

Approximate imposition of boundary conditions in immersed boundary methods

Ramon Codina^{*,†} and Joan Baiges

Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

SUMMARY

We analyze several possibilities to prescribe boundary conditions in the context of immersed boundary methods. As basic approximation technique we consider the finite element method with a mesh that does not match the boundary of the computational domain, and therefore Dirichlet boundary conditions need to be prescribed in an approximate way. As starting variational approach we consider Nitsche's methods, and we then move to two options that yield non-symmetric problems but that turned out to be robust and efficient. The essential idea is to use the degrees of freedom of certain nodes of the finite element mesh to minimize the difference between the exact and the approximated boundary condition. Copyright © 2009 John Wiley & Sons, Ltd.

Received 11 March 2008; Revised 15 April 2009; Accepted 29 April 2009

KEY WORDS: immersed boundaries; Nitsche's method; approximate boundary conditions; finite element methods

1. INTRODUCTION

The numerical approximation of boundary value problems on non-matching grids has the obvious advantage of the freedom to generate the grid. Only a grid *covering* the computational domain has to be created, leaving the imposition of boundary conditions to the numerical formulation being used. The physical boundary is contained in the domain actually discretized, which is the reason why these methods are called *immersed boundary methods* (IBM).

The price to be paid when using IBM is a lack of control on the grid close to the boundary, which may be very important in flow problems with boundary layers, for example. However, this difficulty may be dealt with using composite grids or Chimera-type techniques as that proposed

*Correspondence to: Ramon Codina, Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.

†E-mail: ramon.codina@upc.edu

Contract/grant sponsor: Departament d'Universitats, Recerca i Societat de la Informació of the Generalitat de Catalunya (Catalan Government)

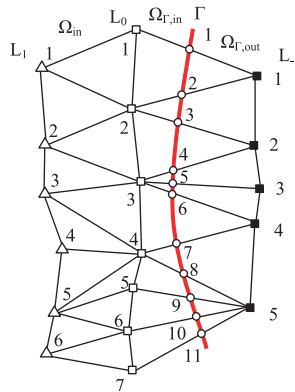


Figure 1. Setting.

in [1]. Nevertheless, we will not touch this point here, nor the aspect that makes methods with non-matching grids really attractive and that has been our main motivation (see [2]), which is the modeling of flows with moving boundaries keeping the grid fixed. In this case, not only the freedom to generate this grid is important but also the fact that re-gridding as the computational domain evolves may be avoided. This is probably the reason why the so-called fixed grid methods have received and are currently receiving a great deal of attention in the numerical literature (see, for example, the reviews [3–5]). Since the fixed grid used is often Cartesian, these methods can be found under the keywords *Cartesian grid methods*.

Our attention will be focused to finite element methods for flow problems, but the ideas to be presented are extendable to other numerical formulations and other physical problems. However, some of the difficulties we shall mention are characteristic of flow problems. Likewise, we will consider general non-structured meshes, the application to Cartesian meshes being obvious. Furthermore, the exposition will be based on 2D linear triangular meshes, although, again, extensions to 3D and other finite element interpolations are straightforward.

Let us describe the problem to be solved. Consider the situation depicted in Figure 1. A domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, with boundary $\Gamma = \partial\Omega$ (dark curve in Figure 1) is covered by a mesh that occupies a domain $\Omega_h = \Omega_{in} \cup \Omega_\Gamma$, where $\Omega_{in} \subset \Omega$ is formed by the elements interior to Ω and Ω_Γ is formed by a set of elements cut by Γ . In turn, let us split $\Omega_\Gamma = \Omega_{\Gamma,in} \cup \Omega_{\Gamma,out}$, where $\Omega_{\Gamma,in} = \Omega \cap \Omega_\Gamma$ and $\Omega_{\Gamma,out}$ is the interior of $\Omega_\Gamma \setminus \Omega_{\Gamma,in}$. Note that $\Omega = \Omega_{in} \cup \Omega_{\Gamma,in}$. For simplicity, we will assume that the intersection of Γ with the element domains is a piecewise polynomial curve (in 2D) or surface (in 3D) of the same order as the finite element interpolation. This will be used in the proof of stability presented in Section 3.2, although in fact it is not necessary to apply the method.

Suppose we want to solve a boundary value problem for the unknown u in Ω with the mesh of Ω_h already created and boundary conditions $u = \bar{u}$ on Γ . The obvious choice would be:

- Obtain the nodes of Γ (circles in Figure 1) from the intersection with the element edges.
- Split the elements of $\Omega_{\Gamma,in}$ so as to obtain a grid matching the boundary Γ .
- Prescribe the boundary condition $u_h = \bar{u}$ in the classical way, where u_h denotes the approximate solution.

This strategy leads to a local remeshing close to Γ that is involved from the computational point of view. Obviously, the implementation of the strategy described is very simple for unstructured

simplicial meshes, but it is not so easy if one wants to use other element shapes and, definitely, prevents from using Cartesian meshes. Moreover, if the boundary Γ evolves in time (a situation not considered in the following) the number of degrees of freedom changes at each time instant, thus modifying the structure and sparsity of the matrix of the final algebraic system. This is clearly an inconvenience even when using unstructured simplicial meshes.

Other possibilities can be found in the literature. One of them is the widely used *Immersed Boundary Method* in its original form [6], which consists in adding point-wise penalty forces in the domain boundary so that the boundary conditions are fulfilled. The method is first-order accurate even if second-order approximation schemes are used, although *formal second-order accuracy* has been reported in [7]. The more recent *Immersed Interface Method* achieves higher-order accuracy by avoiding the use of the Dirac delta distribution to define the forcing terms (see [8–10]).

Another approach is the use of Lagrange multipliers to enforce the boundary conditions. However, the finite element subspaces for the bulk and Lagrange multiplier fields must satisfy the classical inf-sup condition proposed by Babuška [11], which usually leads to the need for stabilization (see [12–14]). Moreover, additional degrees of freedom must be added to the problem. The use of Lagrange multipliers is the basis of the *fictitious domain method* [15, 16] (see also Chapter VIII in [17]).

Recently, *hybrid Cartesian/IBM* have been developed for Cartesian grids, which use the grid nodes closest to the boundary to enforce boundary conditions [18–20]. The method is second-order accurate, but it does not guarantee that the distance between u_h and \bar{u} in Γ is minimized.

A *discontinuous-Galerkin-based IBM* is proposed in [21], which consists in switching elements intersected by the boundary to a discontinuous-Galerkin approximation and impose the Dirichlet boundary conditions strongly. Although optimal-order accuracy is achieved, the method requires additional degrees of freedom.

The target we pose is therefore *to impose the Dirichlet boundary conditions (in an approximate way) without adding new degrees of freedom except from those of the original mesh in Ω_h , in such a way that the distance between u_h and \bar{u} in Γ is minimized in a certain norm*. In the following section we describe Nitsche's method as a first approach to achieve this.

In Section 3 we introduce a first modification of Nitsche's method, the main advantage being that there are no parameters to choose and there is no ill-conditioning of the final algebraic system due to large factors enforcing the boundary condition. This is crucial for general flow problems in which there is no rule to choose the parameter appearing in Nitsche's method. The essential idea is to use the degrees of freedom associated with $\Omega_{\Gamma, \text{out}}$ to prescribe approximately the boundary conditions, while the discrete version of the differential operator is only imposed for nodes in Ω_{in} . The drawback is that the problem obtained is not symmetric even for symmetric problems, although the problems we are interested in are non-symmetric. In particular, we have applied the methods to be described to transient incompressible flow problems in moving domains in [2].

The formulation of Section 3 turns out to be accurate, but depending on the way the physical boundary Γ cuts the elements in Ω_h may lead to ill-conditioned matrices and difficulties in the convergence of iterative schemes for non-linear problems. We present a modification in Section 4. In this case, the idea is to solve the problem only in the domain formed by the elements inside Ω , and prescribe the boundary conditions using the degrees of freedom associated with the first layer of nodes *inside* Ω , that is to say, on $\partial\Omega_{\text{in}}$.

Numerical examples showing the performance of the different methods described are presented in Section 5, and some concluding remarks close the paper in Section 6.

2. NITSCHKE'S METHOD REVISITED

Our intention is to consider flow problems and, in particular, the scalar convection–diffusion–reaction equation and the incompressible Navier–Stokes equations. However, for the exposition it is enough to consider the former, leaving the latter for the numerical examples.

Let us consider the problem

$$\mathcal{L}u := -k\Delta u + \mathbf{a} \cdot \nabla u + su = f \quad \text{in } \Omega \quad (1)$$

$$u = \bar{u} \quad \text{on } \Gamma = \partial\Omega \quad (2)$$

where $k > 0$, \mathbf{a} is the advection velocity, $s \geq 0$, f is a given forcing function and \bar{u} is the given Dirichlet boundary condition. We assume that the subdomain Ω is polyhedral, and covered by the domain Ω_h , as explained in Section 1.

Let $\mathcal{P}_h = \{K\}$ be a finite element partition of Ω_h from which we construct the finite element space $V_h \subset H^1(\Omega_h)$ (we will consider V_h made of continuous functions). On $V_h \times V_h$ we define the bilinear form

$$B(u_h, v_h) = k(\nabla u_h, \nabla v_h) + (\mathbf{a} \cdot \nabla u_h, v_h) + s(u_h, v_h) \quad (3)$$

where (\cdot, \cdot) is the L^2 -inner product in Ω , not in Ω_h . In general, the integral of two functions f_1 and f_2 in a region ω will be denoted by $\langle f_1, f_2 \rangle_\omega$. The norm in a space X will be indicated as $\|\cdot\|_X$, except when $X = L^2(\Omega)$, case in which the subscript will be omitted.

Nitsche's method applied to problem (1) and (2) reads: find $u_h \in V_h$ such that

$$\begin{aligned} B(u_h, v_h) - k\langle \partial_n u_h, v_h \rangle_\Gamma - k\langle u_h, \partial_n v_h \rangle_\Gamma + \frac{\alpha k^*}{h} \langle u_h, v_h \rangle_\Gamma \\ = \langle f, v_h \rangle_\Omega - k\langle \bar{u}, \partial_n v_h \rangle_\Gamma + \frac{\alpha k^*}{h} \langle \bar{u}, v_h \rangle_\Gamma, \quad \forall v_h \in V_h \end{aligned} \quad (4)$$

where $\alpha > 0$ is a numerical parameter, k^* a parameter with the same dimensions as k (here introduced with the only purpose of making the equations dimensionally consistent) and h is the element size, that is to say, $h = \max_K h_K$, with $h_K = \text{diam } K$, $K \in \mathcal{P}_h$. For simplicity, we will consider quasi-uniform partitions \mathcal{P}_h .

It is observed that, apart from the way to impose the boundary conditions, (4) is based on the standard Galerkin method to solve the convection–diffusion–reaction equation. This method is stable only for high values of the diffusion coefficient k . Even though in the examples we will consider convection dominated flows solved using a stabilized formulation, for the sake of conciseness the exposition will be developed in the diffusion dominated case. Likewise, we will consider \mathbf{a} constant, for simplicity.

In the following we will try to 'rederive' method (4). This will allow us to introduce the modification we propose. Let us consider the splitting $V_h = V_{h,0} \oplus V_{h,\Gamma}$, where $V_{h,0}$ is the subspace of V_h of functions vanishing at the nodes outside $\bar{\Omega}_{\text{in}}$ and $V_{h,\Gamma}$ the complement, that is, the subspace of functions that are zero at the nodes of $\bar{\Omega}_{\text{in}}$ (including layer L_0 in Figure 1). According to this splitting, we may split the unknown as $u_h = u_{h,0} + u_{h,\Gamma}$ and the test functions as $v_h = v_{h,0} + v_{h,\Gamma}$.

Nitsche’s method (4) can be obtained from the following set of equations:

$$B(u_{h,0}, v_{h,0}) - k \langle \partial_n u_{h,0}, v_{h,0} \rangle_\Gamma + B(u_{h,\Gamma}, v_{h,0}) - k \langle \partial_n u_{h,\Gamma}, v_{h,0} \rangle_\Gamma = \langle f, v_{h,0} \rangle_\Omega \tag{5}$$

$$B(u_{h,0}, v_{h,\Gamma}) - k \langle \partial_n u_{h,0}, v_{h,\Gamma} \rangle_\Gamma + B(u_{h,\Gamma}, v_{h,\Gamma}) - k \langle \partial_n u_{h,\Gamma}, v_{h,\Gamma} \rangle_\Gamma = \langle f, v_{h,\Gamma} \rangle_\Omega \tag{6}$$

$$-k \langle \partial_n v_{h,0}, u_{h,0} \rangle_\Gamma - k \langle \partial_n v_{h,0}, u_{h,\Gamma} \rangle_\Gamma = -k \langle \partial_n v_{h,0}, \bar{u} \rangle_\Gamma \tag{7}$$

$$-k \langle \partial_n v_{h,\Gamma}, u_{h,0} \rangle_\Gamma - k \langle \partial_n v_{h,\Gamma}, u_{h,\Gamma} \rangle_\Gamma = -k \langle \partial_n v_{h,\Gamma}, \bar{u} \rangle_\Gamma \tag{8}$$

$$\frac{\alpha k^*}{h} \langle u_{h,0}, v_{h,0} \rangle_\Gamma + \frac{\alpha k^*}{h} \langle u_{h,\Gamma}, v_{h,0} \rangle_\Gamma = \frac{\alpha k^*}{h} \langle \bar{u}, v_{h,0} \rangle_\Gamma \tag{9}$$

$$\frac{\alpha k^*}{h} \langle u_{h,0}, v_{h,\Gamma} \rangle_\Gamma + \frac{\alpha k^*}{h} \langle u_{h,\Gamma}, v_{h,\Gamma} \rangle_\Gamma = \frac{\alpha k^*}{h} \langle \bar{u}, v_{h,\Gamma} \rangle_\Gamma \tag{10}$$

The first two Equations (5) and (6) are obtained by multiplying the differential equation by $v_{h,0}$ and $v_{h,\Gamma}$ and integrating by parts. Note that no boundary conditions are imposed, and thus the solution of (5) and (6) is not unique. Equations (7) and (8) can be understood as a weak form of the boundary condition $u_h = \bar{u}$, weighting this equation by $-k \partial_n v_{h,0}$ and $-k \partial_n v_{h,\Gamma}$. These two equations are needed to keep the symmetry of the problem when $B(u_h, v_h) = B(v_h, u_h)$, that is to say, when $\mathbf{a} = \mathbf{0}$ (see (3)). Finally, Equations (9) and (10) are also obtained as a weak form of the boundary condition $u_h = \bar{u}$, weighting now this equation by $v_{h,0}$ and $v_{h,\Gamma}$.

Obviously, Equations (5)–(10) are all consistent, in the sense that if u_h is replaced by the exact solution u of problems (1) and (2) they hold exactly, provided this solution is regular enough. However, system (5)–(10) as a whole is *overdetermined*, and there are several possibilities to extract a system of algebraic equations with a unique solution from it. In particular, Nitsche’s method (4) is obtained by adding together all the equations. The method to be proposed in the following section can be understood as the method obtained *keeping only* (5) and (10). In fact, for stability reasons described later it turns out to be convenient to *subtract* (7) from (5).

Before describing an alternative to Nitsche’s method, let us comment on the role played by the factor $(\alpha k^*/h)$. Suppose that $\mathbf{a} = \mathbf{0}$, so that B is symmetric, and define the functionals $J_1(u_{h,0}, u_{h,\Gamma}) = \frac{1}{2} B(u_{h,0} + u_{h,\Gamma}, u_{h,0} + u_{h,\Gamma}) - k \langle \partial_n (u_{h,0} + u_{h,\Gamma}), (u_{h,0} + u_{h,\Gamma}) \rangle_\Gamma - \langle f, u_{h,0} + u_{h,\Gamma} \rangle_\Omega - k \langle \bar{u}, \partial_n (u_{h,0} + u_{h,\Gamma}) \rangle_\Gamma$ and $J_2(u_{h,0}, u_{h,\Gamma}) = (\alpha k^*/h) \|u_{h,0} + u_{h,\Gamma} - \bar{u}\|_{L^2(\Gamma)}^2$. If $\delta_{(v_{h,0}, v_{h,\Gamma})}$ denotes the weak (Gâteaux) derivative of a functional in the direction of $v_h = (v_{h,0}, v_{h,\Gamma})$ we may write problem (4) as

$$\delta_{(v_{h,0}, v_{h,\Gamma})} (J_1(u_{h,0}, u_{h,\Gamma}) + J_2(u_{h,0}, u_{h,\Gamma})) = 0 \tag{11}$$

From this expression it follows that satisfying the Dirichlet boundary condition must *compete* with satisfying the differential equation, $\alpha k^*/h$ being the weight of the former. Moreover, since the norm $h^{-1/2} \|\cdot\|_{L^2(\Gamma)}$ is equivalent to the norm of $\|\cdot\|_{H^{1/2}(\Gamma)}$ in V_h (see [22, 23]), the relevant weighting is in fact the parameter α . The higher the value of α , the better the approximation to the boundary condition at the expense of a poorer approximation to the differential equation. However, for any value of α it is possible to show that the method is stable and optimally convergent (in fact, stability is even easier to show than for the method to be presented in the following section). See [24] for a proof, including more general boundary conditions than used here (although for Poisson’s problem). The good performance of Nitsche’s method has been exploited also in other contexts, such as the imposition of boundary conditions for discontinuous finite element approximations (see

the original work in [25] and the extension in [26], for example), the imposition of transmission conditions in domain decomposition with non-matching grids (as in [27, 28], among many others) or also in some stabilized finite element methods for which this method fits nicely [29].

Finally, let us remark that the volume integrals in (5) and (6) are performed over $\Omega = \Omega_{\text{in}} \cup \Omega_{\Gamma, \text{in}}$. Integrals over Ω_{in} are easily computed, but in order to compute integrals over $\Omega_{\Gamma, \text{in}}$ some care is needed. The simplest approach is to split the elements of $\Omega_{\Gamma, \text{in}}$ so as to obtain a grid matching the boundary Γ , and then proceed to compute the integrals over the resulting subelements (see [30]). Note that this splitting does not affect the degrees of freedom of the problem.

3. A FIRST MODIFICATION: USING EXTERNAL DEGREES OF FREEDOM

3.1. The first method proposed

The essential idea of the method we propose first is to use only equations (5) and (10) above. As it has been mentioned in the previous section, it is also convenient, mainly for the stability analysis, to subtract (7) from (5). Thus, the problem to be solved is: find $u_{h,0} \in V_{h,0}$ and $u_{h,\Gamma} \in V_{h,\Gamma}$ such that

$$B(u_{h,0}, v_{h,0}) + B(u_{h,\Gamma}, v_{h,0}) - k \langle \partial_n u_{h,0}, v_{h,0} \rangle_{\Gamma} - k \langle \partial_n u_{h,\Gamma}, v_{h,0} \rangle_{\Gamma} + k \langle \partial_n v_{h,0}, u_{h,0} \rangle_{\Gamma} + k \langle \partial_n v_{h,0}, u_{h,\Gamma} \rangle_{\Gamma} = \langle f, v_{h,0} \rangle_{\Omega} + k \langle \partial_n v_{h,0}, \bar{u} \rangle_{\Gamma} \quad (12)$$

$$\frac{\alpha k^*}{h} \langle u_{h,0}, v_{h,\Gamma} \rangle_{\Gamma} + \frac{\alpha k^*}{h} \langle u_{h,\Gamma}, v_{h,\Gamma} \rangle_{\Gamma} = \frac{\alpha k^*}{h} \langle \bar{u}, v_{h,\Gamma} \rangle_{\Gamma} \quad (13)$$

for all $v_{h,0} \in V_{h,0}$ and $v_{h,\Gamma} \in V_{h,\Gamma}$.

Equation (13) can be equivalently written as

$$\delta_{(0, v_{h,\Gamma})} J_2(u_{h,0}, u_{h,\Gamma}) = 0 \quad (14)$$

From this equation it is clear that *the component $u_{h,\Gamma}$ of the unknown is determined from the condition that the distance between $u_{h,0} + u_{h,\Gamma}$ and \bar{u} is minimized in the norm of $L^2(\Gamma)$* . Comparing this equation with (11), it is also seen that now this minimization does not compete with the satisfaction of the differential equation (in weak sense). Obviously, the parameter $(\alpha k^*/h)$ is here unnecessary, and it has been introduced only to compare the resulting method with (4).

Let us enumerate four major differences of (12) and (13) with respect to (4):

1. When Γ coincides with $\partial\Omega_h$, the boundary condition is imposed exactly (provided \bar{u} is a finite element function).
2. There are no parameters to be tuned ($(\alpha k^*/h)$ can be canceled out in (13)).
3. The method is non-symmetric, even if B is symmetric.
4. The method is not well defined when Γ coincides with $\partial\Omega_{\text{in}}$.

The first two points are improvements with respect to Nitsche's method. In particular, they explain why the approximation of boundary conditions is in general better with our approach, as we have experimented from numerical tests. The third point is a drawback from the implementation point of view only for symmetric problems, and not for the flow problems we are interested in. The important issue is point 4. Clearly, when $\Gamma = \partial\Omega_{\text{in}}$ (13) yields $0=0$. In this case, elements outside Ω_{in} could be eliminated and the case reduced to the first one. However, this situation may

be encountered if Ω_{in} is a domain moving in time inside Ω_h . In general, when Γ is close to $\partial\Omega_{\text{in}}$ we may expect instability problems. Small variations in \bar{u} may yield large variations in $u_{h,\Gamma}$. This fact will be used to motivate the method proposed in Section 4. It is worth to mention that this type of instabilities are also encountered in other methods for which modifications are also required (see [19, 31]). We will come back to this point in Section 4.

Precise conditions under which the method is stable are discussed in the following subsection.

3.2. Stability

In this subsection we prove the following result: *if Γ is kept away from $\partial\Omega_{\text{in}}$ the formulation given by (12) and (13) is stable.* As a consequence, the discrete problem admits a unique solution.

Proving this fact requires some analytical technicalities that will make us depart from the line of formulating new methods rather than analyzing them. However, we believe this conclusion is important and deserves this parenthesis in the main syllabus of the paper.

3.2.1. Preliminary result. We will make use of a general result applicable to coupled systems of variational equations of the form

$$a_{11}(u_1, v_1) + a_{12}(u_2, v_1) = l_1(v_1) \quad (15)$$

$$a_{21}(u_1, v_2) + a_{22}(u_2, v_2) = l_2(v_2) \quad (16)$$

where $u_1, v_1 \in V_1$, $u_2, v_2 \in V_2$, a_{ij} is a bilinear form defined on $V_j \times V_i$ and l_i a linear form on V_i , a Banach space with norm $\|\cdot\|_i$, $i, j = 1, 2$. We assume that all the forms a_{ij} are continuous and a_{ii} are coercive. Let C_{ij} be the constants defined by the inequalities

$$\begin{aligned} a_{11}(v_1, v_1) &\geq C_{11}\|v_1\|_1^2, & a_{22}(v_2, v_2) &\geq C_{22}\|v_2\|_2^2 \\ a_{12}(v_2, v_1) &\leq C_{12}\|v_1\|_1\|v_2\|_2, & a_{21}(v_1, v_2) &\leq C_{21}\|v_1\|_1\|v_2\|_2 \end{aligned}$$

We will now prove that if

$$C_{12}C_{21} < C_{11}C_{22} \quad (17)$$

then there exists a constant $C > 0$ such that for all $(u_1, u_2) \in V_1 \times V_2$ there exists $(v_1, v_2) \in V_1 \times V_2$ such that

$$\begin{aligned} B((u_1, u_2), (v_1, v_2)) &:= a_{11}(u_1, v_1) + a_{12}(u_2, v_1) + a_{21}(u_1, v_2) + a_{22}(u_2, v_2) \\ &\geq C(\|u_1\|_1 + \|u_2\|_2)(\|v_1\|_1 + \|v_2\|_2) \end{aligned}$$

that is to say, *problem (15)–(16) is stable.*

In the following, C will denote a generic positive constant, not necessarily the same at different appearances. In the case in which (15) and (16) comes from a finite element approximation, the constant C will be independent of h and inequality (17) will be assumed to hold uniformly in h .

Let us start noting that using Young's inequality we have

$$\begin{aligned} B((u_1, u_2), (u_1, 0)) &\geq C_{11}\|u_1\|_1^2 - C_{12}\left(\frac{\beta_1}{2}\|u_1\|_1^2 + \frac{1}{2\beta_1}\|u_2\|_2^2\right) \\ B((u_1, u_2), (0, u_2)) &\geq C_{22}\|u_2\|_2^2 - C_{21}\left(\frac{\beta_2}{2}\|u_2\|_2^2 + \frac{1}{2\beta_2}\|u_1\|_1^2\right) \end{aligned}$$

where β_1 and β_2 are positive constants to be determined. Thus, for any $\gamma > 0$ we have

$$B((u_1, u_2), (u_1, \gamma u_2)) \geq \left(C_{11} - C_{12} \frac{\beta_1}{2} - C_{21} \frac{1}{2\beta_2} \right) \|u_1\|_1^2 + \left(C_{22}\gamma - C_{21} \frac{\beta_2}{2} \gamma^2 - C_{12} \frac{1}{2\beta_1} \right) \|u_2\|_2^2$$

The constants C_{12} and C_{21} must be such that there exists β_1 , β_2 and γ for which

$$C_{11} - C_{12} \frac{\beta_1}{2} - C_{21} \frac{1}{2\beta_2} > 0 \tag{18}$$

$$C_{22}\gamma - C_{21} \frac{\beta_2}{2} \gamma^2 - C_{12} \frac{1}{2\beta_1} > 0 \tag{19}$$

Condition (18) holds if

$$\beta_1 < \frac{2C_{11}\alpha_1}{C_{12}}, \quad \frac{1}{\beta_2} < \frac{2C_{11}\alpha_2}{C_{21}}, \quad \alpha_1 + \alpha_2 = 1$$

Condition (19) requires then that

$$\gamma > A\gamma^2 + B, \quad A := \frac{C_{21}^2}{4C_{11}C_{22}\alpha_2}, \quad B := \frac{C_{12}^2}{4C_{11}C_{22}\alpha_1}$$

a condition that is possible to fulfill if

$$AB < \frac{1}{4} \iff C_{12}C_{21} < 2C_{11}C_{22}\sqrt{\alpha_1\alpha_2}$$

Since $\alpha_1 + \alpha_2 = 1$, the maximum of $\sqrt{\alpha_1\alpha_2}$ is $\frac{1}{2}$, from where the result follows.

3.2.2. *Some useful relationships.* The next step is to prove some inequalities that will be used later on. These inequalities make use of the inverse estimates (see [22, 23]):

$$\|v_h\|_{L^\infty(\omega)}^2 \leq \frac{C}{h^d} \|v_h\|_{L^2(\omega)}^2 \tag{20}$$

$$\|\nabla v_h\|_{L^2(\omega)}^2 \leq \frac{C}{h^2} \|v_h\|_{L^2(\omega)}^2 \tag{21}$$

where ω is any patch of elements of \mathcal{P}_h (recall that this partition is assumed to be quasi-uniform) and v_h is a finite element function. Because of the assumption on the shape of Γ , ω can be also formed by subdomains of the form $K \cap \Omega_{\Gamma, \text{in}}$, $K \in \mathcal{P}_h$.

From these inequalities one can prove the following:

$$\|v_{h,\Gamma}\|^2 \leq C\delta_1 h \|v_{h,\Gamma}\|_{L^2(\Gamma)}^2 \tag{22}$$

$$\|\nabla v_{h,\Gamma}\|^2 \leq C \frac{1}{\delta_1 h} \|v_{h,\Gamma}\|_{L^2(\Gamma)}^2 \tag{23}$$

$$\frac{\delta_1}{\delta_2^2 h} \|v_{h,0}\|_{L^2(\Gamma)}^2 \leq C \|\nabla v_{h,0}\|_{L^2(\Omega_{\Gamma,\text{in}})}^2 \quad (24)$$

$$\|\partial_n v_h\|_{L^2(\Gamma)}^2 \leq \frac{C}{\delta_1 h} \|\nabla v_h\|_{L^2(\Omega_{\Gamma,\text{in}})}^2 \quad (25)$$

where (see Figure 1)

$$\delta_1 = \frac{1}{h} \min_{\mathbf{x} \in L_0} \text{dist}(\mathbf{x}, \Gamma), \quad \delta_2 = \frac{1}{h} \max_{\mathbf{x} \in L_{-1}} \text{dist}(\mathbf{x}, \Gamma)$$

Let us start noting that (22) is a direct consequence of the shape of $\Omega_{\Gamma,\text{in}}$ and that $v_{h,\Gamma}$ vanishes at the nodes in the interior of this subdomain. The distance from Γ to the nodes of L_0 can be bounded by $C\delta_1 h$, where $1 \leq C$ is a constant, which will be bounded as $h \rightarrow 0$ because of the quasi-uniformity of the partition.

The proof of (23) is as follows:

$$\begin{aligned} \|\nabla v_{h,\Gamma}\|^2 &= \int_{\Omega_{\Gamma,\text{in}}} |\nabla v_{h,\Gamma}|^2 \quad (v_{h,\Gamma} \text{ is zero elsewhere}) \\ &\leq \frac{C}{\delta_1^2 h^2} \int_{\Omega_{\Gamma,\text{in}}} |v_{h,\Gamma}|^2 \quad (\text{by (21) and Poincaré's inequality}) \\ &\leq \frac{C}{\delta_1 h} \int_{\Gamma} |v_{h,\Gamma}|^2 \quad (\text{by (22)}) \\ &= \frac{C}{\delta_1 h} \|v_{h,\Gamma}\|_{L^2(\Gamma)}^2 \end{aligned}$$

For the proof of (24), let K be an element crossed by Γ and $E = K \cap \Gamma$. We have

$$\begin{aligned} \int_E v_{h,0}^2 &\leq \delta_2^2 h^2 \int_E \|\nabla v_{h,0}\|_{L^\infty(K)}^2 \\ &\leq C \delta_2^2 h^2 h^{d-1} \|\nabla v_{h,0}\|_{L^\infty(K)}^2 \\ &\leq C \frac{\delta_2^2}{\delta_1} h^{d+1} h^{-d} \|\nabla v_{h,0}\|_{L^2(K \cap \Omega_{\Gamma,\text{in}})}^2 \quad (\text{by (20)}) \end{aligned}$$

from where (24) is obtained from summation over all E that form Γ . Finally, (25) follows again from the shape of $\Omega_{\Gamma,\text{in}}$.

3.2.3. Application to the first method proposed. Finally, we will apply (22)–(25) to show that condition (17) holds, and thus the method given by (12) and (13) is stable. Let us define the bilinear forms

$$\begin{aligned} a_{0,0}(u_{h,0}, v_{h,0}) &:= B(u_{h,0}, v_{h,0}) - k \langle \partial_n u_{h,0}, v_{h,0} \rangle_{\Gamma} + k \langle \partial_n v_{h,0}, u_{h,0} \rangle_{\Gamma} \\ a_{0,\Gamma}(u_{h,\Gamma}, v_{h,0}) &:= B(u_{h,\Gamma}, v_{h,0}) - k \langle \partial_n u_{h,\Gamma}, v_{h,0} \rangle_{\Gamma} + k \langle \partial_n v_{h,0}, u_{h,\Gamma} \rangle_{\Gamma} \end{aligned}$$

$$a_{\Gamma,0}(u_{h,0}, v_{h,\Gamma}) := \langle u_{h,0}, u_{h,\Gamma} \rangle_{\Gamma}$$

$$a_{\Gamma,\Gamma}(u_{h,\Gamma}, v_{h,\Gamma}) := \langle u_{h,\Gamma}, u_{h,\Gamma} \rangle_{\Gamma}$$

and the norms

$$\| \| v_h \| \|_0^2 := k \| \nabla v_h \|^2 + s \| v_{h,0} \|^2, \quad \| \| v_h \| \|_{\Gamma} := \| v_h \|_{L^2(\Gamma)}$$

As it has been mentioned in Section 2, we assume that the problem is diffusion dominated. More precisely, if $a = |\mathbf{a}|$, in what follows we assume that h is such that

$$k - C \frac{ah}{2} \frac{\delta_2^2}{\delta_1} \geq C_k k, \quad 0 < C_k < 1 \quad (26)$$

for a constant C introduced next.

We have to check (17), and therefore we need to estimate the coercivity constants of $a_{0,0}$ and $a_{\Gamma,\Gamma}$ and the continuity constants of $a_{0,\Gamma}$ and $a_{\Gamma,\Gamma}$. We have

$$\begin{aligned} a_{0,0}(u_{h,0}, u_{h,0}) &= B(u_{h,0}, u_{h,0}) \\ &= k \| \nabla u_{h,0} \|^2 + s \| u_{h,0} \|^2 + (\mathbf{a} \cdot \nabla u_{h,0}, u_{h,0}) \\ &= k \| \nabla u_{h,0} \|^2 + s \| u_{h,0} \|^2 + \int_{\Gamma} \mathbf{n} \cdot \mathbf{a} \frac{1}{2} u_{h,0}^2 \\ &\geq k \| \nabla u_{h,0} \|^2 + s \| u_{h,0} \|^2 - \frac{a}{2} \| u_{h,0} \|_{L^2(\Gamma)}^2 \\ &\geq k \| \nabla u_{h,0} \|^2 + s \| u_{h,0} \|^2 - C \frac{ah}{2} \frac{\delta_2^2}{\delta_1} \| \nabla u_{h,0} \|^2 \quad (\text{by (24)}) \\ &\geq C_k \| \| u_{h,0} \| \|_0^2 \end{aligned}$$

and therefore the coercivity constant of $a_{0,0}$ may be taken as

$$C_{0,0} = C_k$$

On the other hand, we have

$$a_{\Gamma,\Gamma}(u_{h,\Gamma}, u_{h,\Gamma}) = \| u_{h,\Gamma} \|_{L^2(\Gamma)}^2 = \| \| u_{h,\Gamma} \| \|_{\Gamma}^2$$

and hence

$$C_{\Gamma,\Gamma} = 1$$

The continuity constant of $a_{0,\Gamma}$ is obtained from the following bounding process:

$$\begin{aligned} a_{0,\Gamma}(u_{h,\Gamma}, v_{h,0}) &= k (\nabla u_{h,\Gamma}, \nabla v_{h,0}) + \langle \mathbf{a} \cdot \mathbf{n} u_{h,\Gamma}, v_{h,0} \rangle_{\Gamma} \\ &\quad - (u_{h,\Gamma}, \mathbf{a} \cdot \nabla v_{h,0}) + s (u_{h,\Gamma}, v_{h,0}) \\ &\quad - k \langle \partial_n u_{h,\Gamma}, v_{h,0} \rangle_{\Gamma} + k \langle \partial_n v_{h,0}, u_{h,\Gamma} \rangle_{\Gamma} \end{aligned}$$

$$\begin{aligned}
 &\leq k \|\nabla u_{h,\Gamma}\| \|\nabla v_{h,0}\| + a \|u_{h,\Gamma}\|_{L^2(\Gamma)} \|v_{h,0}\|_{L^2(\Gamma)} \\
 &\quad + a \|u_{h,\Gamma}\| \|\nabla v_{h,0}\| + s \|u_{h,\Gamma}\| \|v_{h,0}\| \\
 &\quad + k \|\partial_n u_{h,\Gamma}\|_{L^2(\Gamma)} \|v_{h,0}\|_{L^2(\Gamma)} + k \|\partial_n v_{h,0}\|_{L^2(\Gamma)} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \\
 &\leq k \frac{C}{\delta_1^{1/2} h^{1/2}} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \|\nabla v_{h,0}\| \quad (\text{by (23)}) \\
 &\quad + a \|u_{h,\Gamma}\|_{L^2(\Gamma)} \frac{C h^{1/2} \delta_2}{\delta_1^{1/2}} \|\nabla v_{h,0}\| \quad (\text{by (24)}) \\
 &\quad + a C \delta_1^{1/2} h^{1/2} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \|\nabla v_{h,0}\| \quad (\text{by (22)}) \\
 &\quad + s C \delta_1^{1/2} h^{1/2} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \|v_{h,0}\| \quad (\text{by (22)}) \\
 &\quad + k \frac{C}{\delta_1 h} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \frac{\delta_2 h^{1/2}}{\delta_1^{1/2}} \|\nabla v_{h,0}\| \quad (\text{by (23)–(25)}) \\
 &\quad + k \frac{C}{\delta_1^{1/2} h^{1/2}} \|\nabla v_{h,0}\| \|u_{h,\Gamma}\|_{L^2(\Gamma)} \quad (\text{by (25)})
 \end{aligned}$$

This inequality can be written as

$$a_{0,\Gamma}(u_{h,\Gamma}, v_{h,0}) \leq C K (k \|\nabla v_{h,0}\|^2 + s \|v_{h,0}\|^2)^{1/2} \|u_{h,\Gamma}\|_{L^2(\Gamma)} \tag{27}$$

with

$$K := \frac{k^{1/2}}{\delta_1^{1/2} h^{1/2}} + \frac{ah\delta_2}{k^{1/2} h^{1/2} \delta_1^{1/2}} + \frac{ah\delta_1^{1/2}}{k^{1/2} h^{1/2}} + \delta_1^{1/2} h^{1/2} s^{1/2} + \frac{k^{1/2} \delta_2}{\delta_1^{3/2} h^{1/2}} + \frac{k^{1/2}}{\delta_1^{1/2} h^{1/2}}$$

Using (26) and the fact that $0 < \delta_1, \delta_2 < 1$, from (27) we see that we may take the continuity constant of $a_{0,\Gamma}$ as

$$C_{0,\Gamma} = C \frac{k^{1/2}}{h^{1/2} \delta_1^{3/2}} \left(1 + \frac{s^{1/2} h}{k^{1/2}} \right)$$

The bound for $a_{\Gamma,0}$ is easily obtained using (24)

$$\begin{aligned}
 a_{\Gamma,0}(u_{h,0}, v_{h,\Gamma}) &\leq \|u_{h,0}\|_{L^2(\Gamma)} \|v_{h,\Gamma}\|_{L^2(\Gamma)} \\
 &\leq \frac{C \delta_2 h^{1/2}}{\delta_1^{1/2}} \|v_{h,\Gamma}\|_{L^2(\Gamma)} \|\nabla u_{h,0}\|
 \end{aligned}$$

from where

$$C_{\Gamma,0} = C \frac{\delta_2 h^{1/2}}{\delta_1^{1/2} k^{1/2}}$$

We are now in a position to check condition (17) in our case, which reads

$$C_{0,\Gamma}C_{\Gamma,0} = C \frac{\delta_2}{\delta_1^2} \left(1 + \frac{s^{1/2}h}{k^{1/2}} \right) < C_{0,0}C_{\Gamma,\Gamma} = C_k \quad (28)$$

This inequality is satisfied *provided* δ_2 is small enough, that is to say, Γ is sufficiently close to $\partial\Omega_h$. This is the result we wanted to prove, and which allows us to guarantee that problem (12)–(13) is well posed in this situation.

In passing, condition (28) allows us to observe how stability deteriorates in terms of δ_1 , and also how the rate between reaction and diffusion effects, measured by sh^2/k , affects stability.

3.3. Implementation aspects

The purpose of this subsection is to express in matrix form problem (12)–(13) and to discuss some implementation aspects.

Suppose that the unknown u_h is interpolated as

$$\begin{aligned} u_h(\mathbf{x}) &= \sum_{a=1}^{n_{\text{in}}} I_{\text{in}}^a(\mathbf{x})U_{\text{in}}^a + \sum_{b=1}^{n_{\text{out}}} I_{\text{out}}^b(\mathbf{x})U_{\text{out}}^b \\ &= \mathbf{I}_{\text{in}}(\mathbf{x})\mathbf{U}_{\text{in}} + \mathbf{I}_{\text{out}}(\mathbf{x})\mathbf{U}_{\text{out}} \end{aligned}$$

where $I_{\text{in}}^a(\mathbf{x})$ and $I_{\text{out}}^b(\mathbf{x})$ are the standard interpolation functions, n_{in} is the number of nodes in Ω_{in} (including layer L_0) and n_{out} the number of nodes in layer L_{-1} (see Figure 1).

The objective is to compute \mathbf{U}_{out} . As it has been shown, (13) is equivalent to the minimization problem (14), that is to say, \mathbf{U}_{out} can be computed by minimizing the functional

$$J_2(\mathbf{U}_{\text{in}}, \mathbf{U}_{\text{out}}) = \int_{\Gamma} (u_h(\mathbf{x}) - \bar{u}(\mathbf{x}))^2 = \int_{\Gamma} (\mathbf{I}_{\text{in}}(\mathbf{x})\mathbf{U}_{\text{in}} + \mathbf{I}_{\text{out}}(\mathbf{x})\mathbf{U}_{\text{out}} - \bar{u}(\mathbf{x}))^2$$

Obviously, other options would be possible. In the case we consider

$$\frac{\partial J_2}{\partial \mathbf{U}_{\text{out}}} = 0 \Rightarrow \mathbf{M}_{\Gamma}\mathbf{U}_{\text{out}} = \mathbf{f}_{\Gamma} - \mathbf{N}_{\Gamma}\mathbf{U}_{\text{in}} \quad (29)$$

where

$$\mathbf{M}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x})\mathbf{I}_{\text{out}}(\mathbf{x}), \quad \mathbf{f}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x})\bar{u}(\mathbf{x}), \quad \mathbf{N}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x})\mathbf{I}_{\text{in}}(\mathbf{x})$$

Suppose the matrix form of (12) is

$$\mathbf{K}_{\text{in},\text{in}}\mathbf{U}_{\text{in}} + \mathbf{K}_{\text{in},\text{out}}\mathbf{U}_{\text{out}} = \mathbf{F}_{\text{in}} \quad (30)$$

The domain integrals in matrices $\mathbf{K}_{\text{in},\text{in}}$ and $\mathbf{K}_{\text{in},\text{out}}$ extend only over Ω_{in} . The nodal values \mathbf{U}_{out} are merely used as degrees of freedom to interpolate u_h in the subdomain Ω_{in} . Inserting (29) into (30) results in

$$(\mathbf{K}_{\text{in},\text{in}} - \mathbf{K}_{\text{in},\text{out}}\mathbf{M}_{\Gamma}^{-1}\mathbf{N}_{\Gamma})\mathbf{U}_{\text{in}} = \mathbf{F}_{\text{in}} - \mathbf{K}_{\text{in},\text{out}}\mathbf{M}_{\Gamma}^{-1}\mathbf{f}_{\Gamma} \quad (31)$$

This would be the system to solve. However, since matrix \mathbf{M}_Γ is not diagonal, this option is not feasible unless implemented in an iterative scheme, for example of the form

$$\mathbf{K}_{\text{in,in}}\mathbf{U}_{\text{in}}^k = \mathbf{F}_{\text{in}} - \mathbf{K}_{\text{in,out}}\mathbf{U}_{\text{out}}^{k-1} \tag{32}$$

$$\mathbf{M}_\Gamma\mathbf{U}_{\text{out}}^k = \mathbf{f}_\Gamma - \mathbf{N}_\Gamma\mathbf{U}_{\text{in}}^k \tag{33}$$

where k is the iteration counter.

The most natural option is to solve problem (12)–(13), whose matrix counterpart is (30) and (29), in a coupled way:

$$\begin{bmatrix} \mathbf{K}_{\text{in,in}} & \mathbf{K}_{\text{in,out}} \\ \mathbf{N}_\Gamma & \mathbf{M}_\Gamma \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\text{in}} \\ \mathbf{U}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\text{in}} \\ \mathbf{f}_\Gamma \end{bmatrix} \tag{34}$$

It is important to note that this implementation maintains the connectivity of the mesh of Ω_h , that is to say, the mesh of Ω_{in} extended with the nodes of Ω_{out} corresponding to elements cut by Γ .

Even though system (34) does not offer particular implementation problems, it could be interesting to consider the possibility to obtain an approximation of the form (29) for \mathbf{U}_{out} but *replacing \mathbf{M}_Γ by a diagonal matrix*. The practical reason for this need is clear. For example, in a fluid–structure interaction problem, in order not to duplicate degrees of freedom only nodal values *interior* to the fluid and the solid can be used when solving the corresponding problem.

Let $\mathbf{x}_{\text{out}}^b$ be a node on Ω_{out} corresponding to an element cut by Γ . Consider the edges emanating from $\mathbf{x}_{\text{out}}^b$ cut by Γ , and let Γ_{out}^b be the path (surface in 3D) formed by the intersection of these edges with Γ . These intersections are denoted by \mathbf{x}_Γ with a superscript. In the case of Figure 1, we would have that

- Γ_{out}^1 is the path formed by $\mathbf{x}_\Gamma^1 - \mathbf{x}_\Gamma^2$
- Γ_{out}^2 is the path formed by $\mathbf{x}_\Gamma^3 - \mathbf{x}_\Gamma^4$
- Γ_{out}^3 is just \mathbf{x}_Γ^5
- Γ_{out}^4 is the path formed by $\mathbf{x}_\Gamma^6 - \mathbf{x}_\Gamma^7$
- Γ_{out}^5 is the path formed by $\mathbf{x}_\Gamma^8 - \mathbf{x}_\Gamma^9 - \mathbf{x}_\Gamma^{10} - \mathbf{x}_\Gamma^{11}$

When the path is just a point we can compute U_{out}^b by imposing the boundary condition at that point. In the rest of cases, on each path we have that

$$u_h(\mathbf{x})|_{\Gamma_{\text{out}}^b} = I_{\text{out}}^b(\mathbf{x})U_{\text{out}}^b + \mathbf{I}_{\text{in}}(\mathbf{x})\mathbf{U}_{\text{in}}$$

The idea now is to impose that

$$\frac{\partial}{\partial U_{\text{out}}^b} \int_{\Gamma_{\text{out}}^b} (u_h(\mathbf{x}) - \bar{u}(\mathbf{x}))^2 = 0$$

which yields the scalar equation

$$\left(\int_{\Gamma_{\text{out}}^b} I_{\text{out}}^b(\mathbf{x}) I_{\text{out}}^b(\mathbf{x}) \right) U_{\text{out}}^b = \int_{\Gamma_{\text{out}}^b} I_{\text{out}}^b(\mathbf{x}) \bar{u}(\mathbf{x}) - \int_{\Gamma_{\text{out}}^b} I_{\text{out}}^b(\mathbf{x}) \mathbf{I}_{\text{in}}(\mathbf{x}) \mathbf{U}_{\text{in}}$$

and we can proceed as above, now with a diagonal approximation to \mathbf{M}_{Γ} .

Considering again the situation in Figure 1, it can be seen that with the approximation described we could easily implement (31) *if the connectivities were not modified by the approximate imposition of boundary conditions*. In the case of paths of one or two nodes, that is the case, and (31) could be constructed by *trivial modifications of the element matrices*. However, the situation becomes more involved because of the path formed by $\mathbf{x}_{\Gamma}^8 - \mathbf{x}_{\Gamma}^9 - \mathbf{x}_{\Gamma}^{10} - \mathbf{x}_{\Gamma}^{11}$. The minimization proposed would lead to the coupling of nodes \mathbf{x}_{in}^4 , \mathbf{x}_{in}^5 , \mathbf{x}_{in}^6 and \mathbf{x}_{in}^7 in layer L_0 .

A possibility to avoid the complication described would be *to consider only elemental paths*. In the case of Figure 1 that would mean to consider only paths of two nodes. Possible ways to choose this path are

- The longest among the two-node subpaths.
- The closest to the geometric center of the global path.

The first option has been used in a numerical example of Section 5.

4. SECOND APPROACH: USING INTERNAL DEGREES OF FREEDOM

The method described in the previous section works very well *if Γ is not too close to $\partial\Omega_{\text{in}}$* . When this happens, the method becomes unstable and remedies have to be devised. Let us mention, however, that this instability is *not* particularly strong. In numerical experiments it has manifested as a difficulty for convergence in non-linear problems (the Navier–Stokes equations in our case) and local spurious peaks close to boundaries for the values of \mathbf{U}_{out} with small influence on the values of \mathbf{U}_{in} . In the case of moving boundaries this closeness of Γ to $\partial\Omega_{\text{in}}$ must be accounted for. A possibility is to use the method described next, although a simpler option could be to change the position of the mesh nodes in an arbitrary Lagrangian–Eulerian framework (see [2] for further details).

4.1. Description of the method

The idea of the method described in this section is *to impose the satisfaction of the differential equation in the nodes interior to Ω_{in} , and to use the nodes of $\partial\Omega_{\text{in}}$ to prescribe the boundary conditions on Γ* . Let us elaborate this idea.

Let us consider again (5), which is the weak form of the differential equation to be solved tested with $v_{h,0}$. The space $V_{h,0}$ where this function belongs may be split as $V_{h,0} = V_{h,1} \oplus V_{h,00}$, where $V_{h,1}$ is the subspace of $V_{h,0}$ of functions vanishing on $\partial\Omega_{\text{in}}$ (at nodes of layer L_0 in Figure 1) and $V_{h,00}$ the complement, that is, the subspace of functions that are zero at the *interior* nodes of Ω_{in} . According to this splitting, we may split the unknown as $u_{h,0} = u_{h,1} + u_{h,00}$ and the test functions as $v_{h,0} = v_{h,1} + v_{h,00}$.

Equation (5) can be split as

$$B(u_{h,1}, v_{h,1}) + B(u_{h,00}, v_{h,1}) = \langle f, v_{h,1} \rangle_{\Omega} \quad (35)$$

$$B(u_{h,1}, v_{h,00}) + B(u_{h,00}, v_{h,00}) - k \langle \partial_n u_{h,00}, v_{h,00} \rangle_{\Gamma} - k \langle \partial_n u_{h,\Gamma}, v_{h,00} \rangle_{\Gamma} = \langle f, v_{h,00} \rangle_{\Omega} \quad (36)$$

Recall that integrals are performed over Ω , although the integrals in (35) are extended only over Ω_{in} because this is the support of $v_{h,1}$. The idea now is to keep (35) and to replace (36) by an approximate prescription of the boundary conditions. In order to use only degrees of freedom of nodes in Ω_{in} , let E be the *extrapolation operator* of functions defined on the elements with an edge in 2D or face in 3D on $\partial\Omega_{in}$ to $\Omega_{\Gamma,in}$. The boundary conditions will be approximately imposed by minimizing the functional $J'_2(u_{h,1}, u_{h,00}) = \|Eu_{h,1} + Eu_{h,00} - \bar{u}\|_{L^2(\Gamma)}^2$, that is, by imposing that

$$\delta_{(0, v_{h,00})} J'_2(u_{h,1}, u_{h,00}) = 0 \quad (37)$$

Equations (35) and (37) form the system of equations of the method we propose, which reads: find $u_{h,1} \in V_{h,1}$ and $u_{h,00} \in V_{h,00}$ such that

$$B(u_{h,1}, v_{h,1}) + B(u_{h,00}, v_{h,1}) = \langle f, v_{h,1} \rangle_{\Omega} \quad (38)$$

$$\langle Eu_{h,1}, Ev_{h,00} \rangle_{\Gamma} + \langle Eu_{h,00}, Ev_{h,00} \rangle_{\Gamma} = \langle \bar{u}, Ev_{h,00} \rangle_{\Gamma} \quad (39)$$

for all $v_{h,1} \in V_{h,1}$ and $v_{h,00} \in V_{h,00}$, where now functions in this last space are defined only on Ω_{in} and extrapolated to $\Omega_{\Gamma,in}$.

The description of the method is complete up to the definition of the extrapolation operator. In fact, the obvious choice is to extend the local polynomial expansion within the elements with an edge in 2D or face in 3D on $\partial\Omega_{in}$ to $\Omega_{\Gamma,in}$. Thus, what needs to be defined is only the domain of the extrapolation. The option we use is described in the following subsection.

A comparison between methods (12) and (13) and (38) and (39) in a 1-D case using linear elements is shown in Figure 2. In this case it is possible to satisfy exactly the boundary condition $u_h = \bar{u}$.

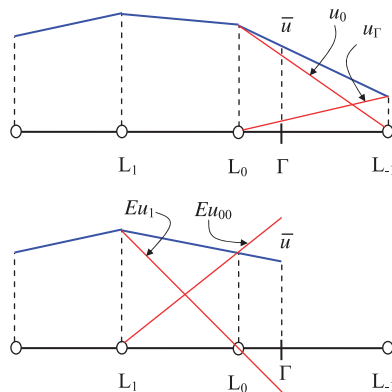


Figure 2. Comparison between methods (12) and (13) (top) and (38) and (39) (bottom) in a 1D case. The dark line denotes the solution computed in both cases.

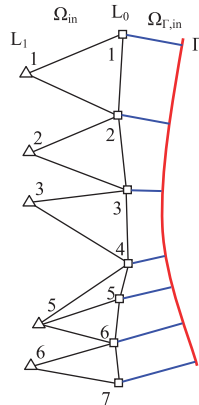


Figure 3. Domain of extrapolation in a 2D example.

Comparing the method proposed in this section with (12) and (13), some remarks need to be made:

- No boundary integrals have to be computed in (38). This is a clear advantage over (12).
- The instability detected for the first method when Γ approaches $\partial\Omega_{in}$ does not appear in this second modification. In fact, the solution is exact when $\Gamma = \partial\Omega_{in}$ (if \bar{u} is a finite element function).
- From the numerical experiments to be presented in Section 5 it is concluded that method (12) and (13) is more accurate than method (38) and (39). However, they have the same order of convergence (two when using linear elements).

4.2. Implementation aspects

The first point to consider is the extrapolation region of the operator E . There are several possibilities, but the one we have found most accurate is the following. Let K be an element with an edge (in 2D) or face (in 3D) F on $\partial\Omega_{in}$. Let K_Γ be the cylinder obtained from projecting F onto Γ in an orthogonal way. Then, E is defined as the extension from functions defined on K to functions defined on $K \cup K_\Gamma$. The extrapolation regions obtained this way in 2D using triangular elements are shown in Figure 3.

Suppose now that in Ω_{in} the unknown u_h is interpolated as

$$u_h(\mathbf{x}) = \sum_{a=1}^{n_1} I_1^a(\mathbf{x})U_1^a + \sum_{b=1}^{n_{00}} I_{00}^b(\mathbf{x})U_{00}^b = \mathbf{I}_1(\mathbf{x})\mathbf{U}_1 + \mathbf{I}_{00}(\mathbf{x})\mathbf{U}_{00}$$

where $I_1^a(\mathbf{x})$ and $I_{00}^b(\mathbf{x})$ are the standard interpolation functions, n_1 is the number of nodes interior to Ω_{in} (up to layer L_1) and n_{00} the number of nodes in layer L_0 (see Figure 3).

The objective is to compute \mathbf{U}_{00} . Equation (39) is equivalent to the minimization problem (37), that is to say, \mathbf{U}_{00} can be computed by minimizing the functional

$$J_2'(\mathbf{U}_1, \mathbf{U}_{00}) = \int_{\Gamma} (Eu_h(\mathbf{x}) - \bar{u}(\mathbf{x}))^2 = \int_{\Gamma} (E\mathbf{I}_1(\mathbf{x})\mathbf{U}_1 + E\mathbf{I}_{00}(\mathbf{x})\mathbf{U}_{00} - \bar{u}(\mathbf{x}))^2$$

which leads to

$$\frac{\partial J'_2}{\partial \mathbf{U}_{00}} = 0 \Rightarrow \mathbf{M}_{00} \mathbf{U}_{00} = \mathbf{f}_{00} - \mathbf{N}_{00} \mathbf{U}_1 \quad (40)$$

where

$$\mathbf{M}_{00} = \int_{\Gamma} E \mathbf{I}_{00}^t(\mathbf{x}) E \mathbf{I}_{00}(\mathbf{x}), \quad \mathbf{f}_{00} = \int_{\Gamma} E \mathbf{I}_{00}^t(\mathbf{x}) \bar{u}(\mathbf{x}), \quad \mathbf{N}_{00} = \int_{\Gamma} E \mathbf{I}_{00}^t(\mathbf{x}) E \mathbf{I}_1(\mathbf{x})$$

Suppose the matrix form of (38) is

$$\mathbf{K}_{1,1} \mathbf{U}_1 + \mathbf{K}_{1,00} \mathbf{U}_{00} = \mathbf{F}_1$$

Combining this with (40) it turns out that the final system to be solved is

$$\begin{bmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,00} \\ \mathbf{N}_{00} & \mathbf{M}_{00} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_{00} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{f}_{00} \end{bmatrix} \quad (41)$$

4.3. Blending

Let us write problem (34) of the previous section as

$$\begin{bmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,00} & \mathbf{0} \\ \mathbf{K}_{00,1} & \mathbf{K}_{00,00} & \mathbf{K}_{00,\text{out}} \\ \mathbf{0} & \mathbf{N}_{\Gamma,00} & \mathbf{M}_{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_{00} \\ \mathbf{U}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_{00} \\ \mathbf{f}_{\Gamma} \end{bmatrix} \quad (42)$$

where the splitting of the matrices corresponds to the splitting of \mathbf{U}_{in} into \mathbf{U}_1 and \mathbf{U}_{00} .

Problem (41) is obtained by considering the degrees of freedom of *all* nodes in layer L_0 as parameters to prescribe the boundary conditions, but of course the last equation in (42) can be kept, in which case the system to be solved is

$$\begin{bmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,00} & \mathbf{0} \\ \mathbf{N}_{00} & \mathbf{M}_{00} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{\Gamma,00} & \mathbf{M}_{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_{00} \\ \mathbf{U}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{f}_{00} \\ \mathbf{f}_{\Gamma} \end{bmatrix} \quad (43)$$

Clearly, \mathbf{U}_{out} depends on \mathbf{U}_{00} , but not the other way around. If Γ is very close to $\partial\Omega_{\text{in}}$, the coefficients in \mathbf{M}_{Γ} can be very small, but this does not affect the unknowns in the interior of the computational domain and, in fact, \mathbf{M}_{Γ} can be replaced by any matrix without altering \mathbf{U}_1 and \mathbf{U}_{00} .

As it has been mentioned and as it will be shown in Section 5, method (42) is more accurate than method (43). In order to use (42) in all situations except when instability problems may appear, we have implemented a blending of methods (42) and (43). The idea is simple. When a node in layer L_0 is detected to be very close to Γ , its degree of freedom is used to prescribe the boundary conditions, that is to say, the row in the equation for \mathbf{U}_{00} in (42) is replaced by the corresponding row in (43). This strategy has proved robust and effective. Since usually only a few equations need to be changed (in our case those for which the distance of a node in L_0 to Γ is less than $0.1h$), the overall accuracy obtained is very close to that of method (42).

5. NUMERICAL EXAMPLES

In this section we present the numerical results obtained with the two approximations of the Dirichlet boundary conditions proposed. As it has been mentioned, we are interested in flow problems, and in particular in situations in which the Galerkin formulation used heretofore may be unstable. This is why we start this section presenting the stabilized formulation used in the numerical examples.

5.1. Stabilized convection–diffusion–reaction and incompressible Navier–Stokes equations

It is well known that when the diffusion coefficient k in (1) is small, the Galerkin method fails and stabilized finite element methods need to be used. It is not our purpose here to explain the roots of the particular method we use (see for example [32]), but only to state it. The bottomline is to replace the bilinear form $B(u_h, v_h)$ and the linear form $\langle f, v_h \rangle_\Omega$ in (4) by $B_{\text{stab}}(u_h, v_h)$ and $\langle f, v_h \rangle_{\text{stab}}$, respectively, given by

$$\begin{aligned} B_{\text{stab}}(u_h, v_h) &= B(u_h, v_h) + \sum_K \tau_K \langle -\mathcal{L}^* v_h, \mathcal{L} u_h \rangle_K \\ &= k(\nabla u_h, \nabla v_h) + (\mathbf{a} \cdot \nabla u_h, v_h) + s(u_h, v_h) \\ &\quad + \sum_K \tau_K \langle k \Delta v_h + \mathbf{a} \cdot \nabla v_h - s v_h, -k \Delta u_h + \mathbf{a} \cdot \nabla u_h + s u_h \rangle_K \end{aligned}$$

and

$$\begin{aligned} \langle f, v_h \rangle_{\text{stab}} &= \langle f, v_h \rangle_\Omega + \sum_K \tau_K \langle -\mathcal{L}^* v_h, f \rangle_K \\ &= \langle f, v_h \rangle_\Omega + \sum_K \tau_K \langle k \Delta v_h + \mathbf{a} \cdot \nabla v_h - s v_h, f \rangle_K \end{aligned}$$

where the so-called stabilization parameter τ_K is given by

$$\tau_K = \left(c_1 \frac{k}{h^2} + c_2 \frac{a}{h} + s \right)^{-1}$$

In the numerical experiments presented below we have taken $c_1=4$, $c_2=2$. The relationship between τ_K and the stabilization parameter of other formulations can be found in [33].

The other problem for which a numerical example is presented below is the incompressible Navier–Stokes equations, which consist in finding a velocity field \mathbf{u} and a pressure p such that

$$\begin{aligned} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}$$

in Ω and for $t > 0$, where \mathbf{f} is the vector of body forces and ν the kinematic viscosity. Appropriate initial and boundary conditions have to be appended to this problem. They are described for the particular example of the flow over a cylinder shown later.

Except for the treatment of the Dirichlet boundary conditions for the velocity, which is similar to the one described in detail for the scalar convection–diffusion–reaction equation, the space-discrete problem we solve is

$$\begin{aligned} & (\partial_t \mathbf{u}_h, \mathbf{v}_h) + \langle \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}_h \rangle_\Omega + v(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) - (p_h, \nabla \cdot \mathbf{u}_h) + (q_h, \nabla \cdot \mathbf{v}_h) \\ & + \sum_K \tau_K \langle v \Delta \mathbf{v}_h + \mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nabla q_h, \partial_t \mathbf{u}_h - v \Delta \mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h \rangle_K \\ & - \langle \mathbf{f}, \mathbf{v}_h \rangle_\Omega - \sum_K \tau_K \langle v \Delta \mathbf{v}_h + \mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nabla q_h, \mathbf{f} \rangle_K = 0 \end{aligned}$$

where \mathbf{v}_h is the velocity test function, q_h the pressure test function and now the stabilization parameter is computed as

$$\tau_K = \left(c_1 \frac{v}{h^2} + c_2 \frac{|\mathbf{u}_h|_K}{h} \right)^{-1}$$

where $|\mathbf{u}_h|_K$ is the mean velocity modulus in element K . Any finite difference scheme can be used to approximate the time derivative $\partial_t \mathbf{u}_h$. In particular, the second-order Crank–Nicolson scheme has been used in the example of Section 5.3.

Details for the motivation of the formulation described and stability and convergence properties can be found in [34]. The most salient property of the formulation is that *equal* velocity–pressure interpolations can be used. In particular, linear velocities and linear pressures have been used in the numerical example of Section 5.3. Note, however, that the pressure interpolation does not affect the approximate imposition of Dirichlet boundary conditions, since these affect only the velocity. Likewise, instabilities of the Galerkin method arising in convection-dominated flows are prevented using the stabilized formulation presented. Let us note, however, that since the use of this formulation changes consistently the forms involved in the problem, the methodology we propose to impose boundary conditions is not affected. In the case of the convection–diffusion–reaction equation, for example, (12) would change, but (13) would remain unaltered.

5.2. Results for the scalar convection–diffusion–reaction equation

In this subsection we illustrate the behavior of the proposed methods for the scalar convection–diffusion–reaction equation. The Poisson, diffusion–reaction and convection–diffusion equations are solved in a domain Ω enclosed in a circle of radius $R < 1$. We choose the *hold-all domain* $B = (-1, 1) \times (-1, 1)$, where a system of Cartesian coordinates (x, y) with its origin at the center of the circle has been adopted. A structured mesh of right-angled linear triangular elements is constructed in B , h being the length of the edges corresponding to the cathetus (see Figure 4).

5.2.1. The Poisson equation. Let us start solving the Poisson equation with $k = 1$, $\mathbf{a} = \mathbf{0}$, $s = 0$, $f = 1$ to check the performance and convergence of the proposed methods. Results are shown in Figure 5 (top and bottom left). No significative difference between the fields u_h obtained with the different methods can be appreciated, even for the coarsest meshes.

The analytical solution for this case is known to be

$$u(x, y) = \frac{1}{4}(R^2 - x^2 - y^2)$$

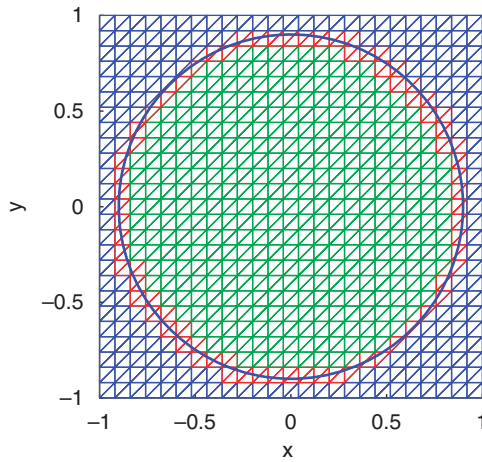


Figure 4. Structured mesh and domains Ω_{in} (light) and Ω_{Γ} (dark).

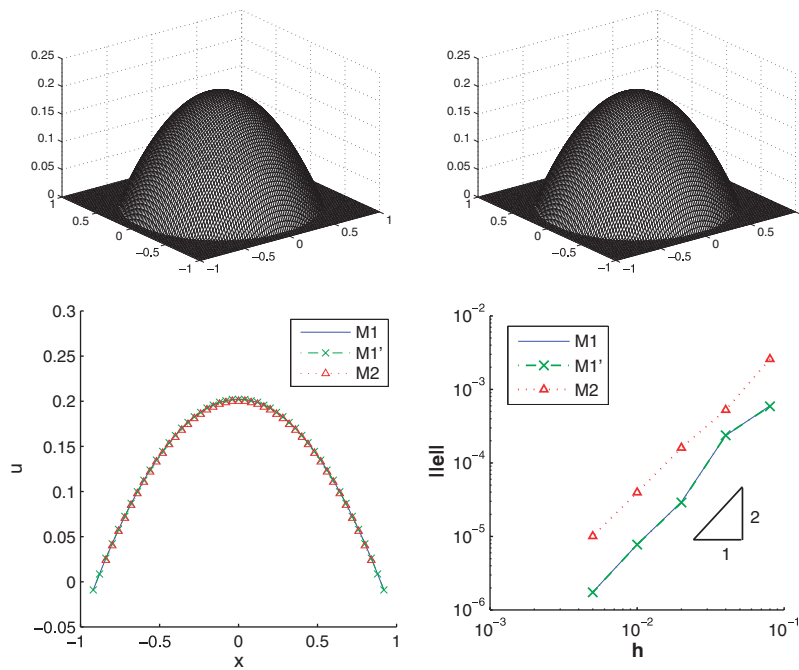


Figure 5. Comparison between the proposed methods. Top left: elevation u_h for the Poisson equation for M1. Top right: same for M2. Bottom left: cut along $y=0$ for the coarsest mesh used, with $h = \frac{2}{25}$. Bottom right: convergence plot in $L^2(\Omega)$ for methods M1, M1' and M2.

Figure 5 (bottom right) shows the errors $\|u - u_h\|_{L^2(\Omega)}$ versus the element size h . As it can be seen, both the first method described in Section 3 (labeled M1 in the following), and the second one described in Section 4 (labeled M2) show quadratic convergence, although the error turns

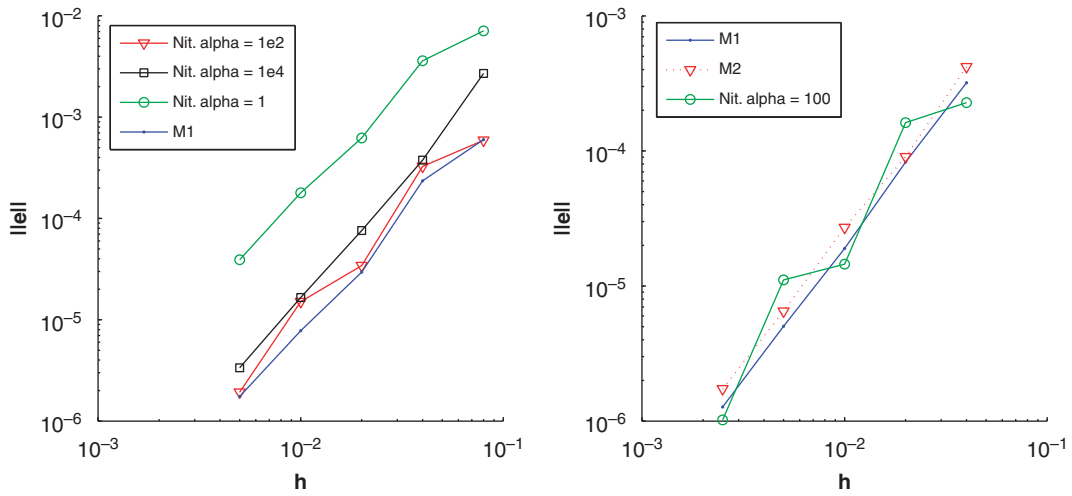


Figure 6. Comparison between M1, M2 and Nitsche’s method. Left: convergence plot in $L^2(\Omega)$ for method M1 and Nitsche’s method with different values of the parameter α . Right: convergence plot in $L^2(\partial\Omega)$ for methods M1, M2 and Nitsche’s method with $\alpha=100$.

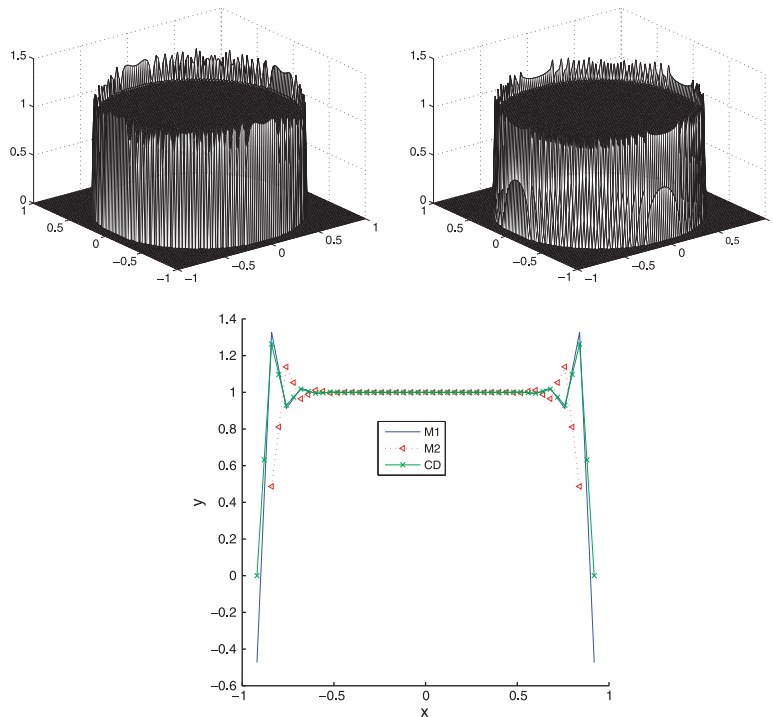


Figure 7. Reaction-dominated case, exact integration. Elevation u_h for M1 (top left) and M2 (top right). Cut along $y=0$ for the coarsest used mesh ($h = \frac{2}{25}$) (bottom).

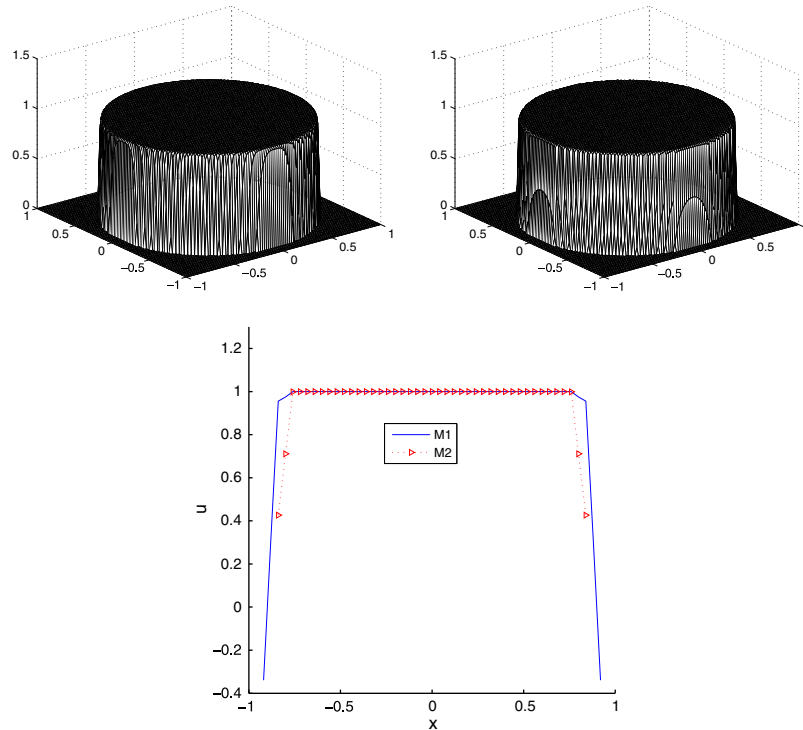


Figure 8. Reaction-dominated case, nodal integration. Elevation u_h for M1 (top left) and M2 (top right). Cut along $y=0$ for the coarsest used mesh ($h = \frac{2}{25}$) (bottom).

out to be smaller for the former. The modified version of M1 (referred to as M1'), which uses a diagonal approximation of matrix \mathbf{M}_Γ computed by considering only the longest elemental paths (see Section 3.3), shows no significant error increment with respect to M1.

In order to compare the performance of the methods proposed with Nitsche's method, in Figure 6 (left) we have also plotted the convergence obtained using this method with three different choices of the parameter α in (4) (taking $k^* = k$), namely, $\alpha = 100$, which is approximately the optimal value found from numerical experiments, $\alpha = 1$ and $\alpha = 10000$. It can be observed that the performance of method M1 is superior to Nitsche's method, even for its optimal case, and that this method is sensitive to the choice of the parameter α . This is aggravated in problems with convection and/or reaction, for which k^* (or, alternatively, α) must be chosen in terms of the advection velocity and the reaction coefficient. In Figure 6 (right) we have plotted convergence in $L^2(\partial\Omega)$, and therefore the error is due only to the imposition of the boundary conditions. Nitsche's method displays a non-monotone behavior due to the way the elements cut the boundary of the domain for different meshes. Again, methods M1 and M2 show a similar behavior when only the errors on the boundary are taken into account.

5.2.2. Reaction–diffusion. When the reactive term s dominates over the diffusive one it is well known that oscillations in the finite element approximated solution u_h appear near the boundary

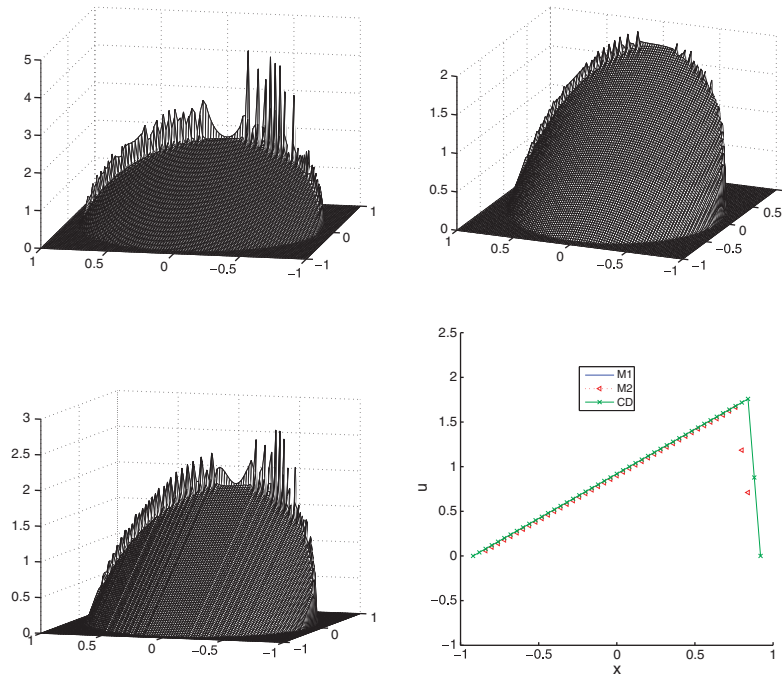


Figure 9. Convection-dominated case. Elevation u_h for M1 (top left), M2 (top right) and CD (bottom left). Cut along $y=0$ for the coarsest used mesh ($h=\frac{2}{25}$) (bottom right).

layer. It is thus convenient to check how do the proposed methods behave in the presence of this *Gibb's phenomenon*. Figure 7 shows u_h for M1, M2 and the local remeshing strategy described in Section 1, labeled CD in Figure 7. These results correspond to the reaction dominated case, where $k=10^{-5}$, $\mathbf{a}=\mathbf{0}$, $s=1$, $f=1$.

Although oscillations remain bounded close to the exact solution u both for M1 and M2, they happen to be greater in the former than in the latter. Nevertheless, when compared with results obtained with CD, oscillations in M1 are practically of the same magnitude as those obtained for the classical method, while the solution for M2 clearly shows a reduction in the amount of oscillation.

If nodal integration is used to compute the contribution of the reactive term to the resulting system of equations, oscillations can be avoided, since the resulting matrix is of *non-negative type*, and thus the *discrete minimum principle* is satisfied, that is to say, for $f \geq 0$ the minimum of the solution is attained at boundary nodes (this principle holds if and only if the discrete maximum principle does, see e.g. [35]). In this case none of the two methods shows any oscillation (see Figure 8), and the only difference between them is due to the fact that M2 uses only the degrees of freedom corresponding to the nodes in the Ω_{in} domain, while M1 incorporates also the nodes corresponding to the Ω_{out} domain (this is also the reason why M1 leads to a better approximation to u_h than M2).

5.2.3. Convection–diffusion. Figure 9 shows the behavior of methods M1 and M2 in the convection-dominated case, where $k=10^{-6}$, $\mathbf{a}=(1,0)$, $s=0$, $f=1$. The stabilized formulation

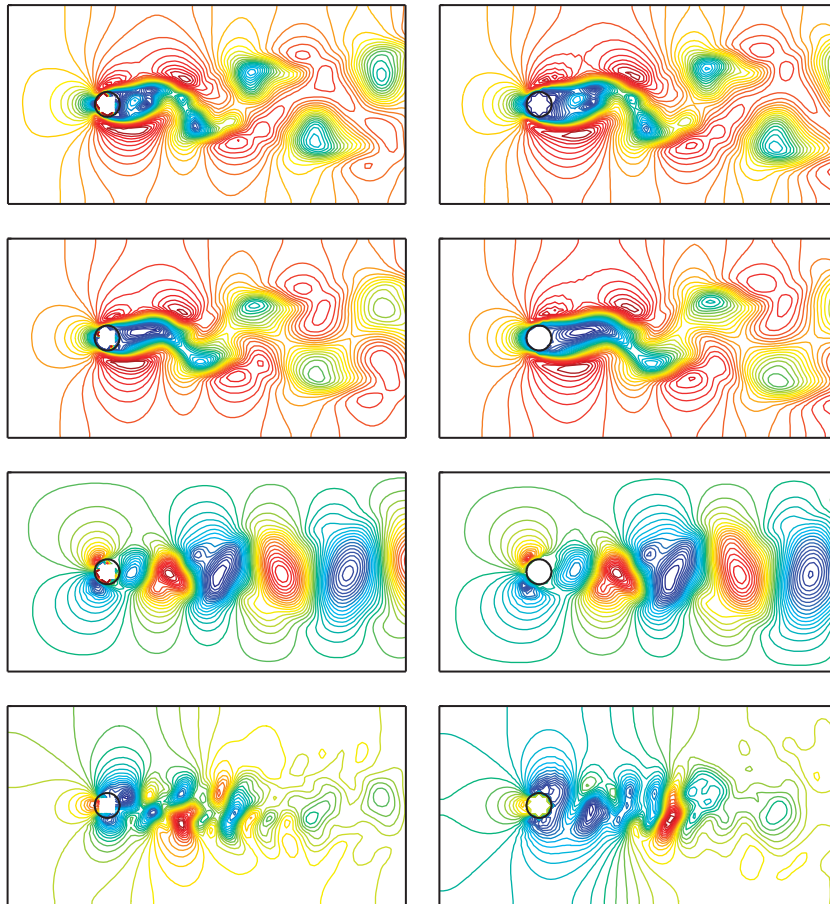


Figure 10. Incompressible Navier–Stokes equations. Solution at $t=200$. Left: method M1, Right: method M2. From the top to the bottom: velocity module, contours of velocity x -component, contours of velocity y -component, pressure contours.

described in Subsection 5.1 has been used. Both methods M1 and M2 perform well, although again oscillations are greater for M1. This time, however, oscillations for M1 are substantially greater than those which appear when applying Dirichlet conditions in boundary fitting meshes CD, with the local remeshing strategy described in Section 1. Again also, M2 shows less oscillations than CD.

Despite the different behavior that both methods show in the boundary layer, the difference between the two methods in the Ω_{in} domain is practically negligible.

The local oscillations appearing in M1, altogether with the fact that the splitting of elements in the Ω_{in} domain can lead to an ill-conditioning of the resulting system of equations when Γ is too close to $\partial\Omega_{\text{in}}$, can prevent convergence in non-linear problems. This is what motivates the *blending strategy* proposed in Section 4.3.

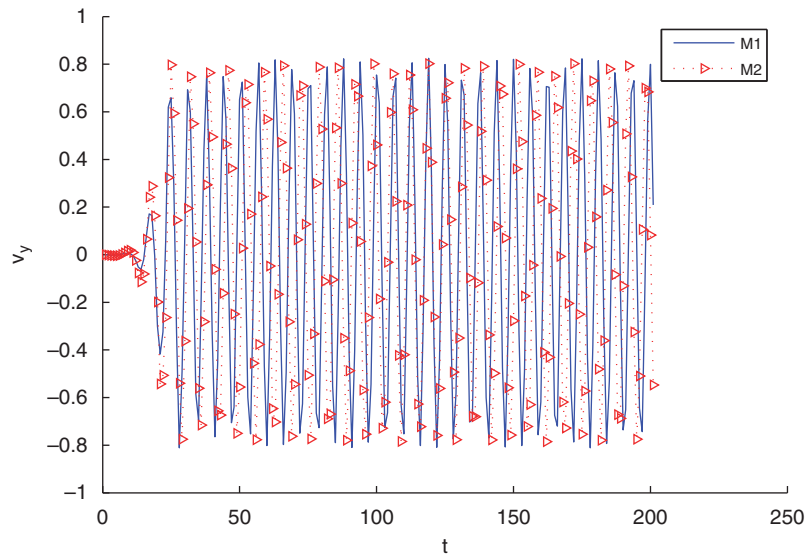


Figure 11. Incompressible Navier–Stokes equations. Evolution of the y -velocity component at point $(10, 4)$ for methods M1 and M2.

5.3. Results for the incompressible Navier–Stokes equations

In this subsection we analyze a numerical example involving the flow past a cylinder. Again the formulation described in Section 5.1 has been used.

The *hold-all domain* is the rectangle $B = [0, 16] \times [0, 8]$, from which a cylinder of diameter $D = 1$ and centered at $(4, 4)$ is extracted. The velocity at $x = 0$ is prescribed to $(10, 0)$, whereas at $y = 0$ and $y = 8$ the y -velocity component is prescribed to 0 and the x -component is left free. The outflow (where both the x - and y -components are free) is $x = 16$. The Reynolds number is 100, based on the cylinder diameter and the prescribed inflow velocity. The finite element mesh employed consists of 10 000 linear triangles. The Crank–Nicolson scheme has been used for the time integration, with a time step size $\delta t = 1$.

Velocity contours and pressure contours at $t = 200$ obtained using methods M1 and M2 are shown in Figure 10. The important issue is to observe that boundary conditions are well approximated both using M1 and M2. The evolution of the y -velocity component at point $(10, 4)$ is shown in Figure 11. It can be observed that both methods yield a similar amplitude, the frequency obtained with method M2 being slightly smaller. The dimensionless period of the oscillations is found to be $T = 6.11$ for method M1 and $T = 6.5$ with method M2. Consistently with the results for the convection–diffusion–reaction equation, method M2 seems to behave always as more dissipative than method M1.

6. CONCLUSIONS

In this paper we have proposed a way to prescribe approximately Dirichlet boundary conditions for IBM. The main idea is to use as degrees of freedom for this imposition those associated to

the nodes adjacent to the boundary of the computational domain. In a first approach, these nodes are taken in the exterior of the domain, but this may yield instabilities (mild and unusual) that can be overcome by using interior nodes and extrapolation. In any case, the degrees of freedom are computed by minimizing the distance of the unknown to the boundary datum in the L^2 norm of the boundary.

The method proposed turns out to be accurate (second order for linear elements) and robust. We have checked its numerical performance in a variety of situations in flow problems, paying particular attention to problems that require stabilization.

From the implementation point of view, the method satisfies the main design condition of using only the degrees of freedom of the mesh of Ω_h . This is particularly important in the case of domains with moving boundaries in which a single fixed mesh is used during the whole calculation, which in fact is the motivation that led us to formulate the method proposed in this paper.

ACKNOWLEDGEMENTS

J. Baiges would like to acknowledge the support received from the *Departament d'Universitats, Recerca i Societat de la Informació* of the *Generalitat de Catalunya* (Catalan Government).

REFERENCES

1. Houzeaux G, Codina R. A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:3343–3377.
2. Codina R, Houzeaux J, Coppola-Owen H, Baiges J. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **8**:1591–1611.
3. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**(2):83–130.
4. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
5. Löhner R, Cezbral JR, Camelli FF, Baum JD, Mestreau EL. Adaptive embedded/immersed unstructured grid techniques. *Archives of Computational Methods in Engineering* 2007; **14**:279–301.
6. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**:252–271.
7. Lai M-C, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705–719.
8. LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994; **31**(4):1019–1044.
9. LeVeque RJ, Li Z. Immersed interface method for incompressible Navier–Stokes equations. *SIAM Journal on Scientific and Statistical Computing* 1997; **18**(3):709–735.
10. Xu S, Wang ZJ. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics* 2006; **216**:454–493.
11. Babuska I. Error bounds for finite element method. *Numerische Mathematik* 1971; **16**:322–333.
12. Dolbow J, Mourad HM, Harari I. A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering* 2007; **69**:772–793.
13. Barbosa HJC, Hughes TJR. The finite element method with Lagrangian multipliers on the boundary: circumventing the Babuška–Brezzi condition. *Computer Methods in Applied Mechanics and Engineering* 1991; **85**:109–128.
14. Ji H, Dolbow JE. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering* 2004; **61**:2508–2535.
15. Glowinski R, Pan TW, Périaux J. A fictitious domain method for Dirichlet problems and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:203–303.
16. Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow. *International Journal for Numerical Methods in Fluids* 1999; **30**:1043–1066.

17. Glowinski R. Finite element methods for incompressible viscous flows. *Numerical Methods for Fluids (Part 3), Handbook of Numerical Analysis*, vol. 9. Elsevier: Amsterdam, 2003.
18. Gilmanov A, Sotiropoulos F. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics* 2005; **207**:457–492.
19. Ferziger JH, Tseng YH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
20. Mohd-Yusof J. Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. *CTR Annual Research Briefs*, Stanford University, NASA Ames, 1997.
21. Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method. submitted for publication.
22. Brenner SC, Scott LR. *The Mathematical Theory of Finite Element Methods*. Springer: Berlin, 1994.
23. Ern A, Guermond J-L. *Theory and Practice of Finite Elements*. Springer: Berlin, 2004.
24. Juntunen M, Stenberg R. Nitsche's method for general boundary conditions. *Research Reports A530*, Helsinki University of Technology, Institute of Mathematics, 2007.
25. Arnold DN. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis* 1982; **19**:742–760.
26. Hansbo P, Larson MG. Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:1895–1908.
27. Becker R, Hansbo P, Stenberg R. A finite element method for domain decomposition with non-matching grids. *Mathematical Modelling and Numerical Analysis* 2003; **37**:209–225.
28. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:5537–5552.
29. Burman E, Fernández MA, Hansbo P. Continuous interior penalty finite element method for Oseen's equations. *SIAM Journal on Numerical Analysis* 2006; **44**:1248–1274.
30. Coppola-Owen H, Codina R. A finite element model for free surface flows on fixed meshes. *International Journal for Numerical Methods in Fluids* 2007; **54**:1151–1171.
31. Durbin P, Majumdar S, Iaccarino G. RANS solvers with adaptive structured boundary non-conforming grids. *Annual Research Briefs*, NASA Ames Research Center/Stanford University Center for Turbulence Research, Stanford, CA, 2001; 353–366.
32. Codina R. Comparison of some finite element methods for solving the diffusion–convection–reaction equation. *Computer Methods in Applied Mechanics and Engineering* 1998; **156**:185–210.
33. Codina R, Zienkiewicz OC. CBS versus GLS stabilization of the incompressible Navier–Stokes equations and the role of the time step as stabilization parameter. *Communications in Numerical Methods in Engineering* 2002; **18**:99–112.
34. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
35. Codina R. A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection–diffusion equation. *Computer Methods in Applied Mechanics and Engineering* 1993; **110**:325–342.