xi_{i+1} - \xi_i = \Delta\xi = const.$ Linear, quadratic and cubic uniform b-spline basis functions are displayed in Figure 1.

In the presence of local refinement, multiple layers of knot vectors with different spacing are employed. The two-scale relation, on which the refinement is based and which is graphically illustrated in Figure 1, may be expressed as

$$N_a(\xi) \;=\; \sum_{i=0}^{a+1} \alpha_i \, N_a(2\,\xi - i) \;, \tag{3}$$

where α_i are functions of binomial coefficients, given as,

$$\alpha_i \;=\; \frac{1}{2^a} \binom{a+1}{i} \;. \tag{4}$$

Thus, any b-spline function of order a can be represented as a linear combination of $a + 2$ b-splines of the same order which are defined on a knot vector with knot spacing $\Delta\xi/2$. Based on this principle, local refinement of the spatial resolution can be achieved by replacing a low level basis function by the appropriate set of basis functions on the next level. This approach is followed in the present work and, for instance, in [23, 39]. The alternative methodology described in [21] allows for more targeted refinement. It is based on the refinement of selected elements and generally leads to the introduction of fewer basis functions on the next higher level. The lower level basis functions are removed once the refinement is sufficiently complete. An efficient computer implementation of hierarchical b-spline grids is typically based on tree data structures and involves complex algorithms.

In higher dimensions, b-spline basis functions are constructed from tensor products. Thus, for two spatial dimensions, the basis functions are

$$N_{ij,a}(\xi, \eta) \;=\; N_{i,a}(\xi) \; N_{j,a}(\eta) \;. \tag{5}$$

The support of linear, quadratic and cubic uniform b-spline functions in two dimensions is illustrated in Figure 2. The evaluation of the basis functions and their derivatives can be implemented in a relatively straightforward manner, especially in the case of the cartesian grids considered in this work. In literature and in the remainder of this article, the knot spans which, in Figure 2, correspond to the square areas between the grid lines are referred to either as 'elements' or as 'cells'. Figure 3 shows a physical domain Ω with boundary Γ embedded in a two dimensional cartesian uniform b-spline grid.

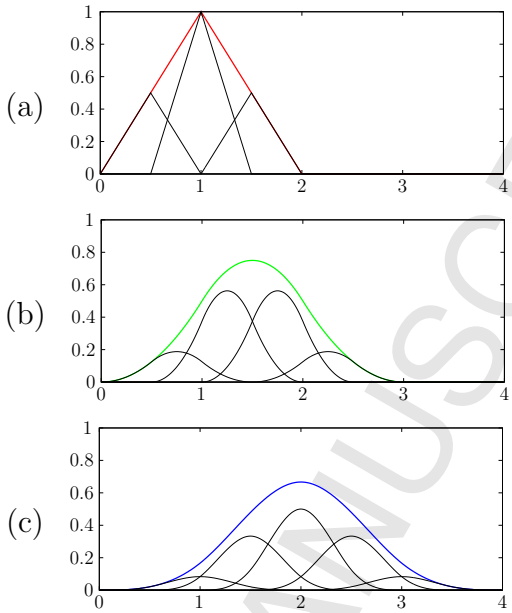

Figure 1: B-splines; two-scale relation of linear (a), quadratic (b) and cubic (c) basis functions.

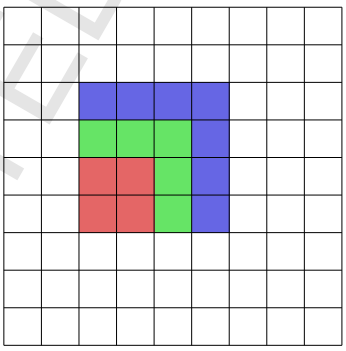

Figure 2: B-splines; support of linear (red), quadratic (green) and cubic (blue) basis functions in two dimensions.

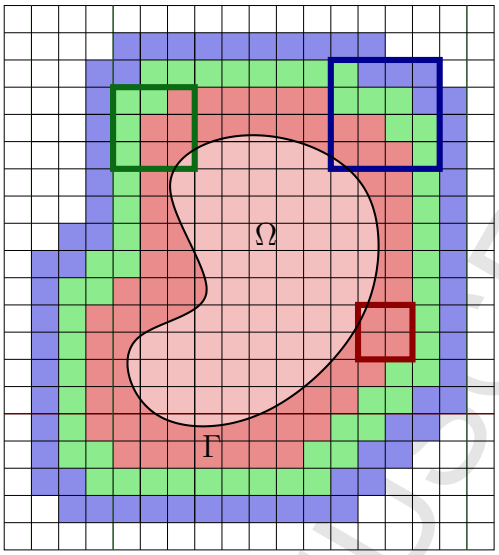

Figure 3: B-splines; domain Ω with boundary Γ immersed in a cartesian uniform b-spline grid; the span of all linear, quadratic and cubic basis functions intersecting the domain is marked, respectively, in red, green and blue; the span of a linear, quadratic and cubic basis function with little support inside the domain is outlined.

2.2. Non-symmetric and Penalty-Free Nitsche's Method

In the context of the variational formulation of a differential boundary value problem, Nitsche's method, which was first published in [28], can be used to weakly impose Dirichlet boundary conditions. The strategy is closely related to the penalty method and to Lagrange multipliers. However, in contrast to the former, Nitsche's method is consistent and, in contrast to the latter, it does not introduce a new type of solution variables. In the following, in preparation for Sections 3.1 and 4.1, Nitsche's method is outlined on the basis of the Poisson problem.

Consider the boundary value problem

$$\Delta u \; + \; f \; = \; 0 \quad \text{in } \Omega \tag{6}$$

$$u \; = \; g \quad \text{on } \Gamma \, , \tag{7}$$

where Γ denotes the boundary of Ω. A finite element formulation based on Nitsche's method leads to the following problem: Find $u_h \in U_h$ such that,

for all $v_h \in U_h$,

$$\int_\Omega \nabla v_h \cdot \nabla u_h \, \mathrm{d}\Omega + \int_\Gamma \nabla v_h \cdot \boldsymbol{n} \, (u_h - g) \, \mathrm{d}\Gamma + \int_\Gamma \nabla u_h \cdot \boldsymbol{n} \, v_h \, \mathrm{d}\Gamma + \frac{\gamma}{h} \int_\Gamma v_h \, (u_h - g) \, \mathrm{d}\Gamma = \int_\Omega v_h \, f \, \mathrm{d}\Omega \,, \tag{8}$$

where U_h denotes the appropriate space of polynomials on the discretised domain. The characteristic element size is denoted by h, while γ is a dimensionless penalty parameter. The formulation is consistent independently of the choice of γ.

It is proven in [29] and in [30] for, respectively, the advection-diffusion problem and for compressible and incompressible elasticity, that the non-symmetric and penalty-free version of Nitsche's method is also consistent and stable. Thus, the problem given by Equation (8) can be reformulated as follows: Find $u_h \in U_h$ such that, for all $v_h \in U_h$,

$$\int_\Omega \nabla v_h \cdot \nabla u_h \, \mathrm{d}\Omega + \int_\Gamma \nabla v_h \cdot \boldsymbol{n} \, (u_h - g) \, \mathrm{d}\Gamma - \int_\Gamma \nabla u_h \cdot \boldsymbol{n} \, v_h \, \mathrm{d}\Gamma = \int_\Omega v_h \, f \, \mathrm{d}\Omega \,. \tag{9}$$

In Sections 3.1 and 4.1, this version of Nitsche's method is used in an immersed boundary finite element method for, respectively, the Laplace equation and the incompressible Navier-Stokes equations. The Laplace problem is covered by the analysis presented in [29], while the corresponding formulation of the incompressible Navier-Stokes equations is motivated by the findings in [29, 30] but, to the knowledge of the authors, has not been presented elsewhere. The recent work in [20] also employs the non-symmetric and penalty-free version of Nitsche's method in the context of an immersed strategy with application to the Laplace equation and to linear elasticity. Early work on the non-symmetric version of Nitsche's method is presented in [40].

2.3. Cut Cell Stabilisation

An intuitive and effective strategy for cut cell stabilisation is based on substituting the degrees of freedom which possess insufficient support with an extrapolation of the solution variables from the inside of the domain towards the boundary (see [17, 22]). For complex boundary geometries, the computer implementation of this methodology is, however, not straightforward.

The present work follows the approach described in [27], which uses ghost penalty terms. The fundamental idea of this technique is to weakly enforce an appropriate amount of smoothness of the solution across the edges between the cut cells and across the edges between the cut cells and the interior cells (see Figure 4). For linear finite element basis functions, the second derivatives vanish and the penalty term is applied to the jump of the first derivative normal to the inter-element edge. Thus, in the context of a linear finite element discretisation for the two dimensional Poisson problem of Section 2.2, the ghost penalty term associated with edge k reads

$$z_k(u_h, v_h) \;=\; \gamma\, h \int_{l_k} [\![\nabla u_h \cdot \boldsymbol{n}]\!]_k \, [\![\nabla v_h \cdot \boldsymbol{n}]\!]_k \, \mathrm{d}s \;, \tag{10}$$

where γ, h and l_k denote, respectively, a non-dimensional penalty parameter, the characteristic element size and the length of edge k. The operator $[\![\bullet]\!]_k$ evaluates the jump of $(\bullet)$ across the inter-element edge k. In the three dimensional case, corresponding ghost penalty terms are constructed for inter-element faces.

The present work uses a cartesian background mesh with b-spline basis functions of arbitrary polynomial degree a. In this case, the ghost penalty term has to be applied to the highest non-zero derivative, since all lower derivatives of the solution variable are continuous across the inter-element edges. Appropriate powers of the characteristic element size h are used to maintain dimensional consistency. Thus, for an arbitrary polynomial degree a, Equation (10) is rewritten as

$$z_k(u_h, v_h) \;=\; \gamma\, h^{2a-1} \int_{l_k} [\![\partial^a u_h / \partial x_j^a]\!]_k \, [\![\partial^a v_h / \partial x_j^a]\!]_k \, \mathrm{d}x_i \;, \tag{11}$$

where, in the two dimensional setting, x_i is the spatial coordinate along edge k and x_j is the coordinate perpendicular to x_i. Given the purpose of the ghost penalty term, it does not seem necessary to accurately evaluate the integral in Equation (11). Thus, in the present work, the ghost penalty terms are evaluated as

$$z_k(u_h, v_h) \;=\; \gamma\, l_k^{2a} \, [\![\partial^a u_h / \partial x_j^a]\!]_{\bar{k}} \, [\![\partial^a v_h / \partial x_j^a]\!]_{\bar{k}} \;\;, \tag{12}$$

where the operator $[\![\bullet]\!]_{\bar{k}}$ evaluates the jump of $(\bullet)$ across the inter-element edge k at the centre of the edge. The expression in Equation (12) also utilises the observation that, on cartesian b-spline meshes, the length l_k of edge k and

the characteristic element size h are generally identical. In case of adaptively refined meshes, edge k may be separating a large and a small element. In this case, l_k is set equal to the edge length of the smaller element. In the context of the Navier-Stokes equations, the factors in front of the jump terms are modified as described in Section 4.1.

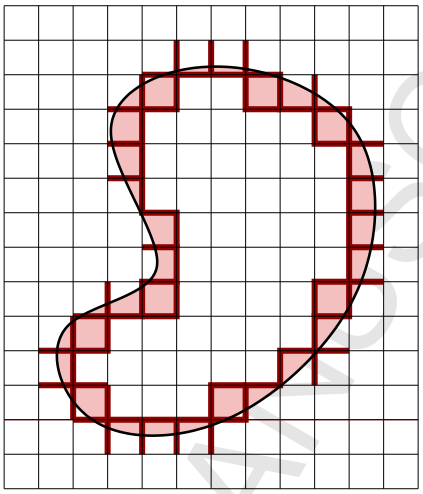

Figure 4: Cut cell stabilisation; cut cells marked light-red and edges where ghost penalty terms are applied marked dark-red.

2.4. Hierarchical integration of Cut Cells

The integration of the cells which are fully contained inside the problem domain is performed in a standard manner based on Gauß quadrature. In the two dimensional setting, 2×2, 3×3 and 4×4 Gauß points are employed for, respectively, $a = 1$, $a = 2$ and $a = 3$.

A commonly adopted strategy for the integration of cut cells is based on the triangulation of the cut cell and the subsequent employment of standard Gauß quadrature. In the two dimensional setting this is a standard procedure whereas, in three dimensions, it requires the consideration of numerous special cases and is rather cumbersome (see, for instance, [27]). Additional effort is needed if the immersed boundary is represented by curved lines or surfaces. An alternative strategy is based on hierarchically subdividing the cut cells into smaller areas and using standard Gauß quadrature for the subcells (see, for instance, [10, 18, 21]). If the integral must be computed with a high degree of accuracy, this approach typically leads to excessive numbers

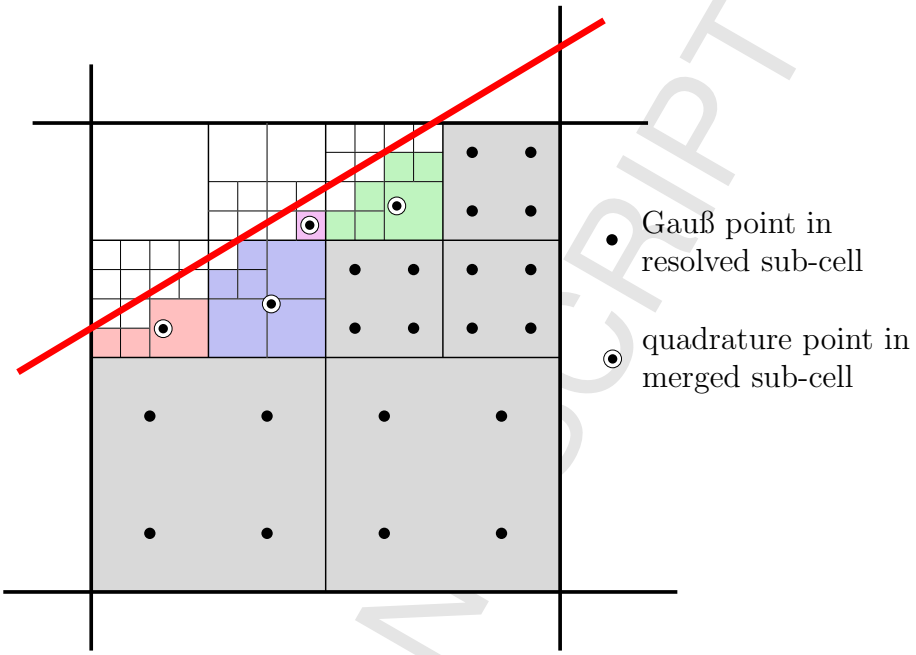

Figure 5: Integration of cut cells; merging of Gauß points near the interface for $d_{\text{resolve}} = 2$ and $d_{\text{merge}} = 4$.

of quadrature points. Its significant advantage, however, lies in the fact that it only requires an 'inside-outside' algorithm to determine if a cell lies inside or outside of the physical domain or needs to be further divided. Such algorithms are standard in the field of computer graphics and efficient versions are available.

In this work, a modification of the latter approach is proposed: The cut cells and the cut sub-cells are hierarchically subdivided whereby, beyond a specified depth d_{resolve}, the integration points are merged together, *i. e.* single quadrature points located at the centroid of the higher level sub-cells are used. The total depth of the subdivision hierarchy is denoted by d_{merge}. This is illustrated in Figure 5 for $d_{\text{resolve}} = 2$ and $d_{\text{merge}} = 4$. Hence, this strategy leads to sub-optimal positioning of the integration points near the interface, but it allows for the accurate representation of the integration area without introducing large numbers of quadrature points. The performance of this approach is assessed in Section 3.3.

3. Poisson equation

3.1. Variational formulation

The Poisson problem is described by Equations (6) and (7) and repeated here for convenience,

$$\Delta u \; + \; f \; = \; 0 \quad \text{in } \Omega \tag{13}$$

$$u \; = \; g \quad \text{on } \Gamma \, . \tag{14}$$

Using the immersed boundary methodology based a cartesian background mesh with hierarchical b-splines, the non-symmetric and penalty-free Nitsche method and ghost penalty terms for cut cell stabilisation, all of which have been described above, the variational problem to be solved reads: Find $u_h \in U_h$ such that, for all $v_h \in U_h$,

$$\begin{aligned} \int_\Omega \nabla v_h \cdot \nabla u_h \, \mathrm{d}\Omega \; + \; \int_\Gamma \nabla v_h \cdot \boldsymbol{n} \, (u_h \, - \, g) \, \mathrm{d}\Gamma \; - \; \int_\Gamma \nabla u_h \cdot \boldsymbol{n} \, v_h \, \mathrm{d}\Gamma \\ + \; \sum_{k=1}^{K} \gamma \; l_k^{2a} \; [\![\partial^a u_h / \partial x_j^a]\!]_{\bar{k}} \; [\![\partial^a v_h / \partial x_j^a]\!]_{\bar{k}} \; = \; \int_\Omega v_h \; f \, \mathrm{d}\Omega \, , \end{aligned} \tag{15}$$

where the space U_h is based on the cartesian b-spline mesh. The number of edges where ghost penalty terms need to be applied is denoted by K.

3.2. Laplace equation on a square domain

The Laplace equation, *i. e.* Equation (13) with $f = 0$, is solved on a two dimensional square domain with $0 < x < 1$ and $0 < y < 1$. Dirichlet boundary conditions are applied such that, for $y = 0$, $\phi = \sin(x/\pi)$, and $\phi = 0$ on all other edges. The exact solution can be expressed as

$$\phi = \big(\cosh(\pi \, y) - \coth(\pi) \sinh(\pi \, y) \big) \; \sin(\pi \, x) \, .$$

This problem has also been used for similar demonstrational purposes in [18, 33]. All system matrix condition numbers presented in this section and in the remainder of the article have been evaluated exactly with MatLab (MathWorks). For clarity, this Laplace problem is referred to as 'aligned', whereas the one described in Section 3.3 is 'tilted'.

3.2.1. Effect of cut cells on system matrix condition number

A grid of 20×20 elements is used to create background meshes with linear, quadratic and cubic b-spline basis functions. Meshes of varying size are generated such that the boundary of the physical domain is always immersed in the layer of elements adjacent to the mesh boundary. In this way, the cut cell ratio, obtained by dividing the resolved part of a cut element by its total size, is varied between 0.01 and one. Figure 6 shows the background mesh and the immersed boundary for a cut ratio of 0.30. For each computation the condition number of the system matrix is evaluated.

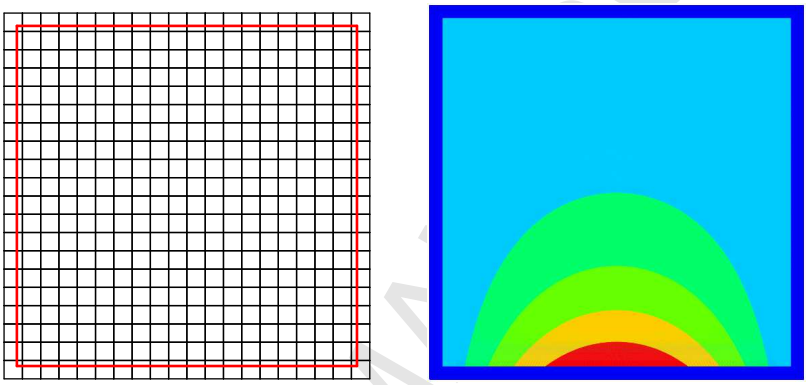

Figure 6: Laplace problem (aligned); mesh plot and contour plot of numerical solution for 20 elements per edge, $a = 2$, cut ratio 0.3, $\gamma = 0.05$.

Figure 6 shows the contour plot of a typical solution. The dependency of the matrix condition number on the cut ratio is displayed in Figure 7. For $a = 1, 2, 3$, the diagram shows the condition numbers obtained with and without cut cell stabilisation ($\gamma = 0.05$ and $\gamma = 0$, respectively).

The following observations are made:

Without stabilisation. Without stabilisation, the condition numbers rise as the cut cell ratio decreases. For cut ratios of less than 0.2 this rise accelerates significantly and, as the proportion of the cut element inside the computational domain tends towards zero, the condition numbers are 10^6 ($a = 1$) to 10^{22} ($a = 3$) times higher than for cut ratios near one. This is due to the well-known destabilising effect of small cut cells and arises from those degrees of freedom which possess too little support inside the computational domain. This observation confirms the need for an efficient stabilisation technique.

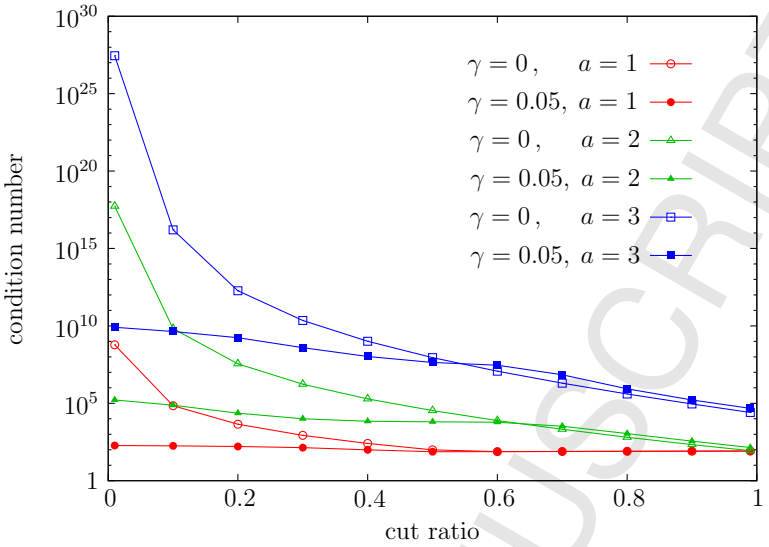

Figure 7: Laplace problem (aligned); condition number of system matrix for $\gamma = 0$ and $\gamma = 0.05$ displayed against cut ratio of the element layer adjacent to the mesh boundary.

With stabilisation. For all orders a considered, the application of the ghost penalty based cut cell stabilisation is effective. Down to a cut cell ratio in the region of 0.5 the condition numbers are similar to those obtained without stabilisation. For smaller cut ratios the condition numbers are reduced by several orders of magnitude and their accelerated increase observed without stabilisation is suppressed.

Effect of b-spline basis function order. The condition numbers for higher orders of b-spline basis functions are several orders of magnitudes higher than those for lower orders. This is a well-known property of b-spline based discretisations (see, for instance, [41] and references therein).

3.2.2. Effect of cut cells on solution accuracy

Accuracy for body fitted mesh. Prior to the assessment of the accuracy of the immersed method, the convergence rates obtained for body fitted meshes are evaluated. For this purpose, background meshes with $N = 5, 10, 20, ..., 320$ elements along the domain edges are generated. Analogously to the immersed case, the Dirichlet conditions along the mesh boundary are applied by means

of the non-symmetric penalty-free version of Nitsche's method. The L_2 norm of the error is computed and displayed in Figure 8. The observed rates of convergence are optimal for all orders a. In [20, 29], the authors employ mathematical analysis to show that the theoretical rate of convergence in the L_2 norm of the non-symmetric penalty-free Nitsche method is sub-optimal. However, similar to the results presented in this work, all numerical examples in [20, 29] also exhibit optimal convergence rates in the L_2 norm.

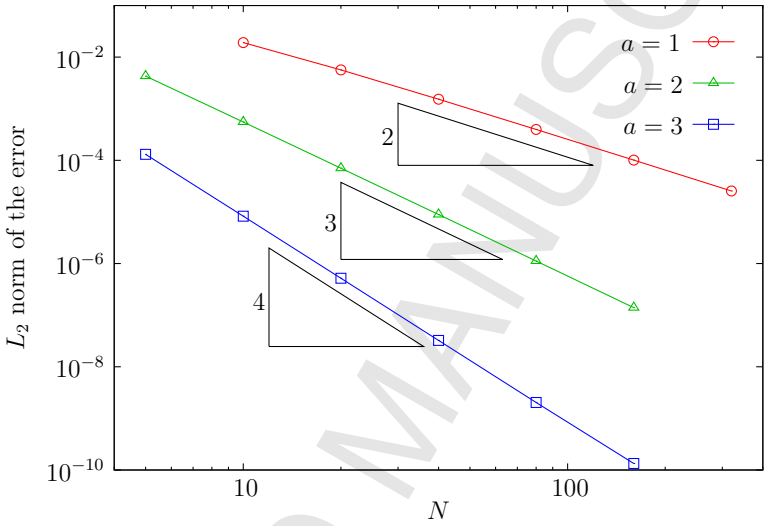

Figure 8: Laplace problem (aligned); convergence of error for body fitted mesh.

Accuracy for immersed method. Next, the error of the solution is evaluated for a number of immersed cases. The cut ratios considered are 1/64, 1/2 and 63/64. For these values, the hierarchical integration strategy described in Section 2.4 leads to exact integration of the computational domain. Thus, inaccurate integration can be excluded as a cause for poor rates of convergence. The ghost penalty parameter is set to $\gamma = 0.05$. Figures 9 to 11 show the rates of convergence for $a = 1, 2, 3$, respectively. In all cases, the slopes of the diagrams are identical to those obtained for the body fitted mesh, but the error magnitude has changed. For $a = 3$ and $a = 2$, the error has increased by approximately one order of magnitude whereas, for $a = 1$, the error has decreased rather than increased. This may be attributed to the fact that the degrees of freedom with the smallest support inside the computational

domain have less impact on the solution for higher than for lower orders a. For large a, they are more strongly influenced by the stabilisation than by the governing equation.

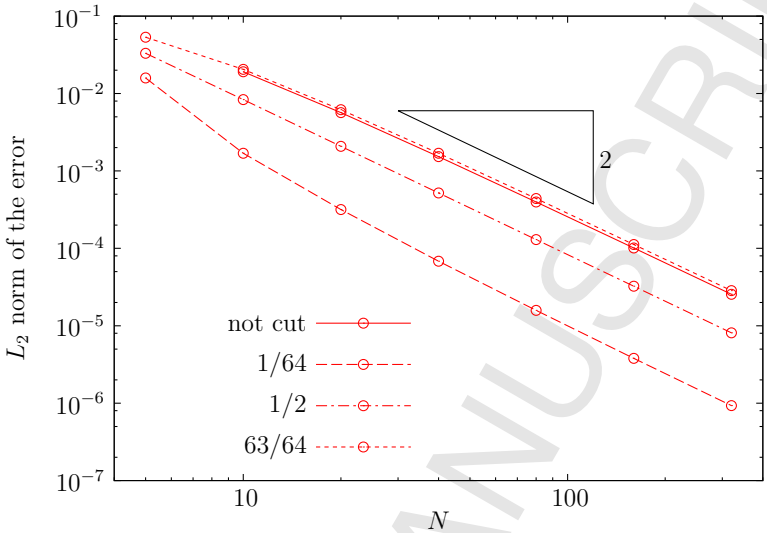

Figure 9: Laplace problem (aligned); convergence of error for different cut ratios; $a = 1$; 20 elements along the edges.

3.2.3. Choice of ghost penalty parameter γ

In order to study the effect of the ghost penalty parameter γ on the solution accuracy and on the matrix condition number, the computation is performed with the cut ratio 1/64 and 20×20 elements for different values of γ. For each computation the solution error and the system matrix condition number are evaluated.

Solution accuracy. Figure 12 shows the dependency of the error on γ. It is observed that the error is constant for small values of γ. For quadratic and cubic b-spline basis functions it rises to a higher level at approximately $\gamma = 10^{-4}$ and $\gamma = 10^{-8}$, respectively. Once γ exceeds a critical value in the region of 0.1 the error rises abruptly and steeply for all orders a. For $\gamma > 0.1$, the poor quality of the solution near the boundaries becomes visible in the contour plots (not shown here).

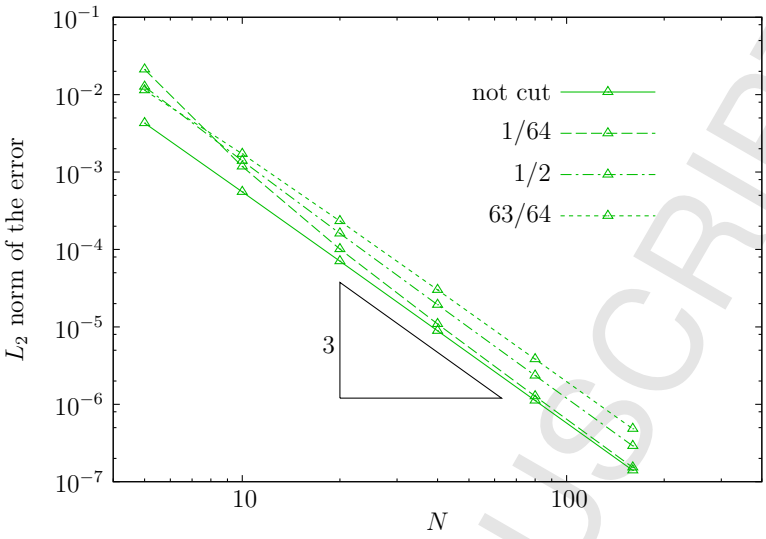

Figure 10: Laplace problem (aligned); convergence of error for different cut ratios; $a = 2$; 20 elements along the edges.

Matrix condition number. The matrix condition numbers for $\gamma = 0$ are 1.04e+8, 4.11e+16 and 1.76e+25 for $a = 1, 2, 3$, respectively. For ghost penalty parameters $\gamma \geq 1.\text{e} - 14$ the matrix condition numbers are shown in the diagram in Figure 13. For all orders a considered, the minimum of the condition number is clearly visible and located in the region $0.001 < \gamma < 0.5$. It is noted that the optimal value of γ is smaller for higher order basis functions than for lower. The ratios between the condition numbers obtained without any stabilisation and those obtained with optimal γ are approximately 10^6, 10^{11} and 10^{15} for $a = 1, 2, 3$, respectively.

The comparison of Figures 12 and 13 shows that the values of γ which render the smallest matrix condition numbers lie close to the critical values beyond which the stabilisation jeopardises the accuracy of the solution. However, the condition numbers can be reduced by several orders of magnitude if γ is chosen within a wide interval of smaller values.

3.2.4. Non-symmetric and penalty-free Nitsche's method

All computations described in Sections 3.2.1 to 3.2.3 employ the non-symmetric and penalty-free version of Nitsche's method. In all cases, the boundary conditions are accurately satisfied and the behaviour of the solution near the boundary is excellent.

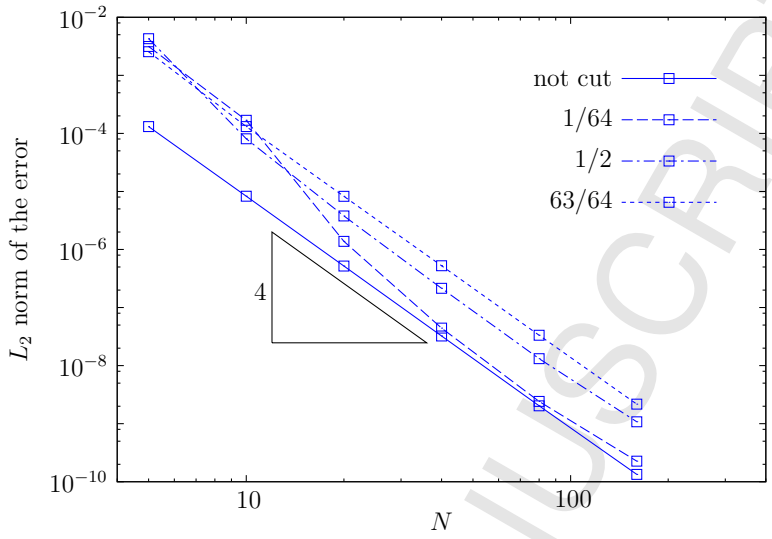

Figure 11: Laplace problem (aligned); convergence of error for different cut ratios; $a = 3$; 20 elements along the edges.

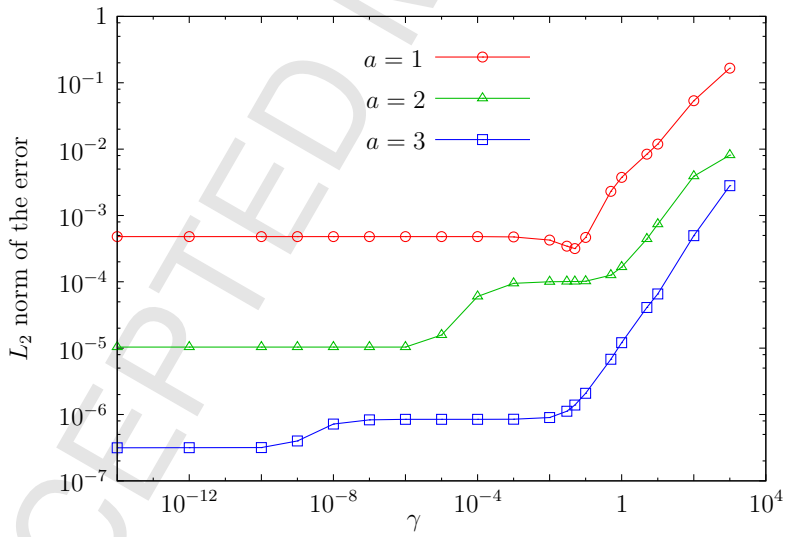

Figure 12: Laplace problem (aligned); error against γ for 20 elements along the edges.

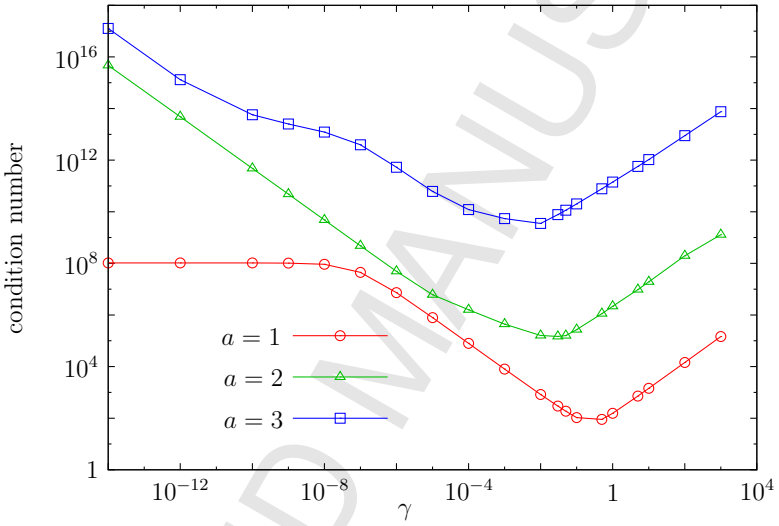

Figure 13: Laplace problem (aligned); condition number of system matrix against γ for 20 elements along the edges.

3.3. Laplace equation on a tilted domain

The problem investigated here is the same as in Section 3.2. However, the square computational domain is now rotated by 30 degrees and immersed in a larger uniform background mesh as shown in Figure 14. Using background meshes with 5, 10, 20, ..., 320 elements along each edge, the solution error is evaluated based on three different sets of integration parameters which are in the following referred to as *fine*, *coarse* and *merged*, see Table 1. The *fine* integration scheme requires very large numbers of quadrature points. For the *coarse* scheme, the number of quadrature points is significantly smaller. The *merged* strategy fully exploits the sub-cell merging as described in Section 2.4 and renders only marginally more quadrature points than the *coarse* scheme, while the integrated area is still represented accurately.

Solution Accuracy: Figure 15 shows the solution errors as obtained with *fine* integration. For $a = 1, 2, 3$, the graphs converge at the optimal rates. The performance of the *coarse* and *merged* schemes is shown in Figures 16 to 18. The contour plot of a typical solution is displayed in Figure 14. For linear basis functions and the *coarse* integration scheme, the rate of convergence is initially optimal, but quickly deteriorates as the mesh is refined. For $a = 2$ and $a = 3$, the optimal rates of convergence are lost at even coarser levels of discretisation. Remarkably, the *merged* integration procedure renders solutions of the same accuracy as the *fine* scheme. Thus, the merging of quadrature points as described in Section 2.4 avoids excessive numbers of quadrature points in cut elements without jeopardising the solution accuracy. It is also concluded that the *merged* and *coarse* schemes possess a sufficient number of quadrature points on the immersed boundary.

Matrix condition numbers: Using $a = 2$, the background mesh with 20

	fine	*coarse*	*merged*
n_{immerse}	$20\ N\ (p+1)$	$2\,N\ (p+1)$	$2\,N\ (p+1)$
d_{merge} / d_{resolve}	15 / 10	4 / 4	15 / 4

Table 1: Laplace problem (tilted); sets of integration parameters; n_{immerse} is the number of uniformly distributed quadrature points on each of the four immersed boundary edges; d_{resolve} is the depth of hierarchical quadrature refinement for cut elements which is resolved with standard Gauß integration; quadrature points of refinement levels d_{resolve}+1 to d_{merge} are merged; $N = 5, 10, 20, 40, ...$ is the number of elements along each edge of the mesh.

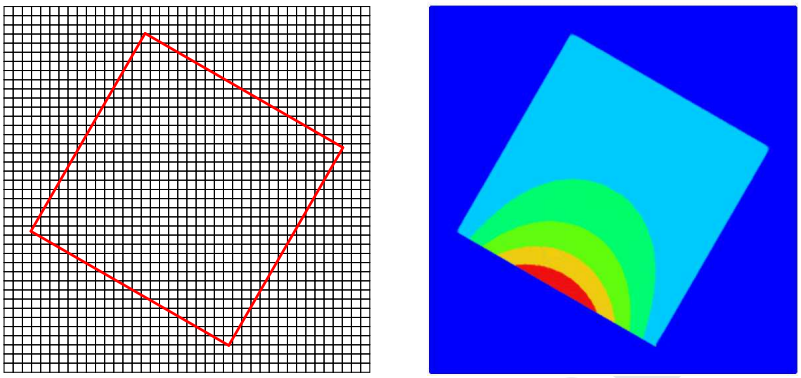

Figure 14: Laplace problem (tilted); mesh plot and contour plot of numerical solution for 40×40 elements and $a = 2$, $\gamma = 0.05$.

elements along each edge and the *merged* integration scheme, the computations are repeated for a range of ghost penalty parameters γ. In Figures 19 and 20, the solution error and the condition numbers of the system matrix, respectively, are displayed against γ. For $\gamma = 0.01$, the condition number is reduced by a factor of nearly 10^7 compared to $\gamma = 0$. For small values of γ, the solution error is affected little by the cut cell stabilisation, but it rises sharply if $\gamma > 0.01$. This behaviour is consistent with the observations made in Section 3.2.

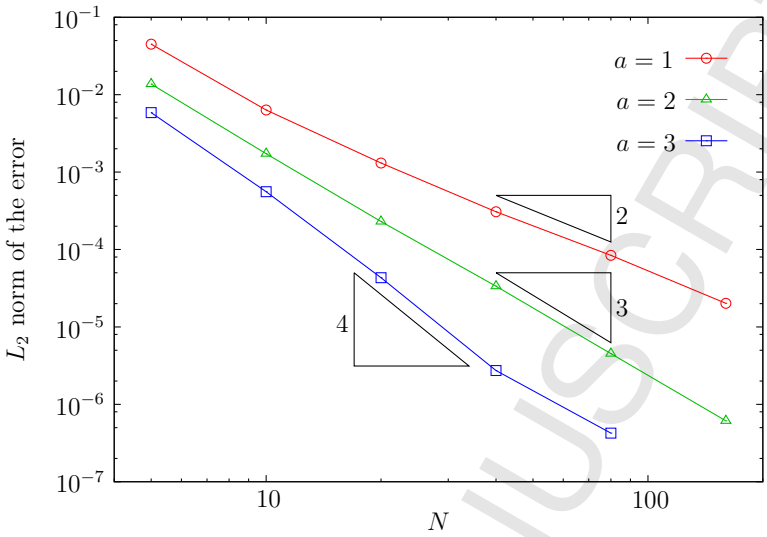

Figure 15: Laplace problem (tilted); convergence of error; accurate integration; $\gamma = 0.05$.

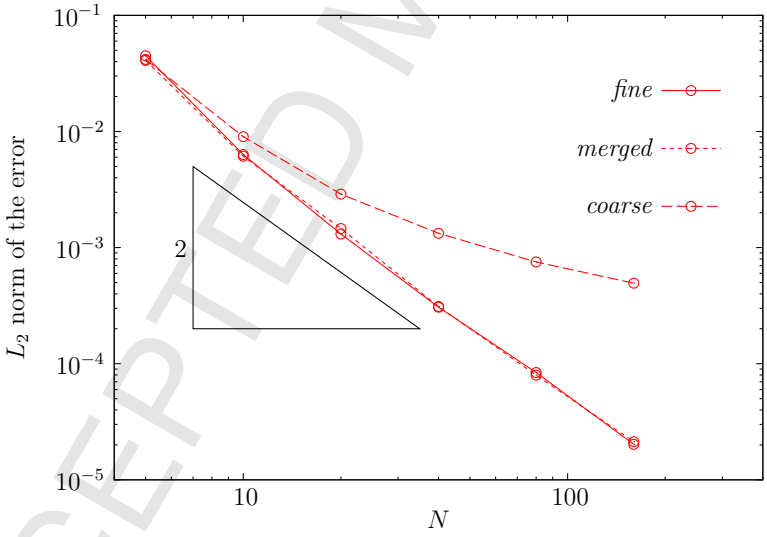

Figure 16: Laplace problem (tilted); convergence of error; different integration parameters; $a = 1$; $\gamma = 0.05$.

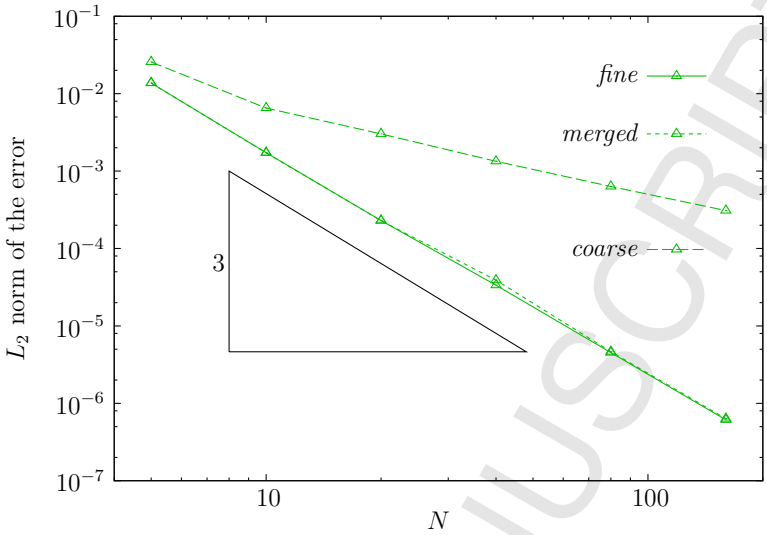

Figure 17: Laplace problem (tilted); convergence of error; different integration parameters; $a = 2$; $\gamma = 0.05$.

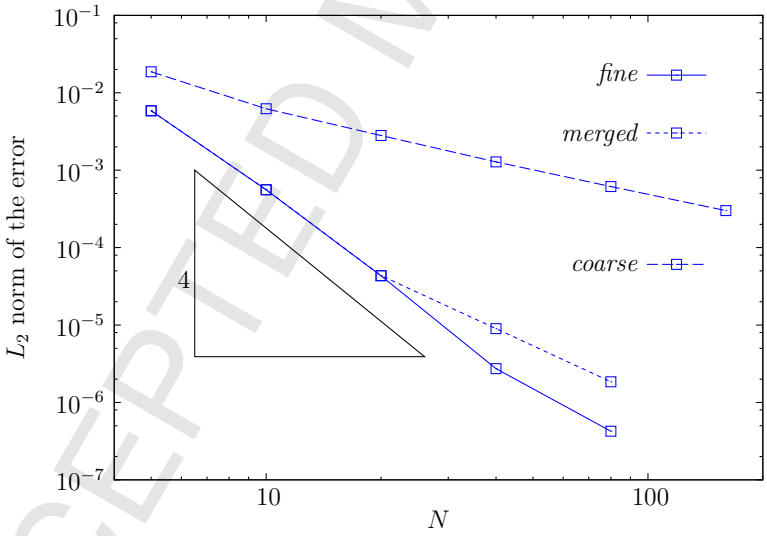

Figure 18: Laplace problem (tilted); convergence of error; different integration parameters; $a = 3$; $\gamma = 0.05$.

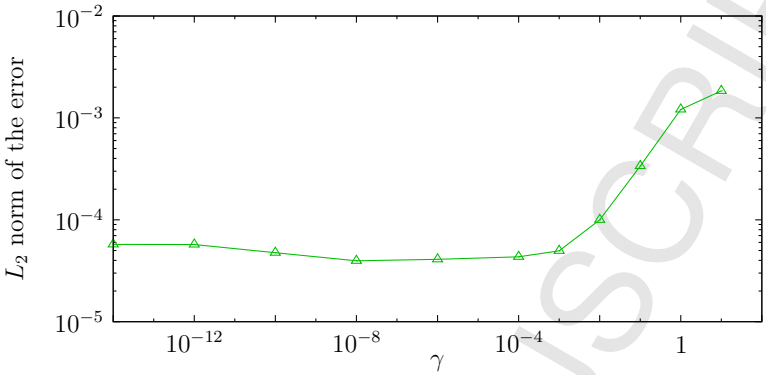

Figure 19: Laplace problem (tilted); solution error against γ; $a = 2$, 20 elements along the edges, *merged* integration scheme.

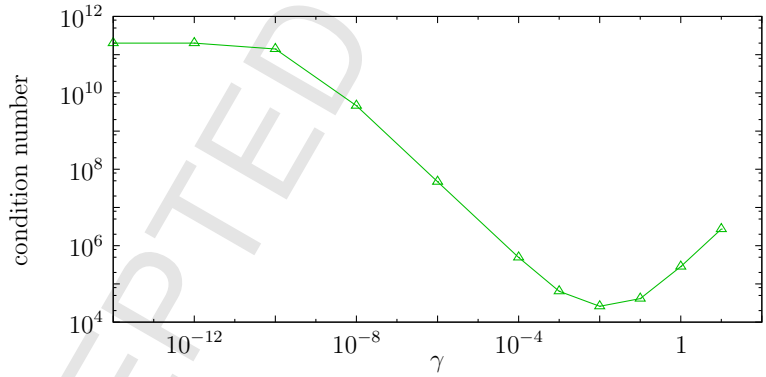

Figure 20: Laplace problem (tilted); condition number against γ; $a = 2$, 20 elements along the edges, *merged* integration scheme.

4. Incompressible Navier-Stokes equations

4.1. Variational formulation

The incompressible Navier-Stokes equations read

$$\rho\,\boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla\cdot\boldsymbol{\sigma} = \boldsymbol{f} \quad \text{in } \Omega \tag{16}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega \tag{17}$$

$$\boldsymbol{\sigma} = 2\,\mu\,\nabla^s\boldsymbol{u} - p\,\boldsymbol{I} \tag{18}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma_g \tag{19}$$

$$\boldsymbol{\sigma}\,\boldsymbol{n} = \boldsymbol{t} \quad \text{on } \Gamma_t \tag{20}$$

where the domain boundary $\Gamma = \Gamma_g \cup \Gamma_t$ consists of distinct Dirichlet and Neumann sections. The quantities $\boldsymbol{u}$ and p denote, respectively, the velocity vector and the pressure, while the parameters ρ and μ represent, respectively, the fluid density and the fluid viscosity. Equation (18) defines the stress tensor $\boldsymbol{\sigma}$. The operator $\nabla^s = \frac{1}{2}(\nabla + \nabla^T)$ denotes the symmetric gradient operator. The section Γ_g of the domain boundary includes all immersed boundaries, whereas Γ_t is restricted to parts of the boundary of the background mesh.

The variational formulation, proposed for the computational setting discussed above, reads: Find $\boldsymbol{u}_h \in U_h$ and $p_h \in P_h$ such that, for all $\boldsymbol{v}_h \in U_h$

and $q_h \in P_h$,

$$\begin{aligned}
&\int_\Omega \Big(\rho\, \boldsymbol{v}_h \cdot (\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h) \;+\; 2\,\mu\, \nabla \boldsymbol{v}_h : \nabla^s \boldsymbol{u}_h \\
&\qquad\qquad - (\nabla \cdot \boldsymbol{v}_h)\, p_h \;+\; q_h\, (\nabla \cdot \boldsymbol{u}_h) \Big)\, \mathrm{d}\Omega \;-\; \int_{\Gamma_t} \boldsymbol{v}_h \cdot \boldsymbol{t}\, \mathrm{d}\Gamma_t \\
&+ \sum_{e=1}^{N} \int_{\Omega^e} \Big(\tau_{\mathrm{SUPG}}\, \rho\, (\boldsymbol{u}_h \cdot \nabla \boldsymbol{v}_h) + \tau_{\mathrm{PSPG}}\, \nabla q_h \Big) \\
&\qquad\qquad \cdot \Big(\rho\, (\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h) - 2\,\mu\, \nabla \cdot \nabla^s \boldsymbol{u}_h + \nabla p_h \Big)\, \mathrm{d}\Omega^e \\
&\qquad + \int_{\Gamma_g} \big(2\,\mu\, \nabla^s \boldsymbol{v}_h - q_h\, \boldsymbol{I} \big)\, \boldsymbol{n} \cdot (\boldsymbol{u}_h - \boldsymbol{g})\, \mathrm{d}\Gamma_g \\
&\qquad\qquad - \int_{\Gamma_g} \big(2\,\mu\, \nabla^s \boldsymbol{u}_h - p_h\, \boldsymbol{I} \big)\, \boldsymbol{n} \cdot \boldsymbol{v}_h\, \mathrm{d}\Gamma_g \\
&+ \sum_{k=1}^{K} \gamma\, l_k^{2a} \Big(\mu\, [\![\partial^a \boldsymbol{u}_h / \partial x_j^a]\!]_{\bar{k}} \cdot [\![\partial^a \boldsymbol{v}_h / \partial x_j^a]\!]_{\bar{k}} \\
&\qquad\qquad + \frac{l_k^2}{\mu}\, [\![\partial^a p_h / \partial x_j^a]\!]_{\bar{k}}\, [\![\partial^a q_h / \partial x_j^a]\!]_{\bar{k}} \Big) \;=\; 0\,,
\end{aligned} \tag{21}$$

where U_h and P_h are appropriate approximation spaces, based on the cartesian background mesh with hierarchical b-splines. The first two integrals of Equation (21) represent the standard Galerkin terms. They are followed by the sum of the SUPG/PSPG stabilisation terms of all elements, which are discussed further below. The integrals over the Dirichlet boundary Γ_g arise from the non-symmetric and penalty-free version of Nitsche's method, while the last term in Equation (21) represents the ghost penalty based cut cell stabilisation.

Since equal orders of interpolation are used for the velocity and for the pressure, it is necessary to stabilise the pressure field and, for advection dominated problems, also the velocity field. In this work, a standard SUPG/PSPG technique is employed (see for instance [34]). Following [35] and references therein, the stabilisation parameters are defined as

$$\tau_{\mathrm{PSPG}} \;=\; \frac{h^2}{4\,\mu} \tag{22}$$

$$\tau_{\mathrm{SUPG}} \;=\; \left(\left(\frac{h^2}{4\,\mu} \right)^{-2} + \left(\frac{h}{2\,\rho\,\|\boldsymbol{u}^h\|} \right)^{-2} \right)^{-\frac{1}{2}} \;. \tag{23}$$

For the evaluation of the stabilisation parameters in cut cells, the characteristic element size h is evaluated, in the two dimensional setting, as

$$h \;=\; \sqrt{A'} \;, \tag{24}$$

where A' is the intersection of the cell area and the physical domain Ω.

4.2. Lid-Driven Cavity Flow

The lid-driven incompressible fluid flow in a square cavity is examined in this widely used benchmark problem. The length of the sides of the cavity is one and the no-slip boundary condition is applied at the left, right and bottom boundaries of the cavity. At the lid, the normal velocity is fixed at zero, while the prescribed tangential velocity varies according to

$$u(x) = 1 - \exp(\alpha\,(x-1)) - \exp(-\alpha\,x) \quad \text{for} \quad 0 \le x \le 1 \;.$$

The dimensionless parameter α controls the sharpness of the discontinuity of the boundary conditions in the corners. In this example, $\alpha = 500$ is used. The fluid density is set to 1. For the viscosity, the values of 0.01 and 0.001 are employed, corresponding to Reynolds numbers 100 and 1000, respectively. The cavity is immersed in a square background mesh with 20 or 40 large elements along each edge of the cavity and two or three levels of adaptive refinement in the top corners as shown in Figures 21 and 22. In order to assess the performance of the cut cell stabilisation, only 1/64 (1/32 for the denser mesh) of the area of the largest elements in the layer adjacent to the mesh boundary is located inside the cavity. A range of values for the ghost penalty parameter γ is considered. As the problem is defined by pure Dirichlet boundary conditions, a standard penalty term is used to enforce zero pressure at the bottom centre of the cavity. For $Re = 1000$, the lid velocity is raised from zero to 1 in five increments. In all cases, the Newton procedure renders asymptotically quadratic convergence of the residual.

Quality of the solutions: Figures 21 and 22 show contour plots of the pressure and the velocity magnitude for $a = 1, 2, 3$ and $Re = 100$ and $Re = 1000$, respectively. The ghost penalty parameter employed is $\gamma = 0.001$. The pressure field does not exhibit any spurious oscillations. For all orders a considered, the PSPG/SUPG stabilisation strategy is effective in the far field as

well as close to the immersed boundaries. The inaccuracies which can be observed for $a = 1$ near the top corners of the cavity are due to the coarseness of the mesh and disappear quickly with some additional refinement. The velocity profiles along the horizontal and vertical cuts through the centre of the cavity are shown in Figure 23. The results obtained for $a = 2$ and $a = 3$ coincide with the reference solution presented in [42] while, as expected, for $Re = 1000$ the linear basis functions require additional mesh refinement.

Matrix condition numbers: For $Re = 100$, the condition numbers of the system matrix associated with the converged solution at the end of the Newton procedure are evaluated. The dependency on the ghost penalty parameter γ is displayed in Figure 24. Values $\gamma > 1$ lead to solutions which show spurious behaviour near the lid. Taking into account the condition numbers associated with $\gamma = 0$ (given in the caption of Figure 24), it is evident that the cut cell stabilisation is effective and achieves reduction factors of more than 10^8, 10^{10} and 10^{12} for $a = 1$, 2, 3, respectively.

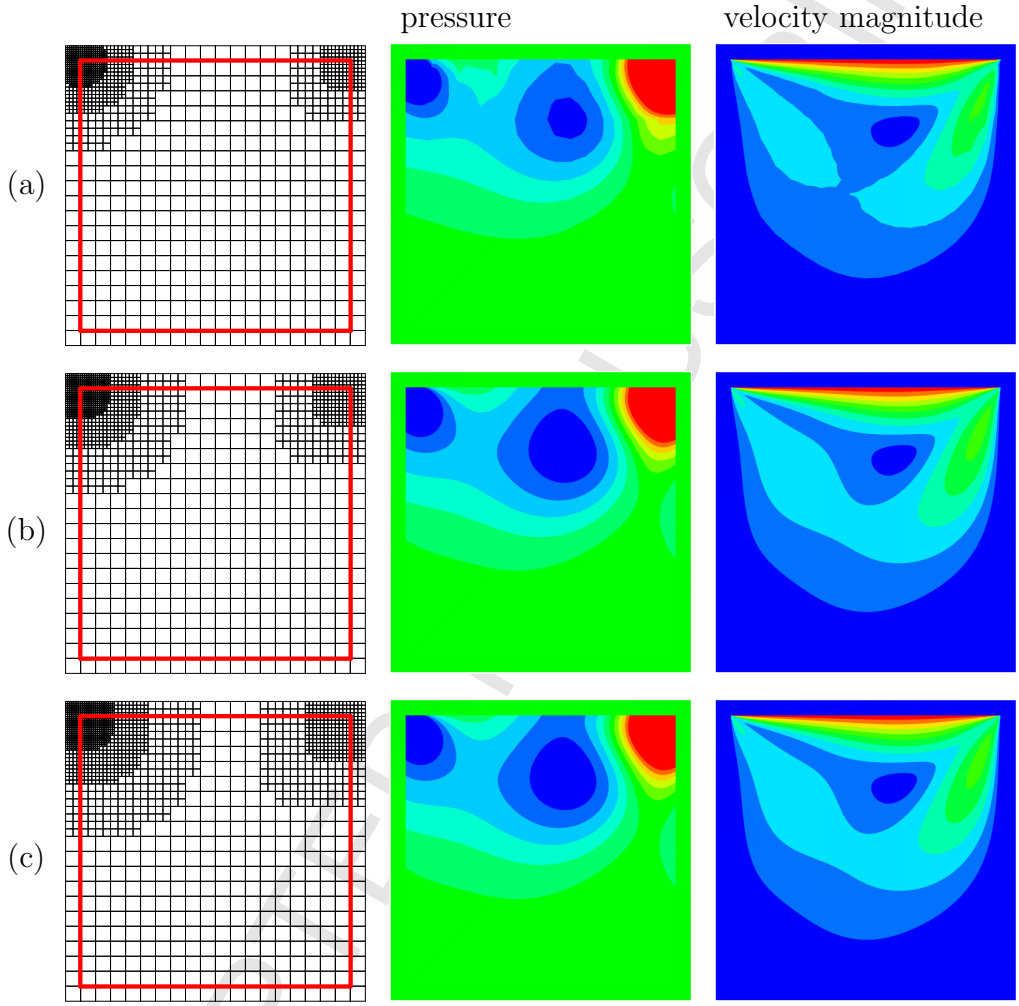

Figure 21: Lid driven cavity flow with $Re = 100$; mesh plots, pressure and velocity magnitude contour plots; (a) $a = 1$ with 2751 degrees of freedom, (b) $a = 2$ with 3459 degrees of freedom, (c) $a = 3$ with 4263 degrees of freedom.

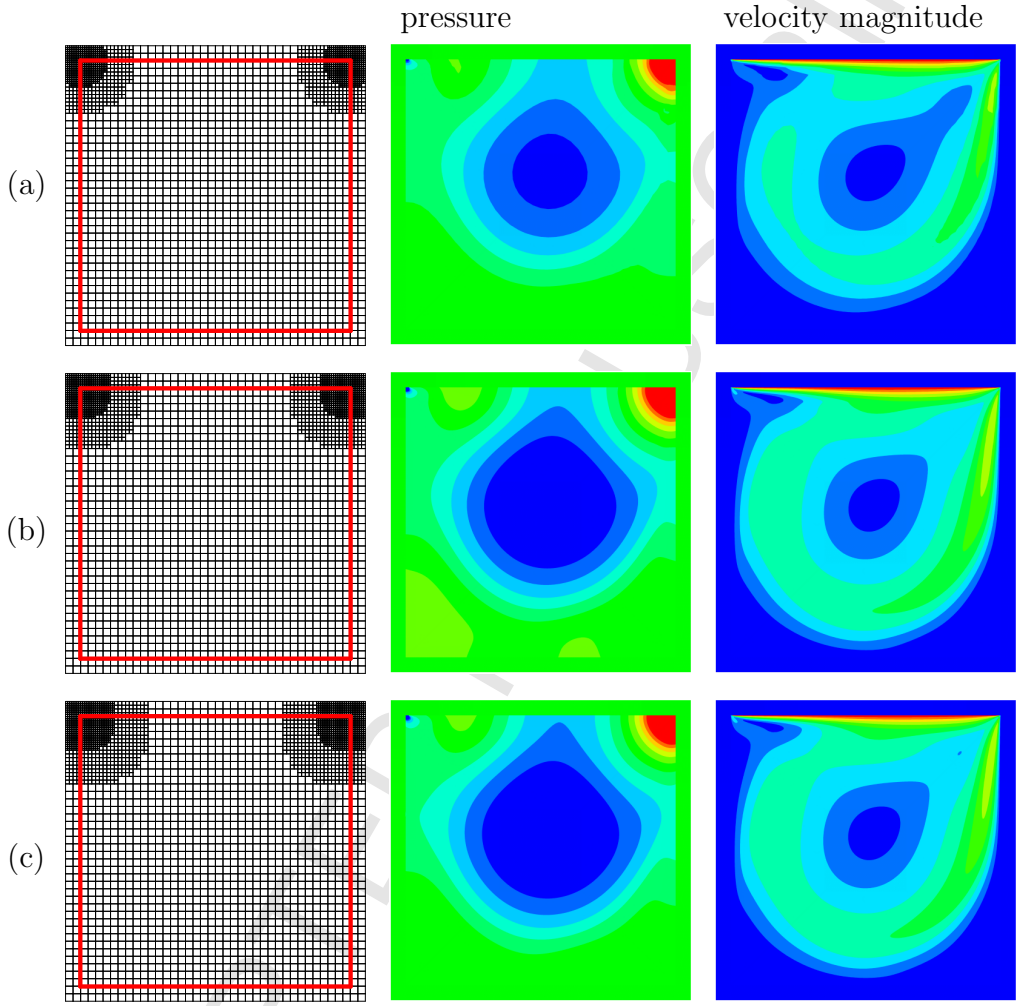

Figure 22: Lid driven cavity flow with $Re = 1000$; mesh plots, pressure and velocity magnitude contour plots; (a) $a = 1$ with 6207 degrees of freedom, (b) $a = 2$ with 7008 degrees of freedom, (c) $a = 3$ with 7887 degrees of freedom.

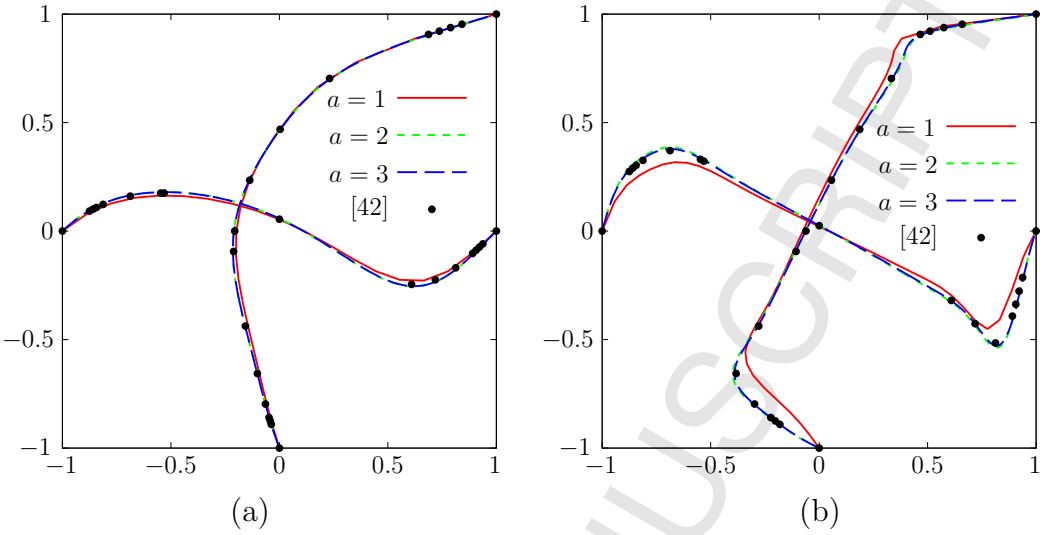

Figure 23: Lid driven cavity flow; velocity profiles along horizontal and vertical cuts through the centre of the cavity; (a) $Re = 100$, $a = 1$, 2, 3 with 2751, 3459, 4263 degrees of freedom, respectively; (b) $Re = 1000$, $a = 1$, 2, 3 with 6207, 7008, 7887 degrees of freedom, respectively.

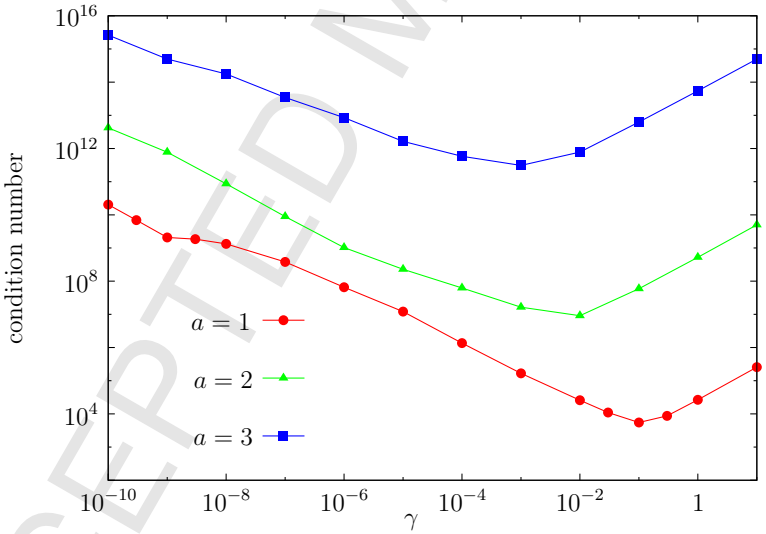

Figure 24: Lid driven cavity flow with $Re = 100$; condition number of system matrix against γ for 20 elements along the edge and cut ratio 1/64; note that, for $\gamma = 0$, the condition numbers are 8.51e+11, 3.89e+17, 3.66e+23 with $a = 1$, 2, 3, respectively.

4.3. Flow Around a Cylinder

The flow around a cylinder at Reynolds number $Re = 40$ is investigated in this section. The computational domain is a square with sides of 30 unit lengths. The cylinder has a diameter of one unit length and is offset from the bottom left corner by 10 horizontal and 15 vertical unit lengths. Slip boundary conditions are applied at the top and bottom boundaries. The prescribed inflow profile on the left boundary is uniform with $\boldsymbol{u} = \{1, 0\}^T$. The outflow boundary on the right is free. The fluid viscosity and density are 0.025 and 1, respectively.

Details of the adaptively refined background meshes are shown in Figure 25. The smallest elements are contained in a small circular region around the cylinder where four levels of refinement are used. The inflow, outflow and slip wall boundaries coincide with the mesh boundary. Here, the Dirichlet boundary conditions are also applied by means of the non-symmetric and penalty-free version of Nitsche's method. At the immersed boundary of the cylinder, the ghost penalty parameter is set to $\gamma = 10^{-3}$ unless otherwise stated. In all computations, the cylinder is represented by 60 straight line segments. The Newton procedure renders asymptotically quadratic convergence as shown in Table 2.

Quality of the solutions: Contour plots of the pressure and the velocity magnitude are shown in Figure 25. In all cases, the pressure does not exhibit any spurious oscillations in the field or near the boundaries.

Convergence: The computations are repeated for a number of different background meshes, which are obtained by varying the size of the base level elements. The convergence of the drag coefficient and of the recirculation length is shown in Figures 26 and 27, respectively. The drag coefficient is obtained from $C_D = (2\,D)/(\rho\,u_{\text{in}}^2\,d)$, where d is the diameter of the cylinder, D represents the horizontal component of the traction forces integrated over the surface of the cylinder and $u_{\text{in}} = 1$ is the inflow velocity. The recirculation length L_w is the distance between the surface of the cylinder and the point in the wake of the cylinder where the horizontal velocity changes from negative to positive. Without requiring dense discretisations, the solutions obtained with quadratic and cubic basis functions quickly converge to $C_D = 1.59$ and $L_w = 2.29$ both of which lie well inside the range of reference values provided in Table 3. For $a = 1$, the results converge at a significantly lower rate. This is attributed to the PSPG/SUPG stabilisation which, for linear basis functions, is based on the incomplete representation of the residual.

	$a = 1$	$a = 2$	$a = 3$
degrees of freedom	16,083	18,639	15,429
	4.25e-00	3.02e-00	3.33e-00
	8.27e-01	7.09e-01	1.02e-00
	4.24e-01	2.77e-01	2.96e-01
	3.64e-02	2.15e-02	2.40e-02
	1.28e-03	9.06e-04	1.08e-03
	1.92e-06	8.60e-07	8.47e-07
	2.56e-12	4.42e-13	5.28e-13

Table 2: Flow around cylinder; convergence of Newton procedure.

	drag coefficient C_D	recirculation length L_w
[43]	1.56	2.22
[44]	1.54	2.28
[45]	1.60	2.29
present work	1.59	2.29

Table 3: Flow around cylinder; reference values for drag coefficient and recirculation length.

Matrix condition numbers: Using $a = 2$ and the background mesh with 4812 degrees of freedom, the computations are repeated for a range of ghost penalty parameters γ. In Figures 28 and 29, the drag coefficients and the condition numbers of the system matrix of each converged solution, respectively, are displayed against γ. It is observed that the cut cell stabilisation can reduce the matrix condition number by a factor of more than 10^9. The minimum matrix condition number of 1.03e+6 is obtained for $\gamma = 1$. However, it is also observed that, for $0 \leq \gamma \leq 0.1$, the drag coefficient is not affected by the stabilisation, whereas it deteriorates as γ increases beyond 1. This agrees well with the observations made for the Laplace problem in Sections 3.2 and 3.3. For $a = 1, 2, 3$, the condition numbers obtained with $\gamma = 0$ and $\gamma = 0.001$ are given in Table 4.

	$a = 1$	$a = 2$	$a = 3$
degrees of freedom	3552	4812	6264
$\gamma = 0$	3.44e+9	1.65e+15	5.20e+19
$\gamma = 0.001$	7.87e+6	6.94e+7	8.37e+9

Table 4: Flow around cylinder; matrix condition numbers.

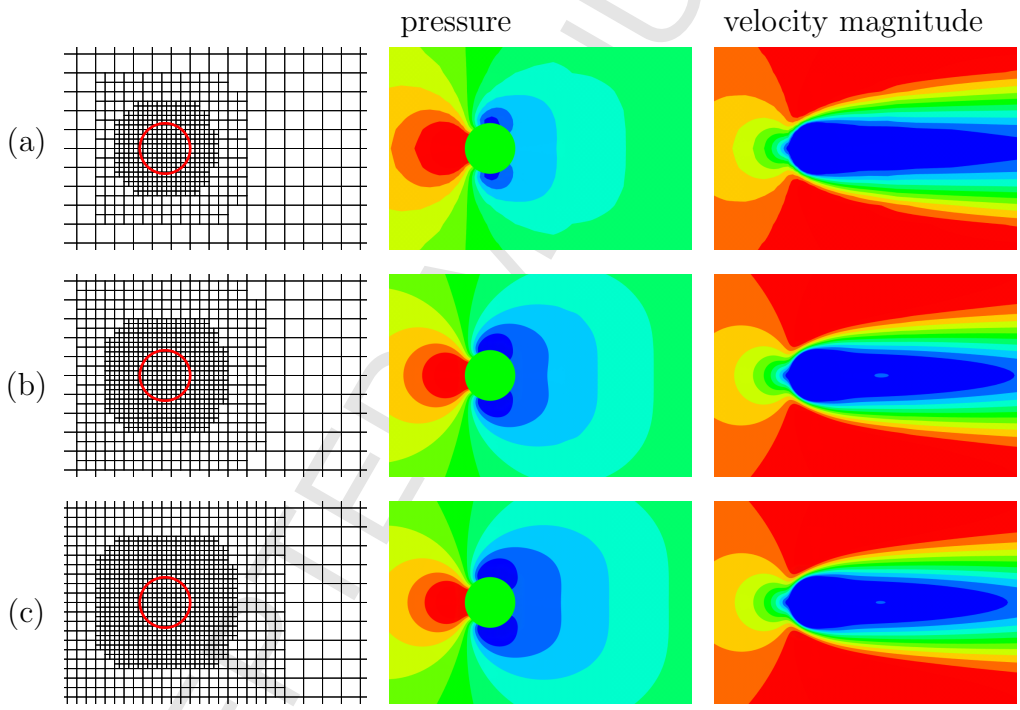

Figure 25: Flow around cylinder; details of meshes, pressure and velocity magnitude contour plots; (a) $a = 1$ with 3552 degrees of freedom, (b) $a = 2$ with 4812 degrees of freedom, (c) $a = 3$ with 6264 degrees of freedom.

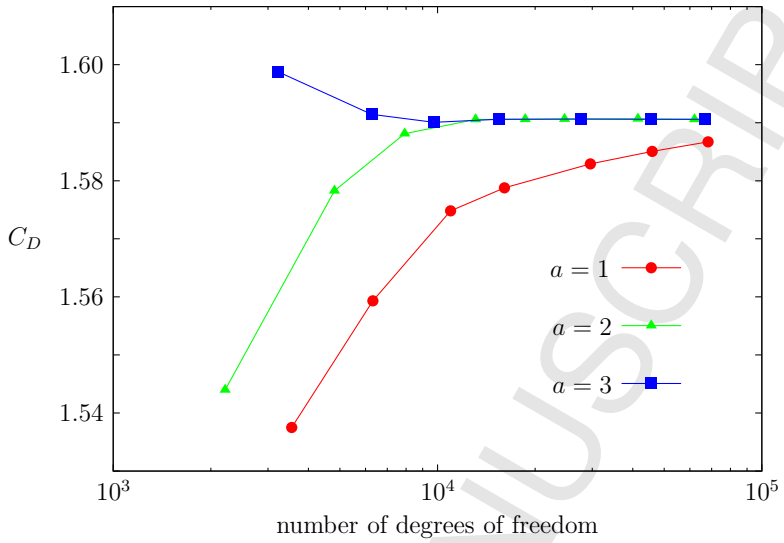

Figure 26: Flow around cylinder; convergence of drag coefficient.

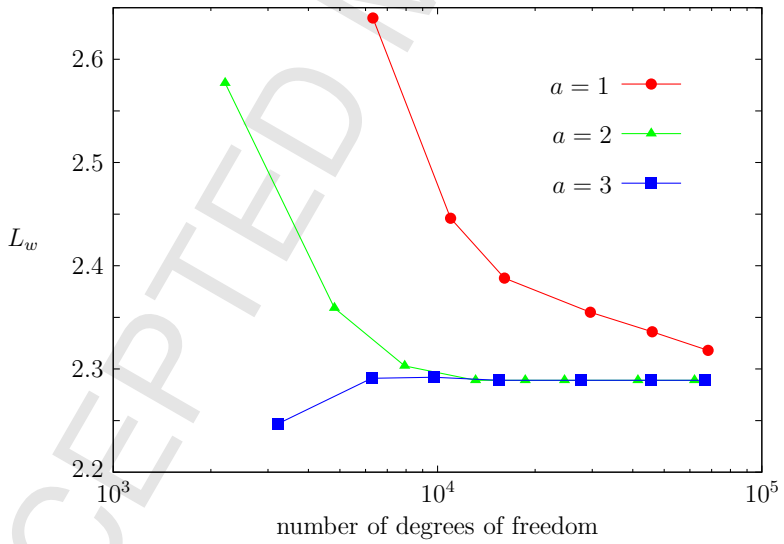

Figure 27: Flow around cylinder; convergence of recirculation length.

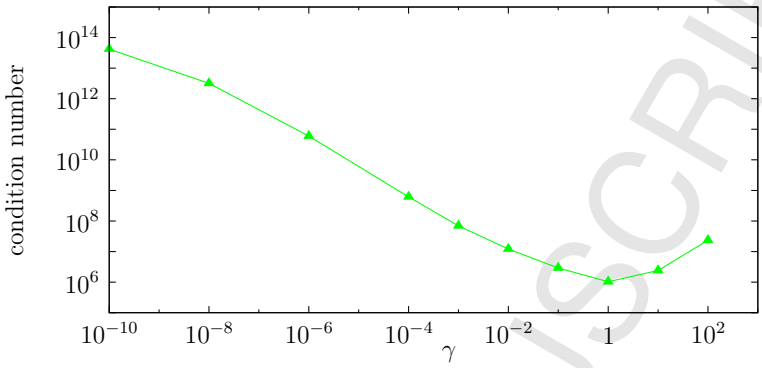

Figure 28: Flow around cylinder; condition number of system matrix against γ for $a = 2$ and 4812 degrees of freedom; for $\gamma = 0$ the condition number is 1.65e+15.

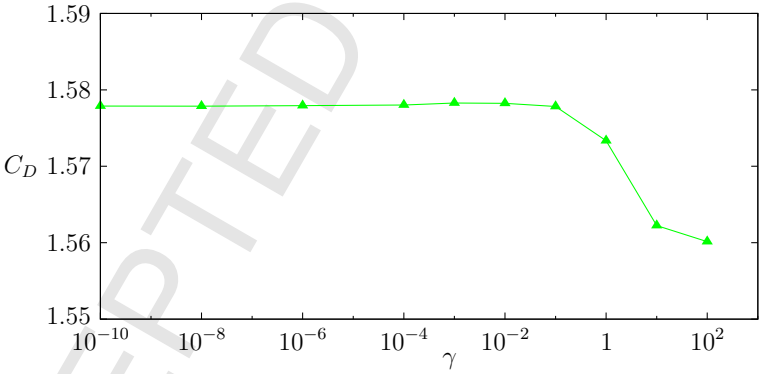

Figure 29: Flow around cylinder; drag coefficient against γ for $a = 2$ and 4812 degrees of freedom.

5. Conclusions

An immersed boundary finite element method based on a hierarchical cartesian b-spline grid and a parameter-free version of Nitsche's method has been proposed. The cut cells are stabilised with ghost penalty terms and integrated with a modified hierarchical quadrature scheme. The performance of the methodology has been extensively tested for the Poisson and for the incompressible Navier-Stokes equations.

All results obtained are accurate and free of spurious oscillations. The cut cell stabilisation is effective and robust. The ghost penalty parameter can be chosen within a wide interval, such that it is large enough to ensure stability, but sufficiently small not to jeopardise the solution accuracy. For the Navier-Stokes equations, the combination of the SUPG/PSPG and the cut cell stabilisation does not cause any spurious behaviour of the solution near the immersed boundaries. The proposed integration scheme for the cut cells has been shown to avoid excessive numbers of quadrature points while still maintaining the solution accuracy.

Future work will be directed at the employment of the presented framework in the context of moving boundary flows and fluid-structure interaction.

References

[1] T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering*, 194:39 (2005) 4135-4195.

[2] Y. Bazilevs, V. M. Calo, T. J. R. Hughes and Y. Zhang, Isogeometric fluid-structure interaction: theory, algorithms, and computations, *Computational Mechanics*, 43:1 (2008) 3-37.

[3] D. J. Benson, Y. Bazilevs, M. C. Hsu and T. J. R. Hughes, Isogeometric shell analysis: the Reissner-Mindlin shell, *Computer Methods in Applied Mechanics and Engineering*, 199 (2010) 276-289.

[4] R. L. Taylor, Isogeometric analysis of nearly incompressible solids, *International Journal for Numerical Methods in Engineering*, 87 (2011) 273-288.

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

[6] F. Cirak, M. Ortiz and P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, *International Journal for Numerical Methods in Engineering*, 47:12 (2000) 2039-2072.

[7] W. G. Dettmer and D. Perić, A computational framework for fluid-structure interaction: finite element formulation and applications, *Computer Methods in Applied Mechanics and Engineering*, 195:41 (2006) 5754-5779.

[8] W. G. Dettmer and D. Perić. On the coupling between fluid flow and mesh motion in the modelling of fluid-structure interaction, *Computational Mechanics*, 43:1 (2008) 81-90.1

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

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

[11] C. Kadapa, W. G. Dettmer and D. Perić, A fictitious domain/distributed Lagrange multiplier based fluid-structure interaction scheme with hierarchical b-spline grids, *Computer Methods in Applied Mechanics and Engineering*, 301 (2016) 1-27.

[12] C. S. Peskin, The immersed boundary method, *Acta Numerica*, 11 (2002) 479-517.

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

[14] E. Burman, M. A. Fernandez, Explicit strategies for incompressible fluid-structure interaction problems: Nitsche type mortaring versus Robin-Robin coupling, *International Journal for Numerical Methods in Engineering*, 97 (2014) 739-758.

[15] W. Jiang, C. Annavarapu, J. E. Dolbow and I. Harari, A robust Nitsche's formulation for interface problems with spline based finite elements, *International Journal for Numerical Methods in Engineering*, 104 (2015) 676-696.

[16] R. A. K. Sanches, P. B. Bornemann and 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.

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

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

[19] D. Schillinger and 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*, 200:47-48 (2011) 3358-3380.

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

[21] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank and T. J. R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed

boundary methods, and T-spline CAD surfaces, *Computer Methods in Applied Mechanics and Engineering*, 249-252 (2012) 116-150.

[22] K. Höllig, *Finite Element Methods with B-Splines*, SIAM, Philadelphia (2003).

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

[24] J. E. Dolbow and I. Harari, An efficient finite element method for embedded interface problems, *International Journal for Numerical Methods in Engineering*, 78 (2009) 229-252.

[25] E. Burman and P. Hansbo, Fictitious domain finite element methods using cut elements: II A stabilized Nitsche method, *Applied Numerical Mathematics*, 62:4 (2012) 328-341.

[26] P. Hansbo, M. G. Larson and S. Zahedi, A cut finite element method for a Stokes interface problem, *Applied Numerical Mathematics*, 85 (2014) 90-114.

[27] E. Burman, S. Claus, P. Hansbo, M. G. Larson and A. Massing, CutFEM: Discretising geometry and partial differential equations, *International Journal for Numerical Methods in Engineering*, 104:7 (2015) 472-501.

[28] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, 36 (1971) 9-15.

[29] E. Burman, A penalty-free non-symmetric Nitsche-type method for the weak imposition of boundary conditions, *SIAM Journal on Numerical Analysis*, 50:4 (2012) 1959-1981.

[30] T. Boiveau and E. Burman, A penalty-free Nitsche type method for the weak imposition of boundary conditions in compressible and incompressible elasticity, *IMA Journal of Numerical Analysis*, (2015) doi: 10.1093/imanum/drv042.

[31] K. Höllig, U. Reif and J. Wipper, Weighted extended b-spline approximation of Dirichlet problems, *SIAM Journal on Numerical Analysis*, 39 (2001) 442-462.

[32] C. Annavarapu, M. Hautefeuille and J. Dolbow, A robust Nitsche's formulation for interface problems, *Computer Methods in Applied Mechanics and Engineering*, 225-228 (2012) 44-54.

[33] A. Embar, J. Dolbow and I. Harari, Imposing Dirichlet boundary conditions with Nitsche's method and spline based finite elements, *International Journal for Numerical Methods in Engineering*, 83:7 (2010) 877-898.

[34] T. E. Tezduyar, S. Mittal, S. E. Ray and R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, *Computer Methods in Applied Mechanics and Engineering*, 94 (1992) 353-371.

[35] T. Tezduyar and S. Sathe, Stabilisation parameters in SUPG and PSPG formulations, *Journal of Computational and Applied Mechanics*, 4:1 (2003) 71-88.

[36] W. G. Dettmer and D. Perić, An analysis of the time integration algorithms for the finite element solutions of incompressible Navier-Stokes equations based on a stabilised formulation, *Computer Methods in Applied Mechanics and Engineering*, 192:9 (2003) 1177-1226.

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

[38] D. F. Rogers, *An Introduction to NURBS with Historical Perspective*, Academic Press, San Diego, CA, U.S.A. (2001).

[39] R. Kraft, Adaptive and linearly independent multilevel b-splines, in: A. L. Mhaut, C. Rabut, L.L. Schumaker (Eds.), *Surface Fitting and Multiresolution Methods*, Vanderbilt University Press, (1997) 209-218.

[40] J. Freund and R. Stenberg, On weakly imposed boundary conditions for second order problems, in M. Cecchi et al., editor, *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids*, (1995) 327-336.

[41] K.P.S. Gahalaut, J.K. Kraus and S.K. Tomar, Multigrid methods for isogeometric discretization, *Computer Methods in Applied Mechanics and Engineering*, 253 (2013) 413-425.

[42] U. Ghia, K. N. Ghia and C. T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multi-grid method, *Journal of Computational Physics*, 48 (1982) 387-411.

[43] D. V. Le, B. C. Khoo and J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *Journal of Computational Physics*, 220 (2006) 109-138.

[44] M. N. Linnick and H. F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *Journal of Computational Physics*, 204 (2005) 157-192.

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

The key contributions of the present work are:

- nonsymmetric and penalty-free Nitsche method applied in the context of an immersed method,
- nonsymmetric and penalty-free Nitsche method applied to the incompressible Navier-Stokes equations,
- ghost-penalty based cut cell stabilisation applied to higher order b-spline basis functions,
- integration of cut cells with sub-cell merging,
- comprehensive study of accuracy and matrix condition numbers.