

MOX–Report No. 37/2010

**Numerical approximation of internal discontinuity  
interface problems**

MARCO DISCACCIATI, ALFIO QUARTERONI,  
SAMUEL QUINODOZ

MOX, Dipartimento di Matematica “F. Brioschi”  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it

<http://mox.polimi.it>



# Numerical approximation of internal discontinuity interface problems

Marco Discacciati<sup>1</sup>, Alfio Quarteroni<sup>1,2</sup>, Samuel Quinodoz<sup>1</sup>

<sup>1</sup> MATHICSE, Chair of Modelling and Scientific Computing, Ecole Polytechnique  
Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland.

`marco.discacciati@epfl.ch`

<sup>2</sup> MOX, Politecnico di Milano, P.zza Leonardo da Vinci 32, I-20133 Milano, Italy.

`alfio.quarteroni@epfl.ch`

## Abstract

This work focuses on the finite element discretization of boundary value problems whose solution presents either a discontinuity and/or a discontinuous conormal derivative across an interface inside the computational domain. The interface is characterized via a level-set function. The discontinuities are accounted for using suitable extension operators whose numerical implementation requires a very low computational effort. Numerical results to validate our approach are presented in one, two and three dimensions.

## 1 Introduction

This paper is focused on the numerical approximation of elliptic problems whose solution features discontinuities across interfaces internal to the computational domain. More precisely, we consider a Poisson problem in two disjoint subdomains of the computational domain  $\Omega \subset \mathbb{R}^N$  ( $N = 1, 2, 3$ ) with jump conditions across the interface  $\Gamma$  separating the two subregions.  $\Gamma$  is a point if  $N = 1$ , a line if  $N = 2$  or a surface if  $N = 3$  that is characterized by a level-set function  $\phi : \Omega \rightarrow \mathbb{R}$ . More precisely, we consider an open bounded domain  $\Omega$  to be partitioned into the two non-overlapping subdomains  $\Omega_1 = \{x \in \Omega \mid \phi(x) < 0\}$  and  $\Omega_2 = \Omega \setminus \Omega_1$ . According to the classical level-set method (see, e.g., [9]),  $\phi$  is regarded as the signed distance function to the interface, whence  $\Gamma$  is defined by the equation  $\phi = 0$ .

The mathematical formulation of our problem is as follows. We look for a function  $u$  in  $\Omega$  that satisfies a Poisson problem in each subdomain:

$$-\Delta u_i = f_i \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (1)$$

where  $u_i = u|_{\Omega_i}$ , with the following conditions on the jumps of the trace and of the normal derivatives across  $\Gamma$ :

$$[[u]]_{\Gamma} = u_{2|\Gamma} - u_{1|\Gamma} = g_d, \quad (2)$$

$$\left[ \left[ \frac{\partial u}{\partial n} \right] \right]_{\Gamma} = \nabla u_2 \cdot n_2|_{\Gamma} + \nabla u_1 \cdot n_1|_{\Gamma} = g_n. \quad (3)$$

$g_d$  and  $g_n$  are two assigned functions on  $\Gamma$ , while  $n_1$  and  $n_2$  are the unit normal vectors on  $\Gamma$  directed outwards of  $\Omega_1$  and  $\Omega_2$ , respectively. Notice that  $n_2 = -n_1$  on  $\Gamma$ . For simplicity, we impose homogeneous Dirichlet boundary conditions on the boundary of the domain  $\Omega$ :  $u = 0$  on  $\partial\Omega$ .

Discontinuities in the normal derivative of the solution of a PDE arise e.g. as soon as a force is localized on a part of the computational domain with lower dimension. The most popular example is the surface tension in flow simulations [6], which is a force that applies at the interface between two fluids. In this framework,  $\Gamma$  is a free surface, that is its location is a further unknown of the problem at hand. It can also happen that the solution itself is discontinuous, for example with phase transition when considering entropy [2] or with incompressible flames [5].

A possible strategy to correctly approximate this kind of problems is to build a mesh which captures the interface. This solution however can be not always practicable, as for a time dependent problem conforming meshes (that is meshes that perfectly match on the interface) have to be rebuilt at each time step, resulting in too expensive schemes in term of computational cost. The mesh could also be cut by the interface and only locally rebuilt, however this could lead to highly deformed cells. For these reasons, methods that do not require the reconstruction of a new mesh are preferable. For the same reasons, a good method should not require mesh refinement near the interface, neither require the computation of quantities (such as integrals) on the interface itself.

The goal of this paper is to present a new method, called SESIC (for Simplified Exact Subgrid Interface Correction), to take into account these jump conditions in a finite element framework. This method has been inspired by the ESIC method developed in [4]: we have improved the construction of the liftings and changed the weak formulation so that the efficiency of the ESIC is kept while improving its mathematical interpretation and its effective implementation. More precisely, the paper has the following content. In section 2, we study the weak formulation of the internal discontinuity problem (1)-(3) and we introduce at the continuous level suitable lifting operators to account for the jumps across the interface. In section 3, we present the finite element approximation of the problem and we discuss several numerical issues of our methodology comparing it to the one developed in [4]. Finally, section 4 presents the numerical results that we have obtained on different test cases.

Even though this paper addresses the model problem (1) only, the numerical methods here considered can be easily extended to address the more complex situations mentioned above, e.g. the problem of numerical approximation of

free surface in multiphase flows. A preliminary example is provided in section 5. This issue will make the object of a future paper.

## 2 Weak formulation for the internal discontinuity interface problem

To derive a weak formulation of (1) - (3), we introduce two suitable liftings (or extensions)  $R_i g_d$  ( $i = 1, 2$ ) of  $g_d$  in  $\Omega_i$  so that the jump of  $R_i g_d$  is  $g_d$  on  $\Gamma$ :  $R_i g_d \in H^1_{\partial\Omega_i \setminus \Gamma}(\Omega_i)$  ( $i = 1, 2$ ) such that  $[[R_i g_d]]_\Gamma = g_d$ . Then, we consider the splitting

$$u_i = \bar{u}_i + R_i g_d \quad \text{in } \Omega_i. \quad (4)$$

We denote  $\bar{u} : \Omega \rightarrow \mathbb{R}$  such that  $\bar{u}_i = \bar{u}|_{\Omega_i}$ . The function  $\bar{u}$  belongs to  $H^1_0(\Omega)$ . We consider a global test function  $v \in H^1_0(\Omega)$  and its restrictions  $v_i$  on  $\Omega_i$ . Then, on each domain, starting from (1), integrating by parts and exploiting the homogeneous Dirichlet boundary conditions on  $\partial\Omega_i \cap \partial\Omega$ , we obtain

$$\int_{\Omega_i} \nabla u_i \cdot \nabla v_i - \int_\Gamma \frac{\partial u_i}{\partial n_i} v_i = \int_{\Omega_i} f_i v_i. \quad (5)$$

Summing up the contributions of each subdomain

$$\sum_{i=1}^2 \int_{\Omega_i} \nabla u_i \cdot \nabla v_i - \int_\Gamma \left( \frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2} \right) v_i = \sum_{i=1}^2 \int_{\Omega_i} f_i v_i \quad (6)$$

and imposing the jump condition on the normal derivative (3) in a natural way, we obtain

$$\sum_{i=1}^2 \int_{\Omega_i} \nabla u_i \cdot \nabla v_i - \int_\Gamma g_n v_i = \sum_{i=1}^2 \int_{\Omega_i} f_i v_i. \quad (7)$$

Finally, using the decomposition (4), we obtain the weak form of problem (1)-(3): find  $\bar{u} \in H^1_0(\Omega)$  such that  $\forall v \in H^1_0(\Omega)$ :

$$\int_\Omega \nabla \bar{u} \cdot \nabla v = \sum_{i=1}^2 \int_{\Omega_i} f_i v_i - \sum_{i=1}^2 \int_{\Omega_i} \nabla R_i g_d \cdot \nabla v_i + \int_\Gamma g_n v. \quad (8)$$

Another possible weak formulation of (1)-(3) is proposed by Huh and Sethian [4], where an additional lifting is considered for the function  $g_n$ . More precisely, they define a function  $S_i g_n \in H^1_{\partial\Omega_i \setminus \Gamma}(\Omega_i)$ ,  $i = 1, 2$  such that  $[[\frac{\partial S_i g_n}{\partial n}]]_\Gamma = g_n$  and, instead of (4), they consider the splitting:

$$u_i = \hat{u}_i + R_i g_d + S_i g_n. \quad (9)$$

The two liftings  $R_i g_d$  and  $S_i g_n$  ideally satisfy the following constraints:

$$[[R_i g_d]]_\Gamma = g_d \quad \left[ \left[ \frac{\partial R_i g_d}{\partial n} \right] \right]_\Gamma = 0, \quad (10)$$

$$\llbracket S_i g_n \rrbracket_\Gamma = 0 \quad \left[ \left[ \frac{\partial S_i g_n}{\partial n} \right] \right]_\Gamma = g_n , \quad (11)$$

so that they take into account independently the jump of the functions and that of the fluxes.

Using the approach by Huh and Sethian, one obtains the weak form: find  $\hat{u} \in H_0^1(\Omega)$  such that,  $\forall v \in H_0^1(\Omega)$ ,

$$\int_\Omega \nabla \hat{u} \cdot \nabla v = \sum_{i=1}^2 \int_{\Omega_i} f_i v_i - \sum_{i=1}^2 \int_{\Omega_i} \nabla (R_i g_d + S_i g_n) \cdot \nabla v_i + \int_\Gamma g_n v . \quad (12)$$

Note that  $\hat{u}$  is such that  $\hat{u}|_{\Omega_i} = \hat{u}_i$  for  $i = 1, 2$ .

We can remark that the bilinear form associated to both methods (8) and (12) is the classical Dirichlet formulation of the Poisson problem in  $H_0^1(\Omega)$  (without internal discontinuity interface). This allows proving the well-posedness of the problem in a direct way by the Lax-Milgram lemma [7].

Both formulations (12) and (8) are equivalent from the mathematical point of view. However, their numerical approximation gives rise to different numerical solutions. We will discuss this issue in Sect. 4.1, while we focus now on the construction of the lifting operators  $R_i$  and  $S_i$ .

## 2.1 The continuous lifting operators

To construct the liftings, we assume that there exist two continuous scalar functions  $\bar{g}_d$  and  $\bar{g}_n$  in  $\Omega$  such that  $g_d = \bar{g}_d|_\Gamma$  and  $g_n = \bar{g}_n|_\Gamma$ . This assumption is more or less strong depending on the way the data  $g_d$  and  $g_n$  are provided. If, as in the cases that we will consider, they are given as functions on the whole domain  $\Omega$  or as a finite element function, this assumption is fulfilled. If they are given in another way, then we have to rely on an extension procedure, depending again on the information available.

Thanks to the extensions  $\bar{g}_d$  and  $\bar{g}_n$ , we are now able to define liftings that satisfy conditions (10) and (11) exactly at the continuous level. For the sake of simplicity, we start with the lifting for  $g_n$  which accounts for the jump in the normal derivative. Consider the function

$$S g_n = H(\phi) \phi \bar{g}_n \quad \text{in } \Omega, \quad (13)$$

where  $H(\phi)$  is the Heaviside function:

$$H(\phi)(x) = \begin{cases} 1 & \text{if } \phi(x) \geq 0 \\ 0 & \text{if } \phi(x) < 0 , \end{cases}$$

whence

$$S g_n(x) = \begin{cases} \phi(x) \bar{g}_n(x) & \text{if } \phi(x) \geq 0 \\ 0 & \text{if } \phi(x) < 0 . \end{cases} \quad (14)$$

Note that  $S g_n$  is continuous across  $\Gamma$  (defined as the 0-level set of  $\phi$ ), that is  $\llbracket S g_n \rrbracket_\Gamma = 0$ . On the other hand,

$$\left[ \left[ \frac{\partial S g_n}{\partial n} \right] \right]_\Gamma = \frac{\partial(\phi \bar{g}_n)}{\partial n} \Big|_\Gamma = \frac{\partial \phi}{\partial n} \Big|_\Gamma \bar{g}_n \Big|_\Gamma + \frac{\partial \bar{g}_n}{\partial n} \Big|_\Gamma \phi \Big|_\Gamma = g_n , \quad (15)$$

thus  $Sg_n$  is a suitable lifting for  $g_n$  in the sense that the conditions (11) are satisfied.

We also need a lifting  $Rg_d$  for the function  $g_d$  which is discontinuous across  $\Gamma$  but featuring a continuous normal derivative.

To this aim we define  $Rg_d$  as:

$$Rg_d = H(\phi)(\bar{g}_d - \nabla \bar{g}_d \cdot \nabla \phi \phi) \quad (16)$$

which can be expressed explicitly as

$$Rg_d(x) = \begin{cases} \bar{g}_d(x) - \nabla \bar{g}_d(x) \cdot \nabla \phi(x) \phi(x) & \text{if } \phi(x) \geq 0 \\ 0 & \text{if } \phi(x) < 0. \end{cases} \quad (17)$$

We can see that  $\llbracket Rg_d \rrbracket_\Gamma = g_d$ , whereas by a direct computation

$$\begin{aligned} \left[ \left[ \frac{\partial Rg_d}{\partial n} \right] \right]_\Gamma &= \frac{\partial(\bar{g}_d - \nabla \bar{g}_d \cdot \nabla \phi \phi)}{\partial n} \Big|_\Gamma \\ &= \frac{\partial \bar{g}_d}{\partial n} \Big|_\Gamma - \frac{\partial(\nabla \bar{g}_d \cdot \nabla \phi)}{\partial n} \Big|_\Gamma \phi \Big|_\Gamma - \frac{\partial \phi}{\partial n} \Big|_\Gamma (\nabla \bar{g}_d \cdot \nabla \phi) \Big|_\Gamma \\ &= \frac{\partial \bar{g}_d}{\partial n} \Big|_\Gamma - \frac{\partial(\nabla \bar{g}_d \cdot \nabla \phi)}{\partial n} \Big|_\Gamma 0 - 1 (\nabla \bar{g}_d \cdot n) \Big|_\Gamma \\ &= \frac{\partial \bar{g}_d}{\partial n} \Big|_\Gamma - \frac{\partial \bar{g}_d}{\partial n} \Big|_\Gamma = 0. \end{aligned} \quad (18)$$

$Rg_d$  is then a suitable lifting at the continuous level in the sense that the conditions (10) are satisfied.

**Example 2.1.** We consider a 1D example for the sake of clarity. The domain is  $\Omega = (0, 1)$  and the interface  $\Gamma$  is composed of the two points  $x_1 = \pi^{-1}$  and  $x_2 = 1 - \pi^{-1}$ . The level set function is defined accordingly. We choose  $g_d(x) = \sin(3x)$  and  $g_n(x) = \exp(2x)$ , so that  $\bar{g}_d(x) = g_d(x)$  and  $\bar{g}_n(x) = g_n(x)$ ,  $x \in (0, 1)$ . The continuous liftings  $Rg_d$  and  $Sg_n$  are shown in figure 1.

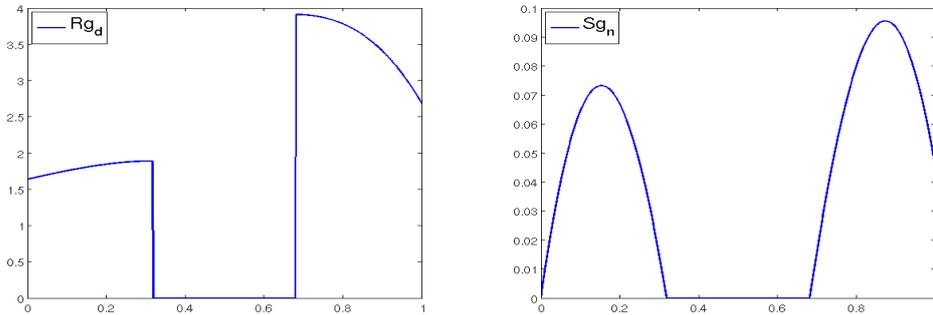


Figure 1: Continuous liftings obtained for the example 2.1:  $Rg_d$  (left) and  $Sg_n$  (right).

### 3 Finite element approximation

In this section, we will address the numerical approximation of the problems introduced thus far, together with the introduction of the approximate lifting operators.

We consider a uniform triangulation of  $\Omega$  made of elements  $K$  (intervals if  $N = 1$ , triangles if  $N = 2$  or tetrahedra if  $N = 3$ ). The interface  $\Gamma$  may intersect the elements  $K$  arbitrarily. As finite element space, we use the continuous  $\mathbb{P}_k$  elements with  $k \geq 1$ :

$$V_h = \{v_h \in H_0^1(\Omega) \cap C^0(\bar{\Omega}) : v_h|_K \in \mathbb{P}_k \ \forall K\}, \quad (19)$$

and we denote by  $\{\Psi_j\}$  the basis functions of  $V_h$ .

The finite element approximation of (8) reads: find  $\bar{u}_h \in V_h$  such that

$$\int_{\Omega} \nabla \bar{u}_h \cdot \nabla v_h = \int_{\Gamma} g_n v_h + \sum_{i=1}^2 \int_{\Omega_i} f_i v_{hi} - \sum_{i=1}^2 \int_{\Omega_i} \nabla R_i^h g_d \cdot \nabla v_{hi} \quad \forall v_h \in V_h, \quad (20)$$

while that of (12) becomes: find  $\hat{u}_h \in V_h$  such that

$$\int_{\Omega} \nabla \hat{u}_h \cdot \nabla v_h = \int_{\Gamma} g_n v_h + \sum_{i=1}^2 \int_{\Omega_i} f_i v_{hi} - \sum_{i=1}^2 \int_{\Omega_i} \nabla (R_i^h g_d + S_i^h g_n) \cdot \nabla v_{hi} \quad \forall v_h \in V_h. \quad (21)$$

#### 3.1 Discrete lifting operators

We introduce now suitable finite element approximations of the continuous liftings  $Rg_d$  and  $Sg_n$ . At the discrete level, we would like to have liftings with minimal support around the interface. Ideally, only the cells crossed by the interface would ought be used in order to keep the computational cost of the finite element approximation as low as possible.

Let  $\pi_h^k : C^0(\bar{\Omega}) \rightarrow V_h$  be the classical finite element interpolant operator

$$\pi_h^k(v) = \sum_j v(x_j) \Psi_j \quad (22)$$

that is  $\pi_h^k(v)$  is the unique function in  $V_h$  which takes the same values of  $v$  at all finite element nodes  $x_j$  while  $\Psi_j$  is the characteristic basis function associated with  $x_j$ , that is  $\Psi_j \in V_h : \Psi_j(x_i) = \delta_{ij} \ \forall i, j$  (see [8]).

Remark now that both liftings  $Rg_d$  and  $Sg_n$  that we have defined at the continuous level are a product of the Heaviside function by a continuous function. Then, for any function  $T(x, y) = H(\phi)(x)t(x)$  with  $t(x) \in C^0(\bar{\Omega})$ , we introduce the following operator:

$$\Pi_h^k(T)(x) = \begin{cases} \pi_h^k(t)(x) & \text{if } \phi(x) \geq 0 \\ 0 & \text{if } \phi(x) < 0. \end{cases} \quad (23)$$

We define then the discrete liftings  $R_{glo}^h g_d = \Pi_h^k(Rg_d)$  and  $S_{glo}^h g_n = \Pi_h^k(Sg_n)$ . The index *glo* stands for *global* and it indicates that these functions are defined on the global domain  $\Omega$ .

**Example 3.1.** With the same settings of Example 2.1, we perform the interpolation on a mesh with 5 intervals using  $\mathbb{P}_1$  finite elements. The resulting liftings are shown in figure 2.

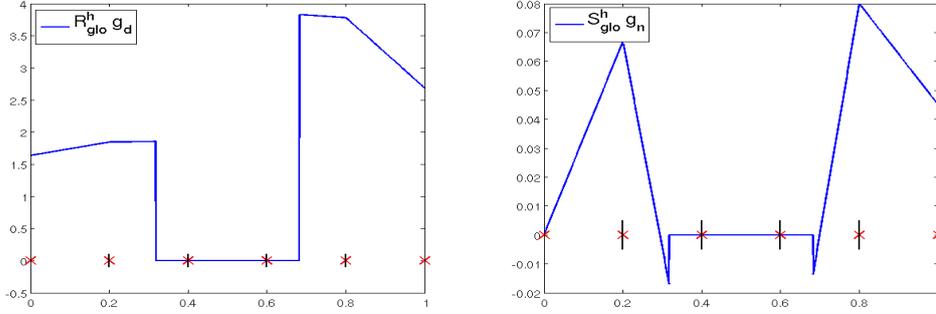


Figure 2: Discrete global liftings  $R_{glo}^h g_d$  (left) and  $S_{glo}^h g_n$  (right). The red crosses show the location of the degrees of freedom.

To reduce the computational cost due to working with the global support of the liftings, we introduce a region  $\Omega_\Gamma$  of width  $h$  around the interface  $\Gamma$  (see Fig. 3) and we reduce the support of  $R_{glo}^h g_d$  and  $S_{glo}^h g_n$  to  $\Omega_\Gamma$  in a way that their values on  $\partial\Omega_\Gamma$  is 0. Notice that  $\Omega_\Gamma$  corresponds to the strip of width  $h$  formed by those triangles that intersect the interface.

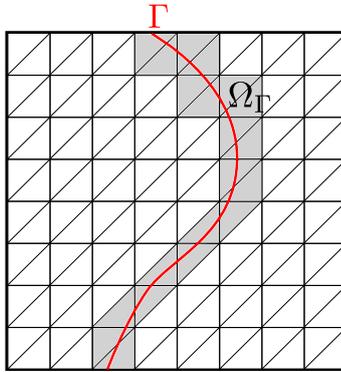


Figure 3: Illustration of  $\Omega_\Gamma$  in a 2D case.

We denote here  $T_{glo}^h$  either  $R_{glo}^h g_d$  or  $S_{glo}^h g_n$ . As by construction  $T_{glo}^h$  belongs to the discrete space  $V_h$ , we can express it on each side of the interface using the finite element basis:

$$T_{glo}^h|_K(x) = \begin{cases} \sum_i \alpha_i \Psi_i(x) & \text{if } \phi(x) \geq 0 \\ 0 & \text{if } \phi(x) < 0. \end{cases} \quad (24)$$

Adding any function of the finite element space on both sides of the interface

does not change the jumps of this lifting. Then we can define the lifting  $T^h$  as

$$T^h|_K = \begin{cases} T_{glo}^h|_K - \sum_{\phi_i \geq 0} \alpha_i \Psi_i & \text{if } K \cap \Gamma \neq \emptyset \\ 0 & \text{otherwise,} \end{cases} \quad (25)$$

where  $\phi_i = \phi(x_i)$ . By definition, the support of  $T^h$  is  $\Omega_\Gamma$ . From the explicit expression of the lifting on the elements  $K$  crossed by the interface  $\Gamma$ :

$$T^h|_K(x) = \begin{cases} \sum_{\phi_i < 0} \alpha_i \Psi_i(x) & \text{if } \phi(x) \geq 0 \\ -\sum_{\phi_i \geq 0} \alpha_i \Psi_i(x) & \text{if } \phi(x) < 0, \end{cases} \quad (26)$$

we can see that  $T^h$  is actually 0 on  $\partial\Omega_\Gamma$  and it is then extended by continuity outside  $\Omega_\Gamma$ . Applying this procedure to  $R_{glo}^h g_d$  and  $S_{glo}^h g_n$  we obtain the liftings  $R^h g_d$  and  $S^h g_n$  that fulfill all our requirements.

We can now give the explicit expression of the two liftings:

$$R^h g_d|_K(x) = \begin{cases} \sum_{\phi_i < 0} (\bar{g}_d(x_i) - \nabla \bar{g}_d(x_i) \cdot \nabla \phi(x_i) \phi(x_i)) \Psi_i(x) & \text{if } \phi(x) \geq 0 \\ -\sum_{\phi_i \geq 0} (\bar{g}_d(x_i) - \nabla \bar{g}_d(x_i) \cdot \nabla \phi(x_i) \phi(x_i)) \Psi_i(x) & \text{if } \phi(x) < 0, \end{cases} \quad (27)$$

$$S^h g_n|_K(x) = \begin{cases} \sum_{\phi_i < 0} (\bar{g}_n(x_i) \phi(x_i)) \Psi_i(x) & \text{if } \phi(x) \geq 0 \\ -\sum_{\phi_i \geq 0} (\bar{g}_n(x_i) \phi(x_i)) \Psi_i(x) & \text{if } \phi(x) < 0. \end{cases} \quad (28)$$

**Example 3.2.** In figure 4 we show the result of support reduction for the liftings of the example 3.1 using  $\mathbb{P}_1$  polynomials, while in figure 5 we show the lifting  $R^h g_d$  computed using  $\mathbb{P}_2$  and  $\mathbb{P}_3$  polynomials.

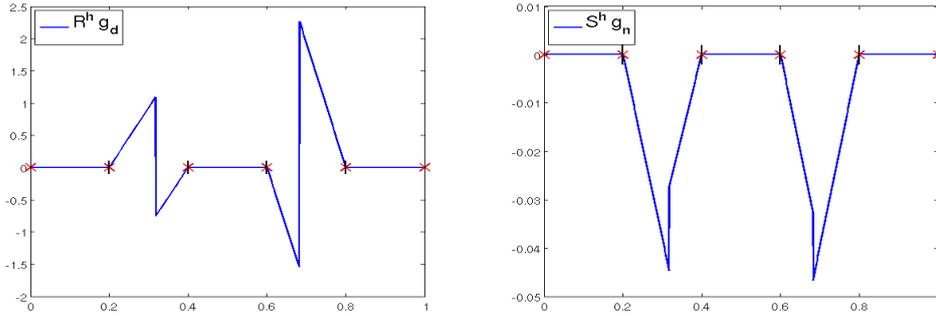


Figure 4: Liftings after the support reduction:  $R^h g_d$  (left) and  $S^h g_n$  (right). The red crosses show the location of the degrees of freedom.

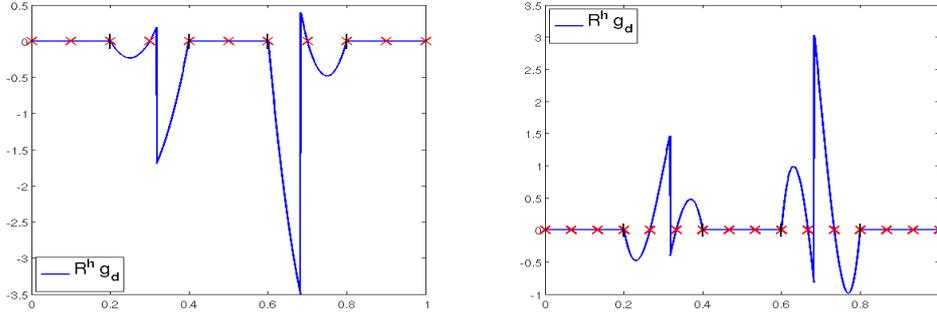


Figure 5: Liftings  $R^h g_d$  after the support reduction using  $\mathbb{P}_2$  elements (left) and  $\mathbb{P}_3$  elements (right). The red crosses show the location of the degrees of freedom.

### 3.2 The SESIC method

The SESIC method that we propose is obtained by using the discrete lifting operators (27) and (28) in the context of the weak formulation (21). The only ingredient that remains to be detailed is the numerical integration formula that will be used to compute the new terms in the weak formulation (21). More precisely, we have to perform one integral on the interface  $\Gamma$  of a continuous function and two integrals over  $\Omega$  of possibly discontinuous functions ( $\nabla R^h g_d$  and  $\nabla S^h g_n$  might be discontinuous across  $\Gamma$ ). We propose two different methods for the integration.

#### “Exact” integration

The first method consists in building quadrature rules that take into account the interface. A possible way to integrate singular functions of type

$$\int_{\Gamma} f = \int_{\Omega} \delta_{\Gamma} f$$

is to reconstruct explicitly the interface  $\Gamma$  and to use on it a  $(N-1)$ -dimensional quadrature rule. If  $N=1$ , the interface reduces to a point and the integration requires only to evaluate  $f$  at a given point. If  $N=2$ , the elements are triangles and then the interface in a single triangle is a segment in the case of a piecewise linear approximation. To apply a suitable integration rule on this segment, we need to compute the intersections of  $\Gamma$  with the edges of the triangle.

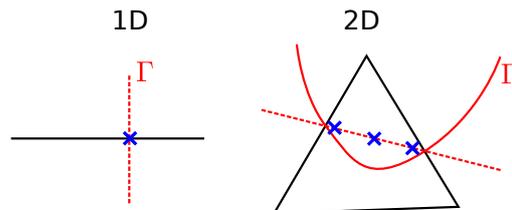


Figure 6: Illustration of the methods used for the computation of the line integral in 1D and 2D.

On the other hand, to integrate discontinuous functions like

$$\int_{\Omega} f_1 + H(\phi)f_2,$$

we define a quadrature rule for an element crossed by  $\Gamma$  considering a quadrature rule on the polygons on each side of the interface. More precisely, if  $N = 1$ , we compute the location of the interface and we combine a quadrature rule for segments on each side of the interface. When  $N = 2$ , the triangles crossed by  $\Gamma$  are split into a triangle and a quadrangle. To integrate discontinuous functions, we combine then a quadrature rule for triangles and a quadrature rule for quadrangles.

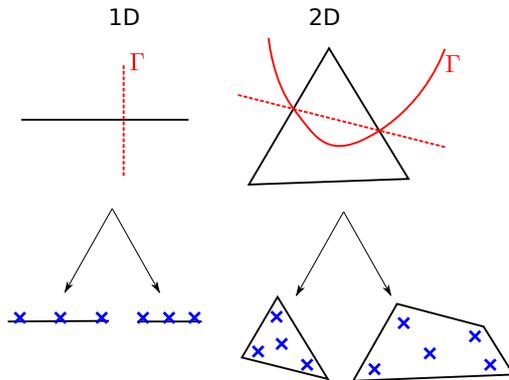


Figure 7: Illustration of the methods used for the computation of the discontinuous functions in 1D and 2D.

The case  $N = 3$  requires a different treatment. Indeed, even if similar methods are available for 3D simulations, they lead to complicated schemes where many different cases have to be distinguished depending on the way the interface cuts the tetrahedra. This is why we have used another method that is simpler and more suitable for high space dimensions or for high polynomial orders of approximation.

### Smooth integration

The idea is to approximate singular or discontinuous integrands by smooth functions. For example, we make the following approximation:

$$\int_{\Gamma} g_n v = \int_{\Omega} g_n v \delta_{\Gamma} \cong \int_{\Omega} g_n v \delta_w$$

where  $\delta_w$  is an approximation of  $\delta_{\Gamma}$  whose support is limited to a band of width  $w$  around  $\Gamma$ . This method is quite widely used, even if, often, there is no real control of the error produced. We refer here to [10] for the error analysis of the regularization step. In this analysis, two errors are distinguished:

- the analytical error produced by the introduction of the regularizing function:

$$\left| \int_{\Gamma} g_n v - \int_{\Omega} g_n v \delta_w \right| ;$$

- the quadrature error coming from the inexact integration of the regularized integrand.

The usual procedure is to take  $w$  as a given quantity of cells, what means that  $w$  is proportional to  $h$ . However, the idea that we apply in this paper is to use a *width  $w$  that is proportional to  $\sqrt{h}$* . Our choice is motivated by the following considerations. First of all, we have to remark that the function  $\delta_w$  must have the form  $\delta_w(d) = \frac{1}{w}\hat{\delta}(d/w)$  where  $\hat{\delta}$  is a function that does not depend on  $w$ , the factor  $\frac{1}{w}$  making the weight of  $\delta_w$  constant with respect to  $w$ , and  $d$  is the distance to the interface, that is  $d(\mathbf{x}) = \phi(\mathbf{x})$ .

- The analytical error is then proportional to  $w^\beta$ , where  $\beta$  can be computed using the properties (vanishing moments) of  $\hat{\delta}$  [10].
- If the integrand  $g_n v \delta_w$  is a function of  $C^k(\Omega)$  and the quadrature rule (based on the finite element mesh, with typical size  $h$ ) has a degree of exactness  $k-1$ , the quadrature error is proportional to  $h^k \|(g_n v \delta_w)^{(k)}\|_{L^\infty(\Omega)}$ . However,  $\delta_w^{(k)}$  scales like  $w^{-(k+1)}$ . Therefore, the quadrature error is dominated by  $h^k w^{-(k+1)}$ .

Based on these arguments, we derive that, by choosing  $w = h$  we cannot ensure that the quadrature error will decrease with  $h$ . Indeed, with  $w = h$ , the number of quadrature points in the band of width  $w$  is constant while the function  $\delta_w$  is becoming steeper to conserve the mass. Our choice of  $w = c\sqrt{h}$  leads the analytical error to be controlled by  $h^{\beta/2}$  and the quadrature error by  $h^{(k-1)/2}$ . We can then fully control the decay rate of the overall error by choosing the appropriate  $\hat{\delta}$  function.

If we look for second order accuracy, building  $\hat{\delta}$  with 3 vanishing moments (then  $\beta = 4$ , see [10]) and 5 continuous derivatives would be sufficient. By looking for the polynomial function with the smallest degree featuring these properties, we end up with:

$$\hat{\delta}(d) = \frac{6435}{8192}(3 - 35d^2 + 147d^4 - 315d^6 + 385d^8 - 273d^{10} + 105d^{12} - 17d^{14})$$

The same approach can be applied for transition function to integrate discontinuous integrand across the interface. We used for our tests the transition function:

$$\hat{H}(d) = \int_{-1}^d \hat{\delta}(\xi) d\xi.$$

### 3.3 The ESIC method

As stated before, the SESIC was inspired by the ESIC method first proposed in [4]. For the sake of comparison, let us recall the principle of the ESIC method and emphasize the differences with the SESIC method.

The two methods are built on different weak formulations of the given problem (1)-(3). SESIC stems from the weak form (21) whereas in order to get rid

of the line integral in (21), Huh and Sethian perform a counter integration by parts yielding:

$$\begin{aligned} \int_{\Gamma} g_n v_h - \sum_{i=1}^2 \int_{\Omega_i} \nabla S_i g_n \cdot \nabla v_{hi} \\ = \sum_{i=1}^2 \sum_K \int_{K \cap \Omega_i} \Delta S_i g_n v_{hi} - \sum_{i=1}^2 \sum_K \int_{\partial K} \frac{\partial S_i g_n}{\partial n} v_{hi}. \end{aligned} \quad (29)$$

However, this formulation introduces a new error source, as it makes use of the equality  $g_n = \left[ \frac{\partial S g_n}{\partial n} \right]$ , which might be wrong at the discrete level. This is in fact documented by the numerical tests that we will present in section 4.

The second major difference relies on the construction of the lifting operators. According to [4], the extension  $\tilde{g}_*$  on  $\Omega_{\Gamma}$  of a generic function  $g_*$  defined only on  $\Gamma$  is defined as follows

$$\tilde{g}_*(x) = g_*(x_{\Gamma}) \quad \text{for all } x \in \Omega_{\Gamma}, \quad (30)$$

where  $x_{\Gamma}$  is the point of  $\Gamma$  that minimizes the distance to  $x$ . As a consequence,  $\tilde{g}_*$  is constant along any normal direction issuing from  $\Gamma$ .

Let now  $\tilde{g}_d$  be the extension in  $\Omega_{\Gamma}$  of  $g_d$  according to (30). Then, considering the triangulation of  $\Omega$  and the basis  $\{\Psi_j\}$  of  $V_h$  (19), in [4] the lifting is defined as follows

$$\tilde{R}g_d = \begin{cases} - \sum_{\phi_j \geq 0} \Psi_j \tilde{g}_d & \text{in } \Omega_1 \\ \sum_{\phi_j < 0} \Psi_j \tilde{g}_d & \text{in } \Omega_2. \end{cases} \quad (31)$$

By construction,  $\tilde{R}g_d$  has the prescribed jump  $[[\tilde{R}g_d]]_{\Gamma} = g_d$  and it has continuous normal derivative across  $\Gamma$ :  $\left[ \frac{\partial \tilde{R}g_d}{\partial n} \right]_{\Gamma} = 0$ . At the discrete level, the lifting becomes:

$$\tilde{R}^h g_d = \begin{cases} - \sum_{\phi_j \geq 0} \Psi_j \tilde{g}_{d,j} & \text{in } \Omega_1 \\ \sum_{\phi_j < 0} \Psi_j \tilde{g}_{d,j} & \text{in } \Omega_2, \end{cases} \quad (32)$$

where  $\tilde{g}_{d,j}$  denotes the value of the function  $\tilde{g}_d$  at the node  $x_j$ . In this case, the jumps through the interface are satisfied in an interpolation sense: indeed it is  $[[\tilde{R}^h g_d]]_{\Gamma} = \sum_j \tilde{g}_{d,j} \Psi_j$  and  $\left[ \frac{\partial \tilde{R}^h g_d}{\partial n} \right]_{\Gamma} = \sum_j \tilde{g}_{d,j} \frac{\partial \Psi_j}{\partial n}$ .

The potential disadvantage of this methodology with respect to the approach that we have developed in the previous section 3.1 is that the interface has to be reconstructed since in (30) the closest point  $x_{\Gamma}$  is requested and, according to [4], this operation has to be performed using the interface explicitly.

A similar approach is proposed in [4] to construct the lifting of  $g_n$ . Note that after multiplying a function built as in (32) by the level-set function  $\phi$ , it becomes continuous across the interface (since  $\phi$  is equal to 0 on the interface),

while its normal derivative exhibits a jump of the desired magnitude. So, after constructing  $\tilde{g}_n$  as in (30), the discrete lifting for  $g_n$  becomes:

$$\tilde{S}^h g_n = \begin{cases} -\sum_{\phi_j \geq 0} \Psi_j \tilde{g}_{n_j} \phi & \text{in } \Omega_1 \\ \sum_{\phi_j < 0} \Psi_j \tilde{g}_{n_j} \phi & \text{in } \Omega_2. \end{cases} \quad (33)$$

The potential drawback of this construction is that the multiplication by  $\phi$  increases the polynomial order of the lifting function, thus requiring a polynomial refinement in the neighborhood of the interface. Moreover, the closest point extension requires again to rebuild the interface explicitly.

**Remark 3.1.** *The construction of the liftings of Sect. 3.1 can be seen as a generalization of those presented here. Indeed, if instead of the extensions of  $g_*$  introduced in Sect. 2.1, we took the closest point extension (30), then the liftings  $R^h g_d$  and  $S^h g_n$  would coincide with (32) and (33). Notice again that in Sect. 3.1 we do not need at all to reconstruct  $\Gamma$ .*

### 3.4 Alternative construction of the discrete liftings

As alternative to (23), one may consider, instead the interpolation introduced in (23), a suitable projection (e.g. with respect to the  $L^2$  or the  $H^1$  scalar products) of the continuous liftings (13) and (16) on the finite element space  $V_h$ . This alternative could be useful in those cases where interpolation operator cannot be defined. However, interpolation has to be privileged if it is available as it usually leads to the same error behavior and enables an explicit formulation of the lifting at the discrete level, while the projection requires to solve a linear system.

Remark also that in the case of weak regularity in (23) another option to construct the discrete liftings would be to consider Clement interpolation instead of Lagrange interpolation.

Finally, let us point out that to construct the liftings we could also use a more analytical approach instead of the ones illustrated before. A possibility would consist in solving a particular PDE in each triangle crossed by the interface to play the role of extension operators. To control both the trace and the normal derivative at the same time on the interface, we can solve a fourth order biharmonic problem in  $K_i = K \cap \Omega_i$  for each  $K$  such that  $K \cap \Gamma \neq \emptyset$ . Precisely, the problem would be: find  $R \in H^2(K_i)$  such that:

$$\Delta^2 R = 0 \quad \text{in } K_i \quad (34)$$

$$R = g_d \quad \text{on } K_\Gamma \quad (35)$$

$$\frac{\partial R}{\partial n} = g_n \quad \text{on } K_\Gamma. \quad (36)$$

Suitable Dirichlet and Neumann boundary conditions are then added on  $\partial_i K_i$  to close the problem and to keep the support of  $R$  restricted to  $\Omega_\Gamma$  ( $K_\Gamma$  and  $\partial_i K_i$  are defined as in figure 8).

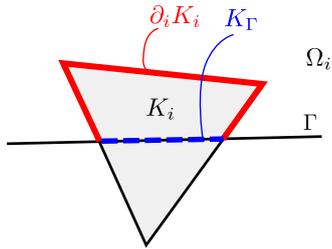


Figure 8: Illustration of the geometry of a triangle  $K$  cut by the interface  $\Gamma$ .

## 4 Numerical results

In this section, we present numerical results obtained using the methodologies described in the previous sections for different geometric dimensions.

### 4.1 1D test case

First of all, we consider a 1D Poisson problem, as this allows us to make complete error measurements and visualizations. We consider the unit interval  $\Omega = (0, 1)$  with an interface located in  $\Gamma = \{\pi^{-1}\}$  so that the uniform meshes that we will use will not conform with the interface. The level set function is defined as  $\phi(x) = \pi^{-1} - x$ . The Poisson problem consists in finding  $u : \Omega \rightarrow \mathbb{R}$  such that

$$\begin{aligned} -u''(x) &= -e^x & \text{in } \Omega \\ u(0) &= 1 \\ u(1) &= e + 2, \end{aligned} \tag{37}$$

with the jump conditions

$$[[u]]_{\Gamma} = -2\pi^{-1} \quad \left[ \left[ \frac{\partial u}{\partial n} \right] \right]_{\Gamma} = 2. \tag{38}$$

The exact solution reads

$$u = \begin{cases} e^x & \text{if } x \leq \pi^{-1} \\ e^x + 2x & \text{if } x > \pi^{-1}. \end{cases} \tag{39}$$

As both jump conditions are non-homogeneous, we need to extend them in the whole domain  $\Omega$ . To this aim, we define two possible sets of extension to highlight the role of the choice of the extensions for the convergence of the method. The first set is made of arbitrary functions:

$$\bar{g}_d(x) = -(2\pi^{-1} + \sin(x - \pi^{-1})) \quad \bar{g}_n(x) = 1 + e^{(x - \pi^{-1})} \tag{40}$$

while the second set, called simplified extensions, is made of constant functions:

$$\bar{\bar{g}}_d(x) = -2\pi^{-1} \quad \bar{\bar{g}}_n(x) = 2 \tag{41}$$

For the simplified extensions (41), thanks to the definitions (14) and (17), it is easy to see that the interpolation does not introduce any error while if we take

the extensions in (40), the interpolation will produce some error on the jumps and the conditions (38) will not be satisfied exactly. In the latter case, we have measured the error due to the liftings on the jump conditions (38) for  $\mathbb{P}_1$  and  $\mathbb{P}_2$  finite elements. The following table shows the order of convergence of these errors for  $h \rightarrow 0$ :

Elements	Convergence rates for errors on:			
	$\llbracket Rg_d \rrbracket$	$\left\  \frac{\partial Rg_d}{\partial n} \right\ $	$\llbracket Sg_n \rrbracket$	$\left\  \frac{\partial Sg_n}{\partial n} \right\ $
$\mathbb{P}_1$	3	2	2	1
$\mathbb{P}_2$	3	2	3	2

We can see that the orders are optimal for all the quantities and that we have a superconvergence for  $Rg_d$  with  $\mathbb{P}_1$  elements. This is because of the special circumstance that we are interpolating a function with zero derivative.

We apply now the SESIC method to solve the 1D problem. To measure the associated error, we use three error measures:

- the  $H^1$  norm of the error in the domain  $\Omega^* = (0, \pi^{-1} - 0.1) \cup (\pi^{-1} + 0.1, 1)$ ,
- the  $L^2$  norm of the error in the domain  $\Omega^*$ ,
- the  $L^\infty$  norm of the error in the entire domain  $\Omega$ .

We typically get a quite smooth error pattern on the whole domain, as shown in Fig. 9 (left), what provides an evidence that all the components of the error are balanced.

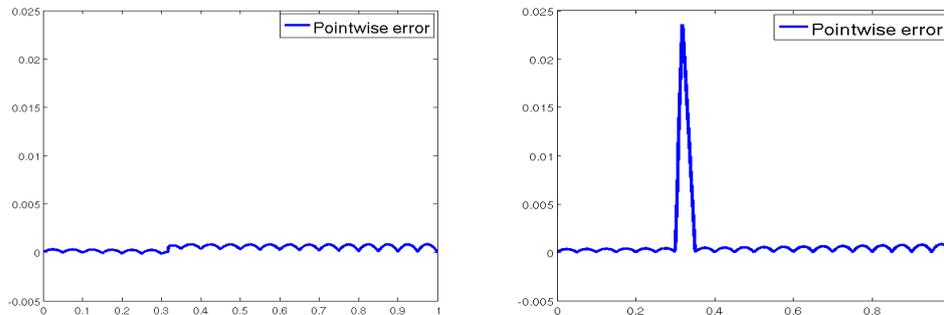


Figure 9: Pointwise error in the solution for the 1D test using  $\mathbb{P}_1$  elements and a grid of 20 intervals: left, using the SESIC method; right, without lifting for the normal derivative.

However, as shown in the next table, we do not get optimal orders for the maximum error with  $\mathbb{P}_2$  elements, while the errors in  $\Omega_*$  and the errors for the  $\mathbb{P}_1$  elements exhibit optimal convergence rates:

Norm	$\mathbb{P}_1$ elements	$\mathbb{P}_2$ elements
$L^2(\Omega^*)$	2	3
$H^1(\Omega^*)$	1	2
$L^\infty(\Omega)$	2	2

Optimal orders also in the  $L^\infty$  norm can be recovered if we use the simplified extensions (41).

Using this test case, we can also provide a justification of the use of the lifting for the jump in the normal derivative. If we do not take into account the lifting  $Sg_n$ , we have to use the weak formulation (20). We keep the same definition for  $Rg_d$  (with the extension given in (40)).

This produces results that are different from our method mainly near the interface: figure 9 (right) shows a large error peak in the element crossed by the interface. The error located in that element is far larger than the interpolation error visible in the other elements.

This additional error comes from the fact that the underlying finite element space cannot reproduce jumps inside the elements. It is then impossible to reduce this error without providing the finite element space with the ability to capture jumps. In the SESIC method, this is the role of the lifting, that carry the jumps but does not belong to the finite element space. We can also see this behavior in the following table that shows the convergence orders for the method without  $Sg_n$ : even if the errors computed in the domain  $\Omega_*$  show optimal convergence orders, the high error near the interface reduces the convergence rates in the maximum norm.

Norm	$\mathbb{P}_1$ elements	$\mathbb{P}_2$ elements
$L^2(\Omega^*)$	2	3
$H^1(\Omega^*)$	1	2
$L^\infty(\Omega)$	1	1

The role of the lifting  $Sg_n$  for the normal derivative is then clear: it helps to reduce the magnitude of the error in the neighborhood of the interface.

For the sake of comparing our method with the ESIC method described in section 3.3, we have also tested the weak formulation modified with (29) and with the liftings described in section 3.1. This allows us to bring to light the consequences of using (29). The following table shows the convergence rates for this test case: we can clearly see that the convergences are slower than with the weak formulation (21).

Norm	$\mathbb{P}_1$ elements	$\mathbb{P}_2$ elements
$L^2(\Omega^*)$	1	2
$H^1(\Omega^*)$	1	2
$L^\infty(\Omega)$	1	2

The following picture shows the typical pattern that we get using the modified weak formulation. The solution looks like if the force applied on the interface (by the term  $\int_\Gamma g_n v_h$  in (20) and (21)) was badly estimated, leading to the trend of the error to be greater near the interface, while producing no peak there.

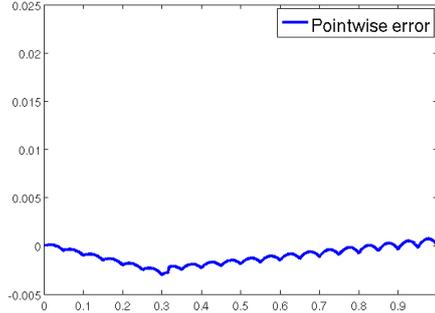


Figure 10: Pointwise error in the solution for the 1D test, using  $\mathbb{P}_1$  elements on 20 intervals when using the modification (29)

The origin of the error is also emphasized in the next figure, that shows that there is a big correlation between the error  $\left| \left[ \left[ \frac{\partial S g_n}{\partial n} \right] \right] - g_n \right|$  and the  $L^2$  error.

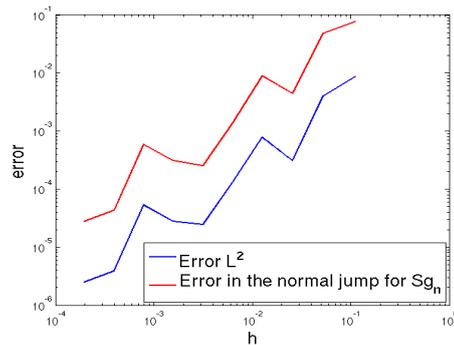


Figure 11: Comparison between the error on the normal jump of  $Sg_n$  and the global  $L^2$  error.

This error does not show up in the original ESIC method as shown in the next table.

Norm	Convergence rate
$L^2(\Omega^*)$	2
$H^1(\Omega^*)$	1
$L^\infty(\Omega)$	2

The reason is that in the latter method, polynomial refinement is performed near the interface. In this example,  $\mathbb{P}_1$  elements have been used except for the elements containing the interface where a  $\mathbb{P}_2$  basis was defined. As shown at the beginning of this section, when  $\mathbb{P}_2$  are used, the error  $\left| \left[ \left[ \frac{\partial S g_n}{\partial n} \right] \right] - g_n \right|$  has a second order convergence and then has the same behavior as the interpolation error. However this approach requires an additional programming effort as well as unnecessary addition of degrees of freedom: the next figure shows the pointwise error for the ESIC method and we can observe that the error in the element containing  $\Gamma$  is smaller than in the rest of the domain.

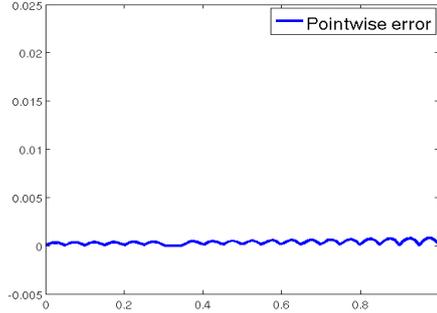


Figure 12: Pointwise error in the solution for the 1D test, using the ESIC method with 20 intervals.

Finally, we investigate the effects of computing integrals using the regularized integrands introduced in section 3.2. We test both widths  $w = \sqrt{\frac{h}{2}}$  and  $w = h$ . The effects of the thickness of the regularization band is clearly visible in Fig. 13 (left) where we show the behavior of the  $L^2(\Omega^*)$  norm of the error.

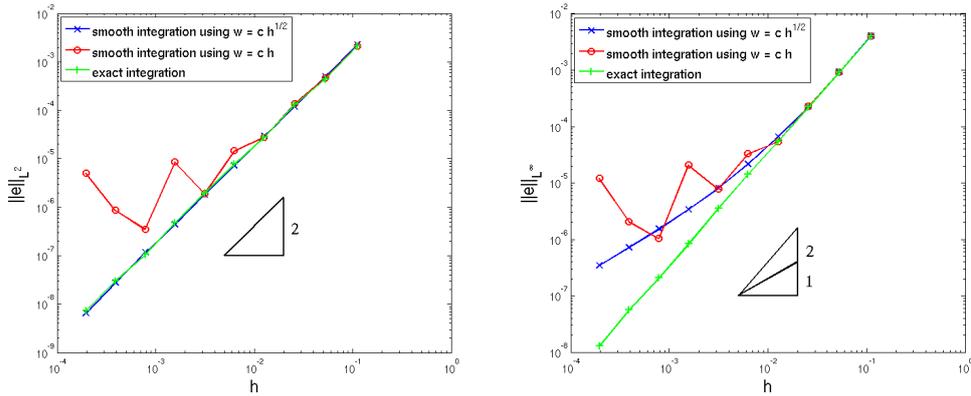


Figure 13:  $L^2(\Omega^*)$  error (left) and  $L^\infty(\Omega)$  error (right) associated with the different integration methods.

As stated previously in this section, optimal order of convergence is achieved with the exact integration. Using the proposed smooth integration, i.e., with  $w$  proportional to  $\sqrt{h}$ , we also get the optimal order of convergence in this norm. On the contrary, using a regularization band with thickness  $w = h$ , we obtain an unpredictable behavior when  $h$  becomes small. The convergence in that case is difficult to assess. We can observe the same kind of behavior in the  $L^\infty(\Omega)$  norm as shown in Fig. 13 (right). In this case we can see that using  $w$  proportional to  $\sqrt{h}$  leads to a convergence slower than the optimal one: if for coarse meshes the convergence rate seems to be close to 2, it then slows down to 1 for finer meshes.

All these results correspond quite well to the remarks that we made in Sect. 3.2. The smooth integration using  $w$  proportional to  $\sqrt{h}$  permits to control the error leading to regular convergence, even optimal in the  $L^2(\Omega^*)$  norm. This

means that the error generated near the interface, reported in the  $L^\infty(\Omega)$  norm, is confined in that area and does not pollute the solution in the whole domain. On the contrary, with  $w = h$ , we lose the control on the quadrature error causing a large error in the interface area that eventually spreads in the whole domain.

## 4.2 2D test case

We test our method on the two dimensional test case defined in [4]. This test is quite simple as the exact solution is continuous, so that the jump is only in the normal derivative. The domain is defined as the square  $\Omega = (-1, 1)^2$  and the interface is the circle with radius 0.5 centered at the origin.

The exact solution reads:

$$u(x, y) = \begin{cases} 1 & \text{if } x^2 + y^2 \leq 0.25 \\ 1 - \log(2\sqrt{x^2 + y^2}) & \text{if } x^2 + y^2 > 0.25. \end{cases}$$

Dirichlet boundary conditions are set to ensure this exact solution and the jump in the normal derivative to be

$$\left[ \left[ \frac{\partial u}{\partial n} \right] \right]_{\Gamma} = -2.$$

Cartesian meshes with  $n$  cells on each side were used. The results that we obtain are listed in the following table:

$n$	Maximal error on $\Gamma$	rate	Maximal error in $\Omega$	rate
9	$7.13 \times 10^{-3}$		$1.84 \times 10^{-2}$	
19	$2.85 \times 10^{-3}$	1.23	$4.61 \times 10^{-3}$	1.85
39	$7.10 \times 10^{-4}$	1.93	$1.13 \times 10^{-3}$	1.96
79	$1.71 \times 10^{-4}$	2.02	$2.71 \times 10^{-4}$	2.02

We can see that the SESIC method gives optimal orders of convergence both at the interface and in the entire domain. This means that the error decreases with the same rate everywhere in the domain, included near and on the interface. Moreover, the magnitude of the error is lower than for the methods (ESIC, XFEM and IBM) compared in [4], while being easier to implement and cheaper to compute.

## 4.3 3D test case

We finally consider a 3D problem. This example was implemented in the parallel version of the finite element library LifeV ([www.lifev.org](http://www.lifev.org)). We consider the domain  $\Omega = (-1, 1)^3$ . The level set function is  $\phi(x, y, z) = (x^2 + y^2 + z^2)^{1/2} - 0.5$  so that the interface  $\Gamma$  is a sphere centered in the origin with a radius 0.5. In  $\Omega$ , we want to find the solution  $u : \Omega \rightarrow \mathbb{R}$  of the problem  $-\Delta u = 0$  with jumps conditions through  $\Gamma$ :

$$[[u]]_{\Gamma} = 2 - e^{x+z} \sin(\sqrt{2}y)$$

$$\left[ \left[ \frac{\partial u}{\partial n} \right] \right]_{\Gamma} = 4 + 2e^{x+z} \left( (x+z) \sin(\sqrt{2}y) + \sqrt{2}y \cos(\sqrt{2}y) \right).$$

Boundary conditions are such that the exact solution is

$$u(x, y, z) = \begin{cases} (x^2 + y^2 + z^2)^{-1/2} & \text{if } \phi(x, y, z) \geq 0 \\ e^{x+z} \sin(\sqrt{2}y) & \text{if } \phi(x, y, z) < 0. \end{cases}$$

We solved this problem using  $\mathbb{P}_1$  finite elements. To measure the error, we computed both the  $L^2$  error in the domain  $\Omega^* = \{\mathbf{x} \in \Omega \mid |\phi(\mathbf{x})| > 0.1\}$  for regularity and maximal error in all the finite element nodes (denoted hereafter  $l^\infty$ ). We used Cartesian meshes with  $n$  representing the number of nodes in each direction. The computed errors and convergence rates are given in the next table.

$n$	degrees of freedom	error $L^2(\Omega^*)$	rate	error $l^\infty$	rate
5	125	$6.26 \times 10^{-1}$		$3.52 \times 10^{-1}$	
10	1000	$2.11 \times 10^{-1}$	1.34	$8.39 \times 10^{-2}$	1.79
20	8000	$1.72 \times 10^{-2}$	3.36	$2.09 \times 10^{-2}$	1.86
40	64000	$3.07 \times 10^{-3}$	2.40	$5.63 \times 10^{-3}$	1.82
60	216000	$1.34 \times 10^{-3}$	2.00	$2.63 \times 10^{-3}$	1.84
80	512000	$7.50 \times 10^{-4}$	1.99	$1.67 \times 10^{-3}$	1.56
100	1000000	$4.78 \times 10^{-4}$	2.00	$1.17 \times 10^{-3}$	1.58

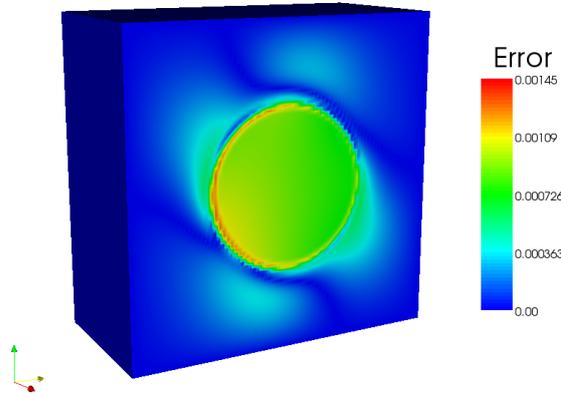


Figure 14: Representation of the error on the surface  $x = 0$  for  $n = 80$ . We can remark that the error near the interface is of the same order of magnitude as far from it.

The order of convergence in the  $L^2(\Omega^*)$  norm is close to the optimal rate 2 when  $n$  increases. This is due to the fact that the error produced by the smooth integration is well controlled and it is reflected only in the  $l^\infty$  norm as it is confined to the interface area (as shown in Fig. 14). The error in the  $l^\infty$  norm is expected to behave like the  $L^\infty(\Omega)$  error in the 1D test case, i.e., to decrease slowly for finer meshes until it reaches the convergence rate 1.

Finally, with this test case we can emphasize the need for a good integration scheme for the singular functions. Indeed, if instead of smoothing the integrands on a width proportional to  $\sqrt{h}$ , we use a width of  $h$ , we get the following results:

$n$	degrees of freedom	error $L^2(\Omega^*)$	rate	error $l^\infty$	rate
5	125	$6.33 \times 10^{-1}$		$3.18 \times 10^{-1}$	
10	1000	$2.16 \times 10^{-1}$	1.33	$7.23 \times 10^{-2}$	1.83
20	8000	$1.71 \times 10^{-2}$	3.39	$1.86 \times 10^{-2}$	1.82
40	64000	$3.92 \times 10^{-3}$	2.05	$1.27 \times 10^{-2}$	0.53
60	216000	$1.38 \times 10^{-3}$	2.52	$5.47 \times 10^{-3}$	2.03
80	512000	$1.17 \times 10^{-3}$	0.57	$7.57 \times 10^{-3}$	-1.11
100	1000000	$1.01 \times 10^{-3}$	0.65	$5.82 \times 10^{-3}$	1.16

We can observe that the convergence is slower when the mesh gets finer because the quadrature error is more dominating.

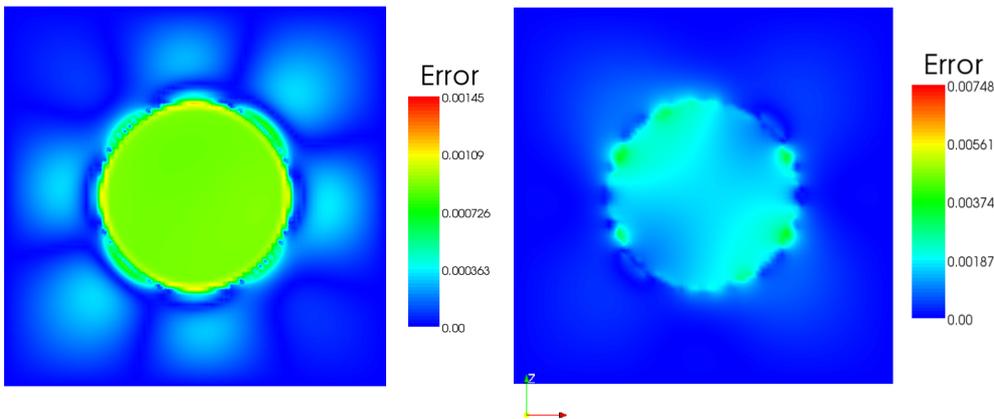


Figure 15: Error for the 3D test case on the plane  $y = 0$ . The upper figure represents the error when the interface width is  $w = 0.0796$  ( $= \sqrt{h/2}$ ), whereas for the lower figure we used  $w = 0.0127$  ( $= h$ ).

## 5 An application to free surface flows

In this section we present a possible application of the SESIC method to free surface flows. Another paper containing a more complete description of this method is in preparation.

### 5.1 Principle of the correction

We consider the problem of simulating the motion of a surface separating two incompressible Newtonian fluids. Differently from the problems introduced in the previous numerical results, the position of the interface is now unknown and represented via a level set function.

To represent the motion of the two fluids, we use the Navier-Stokes equations on each side of the surface, but with different densities  $\rho$  and viscosities  $\mu$  within

each subregion:

$$\begin{aligned}\rho\partial_t\mathbf{u} - \nabla \cdot \mathbf{T}(\mathbf{u}, p) + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}$$

where  $\mathbf{T}(\mathbf{u}, p) = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) - p\mathbf{I}$  is the Cauchy stress tensor,  $\mathbf{u}$  is the velocity of the fluid and  $p$  its pressure.  $\rho$  and  $\mu$  are constant in each fluid domain, but they are discontinuous across the interface where they can differ by several orders of magnitude. For example, if we want to simulate the surface between water and air, the density ratio is approximately 1000 whereas the viscosity ratio is 50.

Usually, gravity has an important effect on bifluid flow simulations. However, due to the difference of the densities, the gravity force  $\mathbf{f} = \rho\mathbf{g}$  exhibits a large discontinuity across the interface that leads to a jump in the normal derivative of the pressure. At the discrete level, a standard choice of the finite element spaces does not allow to represent discontinuities in the gradients of the unknown pressure inside the elements crossed by the free surface. This can result in oscillations in the pressure that may possibly pollute the whole simulation and give an unphysical shape to the surface (this is especially visible when the viscosities are low).

This problem had already been studied in [3] where the authors propose a pressure correction to get rid of these unphysical oscillations. More precisely, they add new finite element shape functions with discontinuous gradients across the interface and condense a priori the new degrees of freedom. Although this approach leads to real improvements in the numerical simulations, its main disadvantages are the construction of the new shape functions done geometrically in each element and the extra cost represented by the a priori condensation.

Using the SESIC method, we can provide a simpler and faster way of constructing such a pressure correction. Indeed, since the jump in the gradient of the pressure is known a priori:

$$\left[\left[\frac{\partial p}{\partial n}\right]\right]_{\Gamma} = \llbracket\rho\rrbracket_{\Gamma}|\mathbf{g}|,$$

we can construct a lifting for this jump like (28) with  $g_n = \llbracket\rho\rrbracket_{\Gamma}|\mathbf{g}|$ . Then, we subtract this lifting to the original pressure obtaining an additional term in the right-hand side of the momentum equation of the Navier-Stokes system. The lifting can be built without reconstructing the interface as explained in the previous sections.

## 5.2 Numerical results

We consider a test case inspired to the one in [3] that consists in water being pushed up in a rectangular pipe with constant velocity. The fluid is initially at rest with a flat horizontal free surface at height  $z = 0.25$  m from the bottom of the pipe. The liquid is then pushed upward with a vertical velocity of 1 m/s so that the free surface should reach  $z = 0.75$  m at  $t = 0.5$  s and  $z = 1.25$  m

at  $t = 1$  s. Boundary conditions are of free-slip type so that the surface should only move vertically while remaining completely flat. The viscosity of the fluid is  $\mu_f = 10^{-3}$  Pa·s and its density is  $\rho_f = 1000$  kg/m<sup>3</sup>, while the viscosity of the air-phase is  $\mu_a = 2 \cdot 10^{-5}$  Pa·s with density  $\rho_a = 1$  kg/m<sup>3</sup>. The computational mesh that we have used for our tests is structured, uniform and made of 6762 tetrahedra. We have used the MINI elements [1] for the Navier-Stokes equations first with standard quadrature formulas, then using a quadrature formula that accounts for the discontinuities in  $\rho$  and  $\mu$  across the interface (see figure 7) and finally adding also a pressure-correction term.

The numerical simulations shown in Fig. 16-17 show that without pressure correction the shape of the free surface gets distorted after few time steps and also the velocity field is not captured correctly. Moreover, we can see that only using pressure correction the right elevation of the free surface is obtained, while the other two approaches lead to some numerical dissipation which slows down the advancement of the free surface.

## 6 Conclusions

In this paper, we have investigated a new method, the SESIC method, that can be used to solve interior discontinuity interface problems. It relies on simple construction of liftings, i.e. finite element functions that are built to carry the discontinuities across the interface.

The method that we proposed was inspired by the ESIC method (see [4]) and in fact inherits some properties of that method. First of all, the jumps across the interface are actually reproduced by the method. The cost for this method is also quite low, as only the assembly for a small part (corresponding to the elements crossed by the interface) of the right hand side is needed.

The SESIC method has also the advantage of requiring no additional degree of freedom, as no new basis function nor refinement near the interface are necessary. The consequence is that the stiffness matrix remains unchanged with respect to that associated with the given PDE without interface discontinuities. At the algebraic level, solution strategies and preconditioning need not be modified. The liftings introduced in the SESIC method have removed one of the bottlenecks of the ESIC method: there is no need to reconstruct the interface explicitly for building the liftings. This adds more generality to the method, as the level set can now be given in all the possible forms: even a level set given as a finite element function with high polynomial degree would be acceptable, as there is no need to solve non-linear equation for finding the zero level set. Moreover, if we use regularizing functions for the integration of discontinuous integrands, then the SESIC method treats the interface in a fully implicit way.

The numerical results show that in the 1D and 2D cases that we have tested, the method exhibits optimal orders of convergence. The use of regularized integrands rather than exact integration across the interface leads to a slower convergence in the regularization band around the interface, but it keeps the

optimal convergence in the remaining part of the domain, as shown on the 3D test case that we have considered.

**Acknowledgements.** The authors acknowledge the support of the Swiss National Science Foundation, Project Sinergia 125444: “Fluid dynamics and mixing behavior in orbitally shaken bioreactors for mammalian cell cultivation”. The authors would also like to warmly thank professors J. Huh and J. Sethian for the fruitful discussions and their relevant remarks on an early version of this work.

## References

- [1] D.N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344 (1985), 1984.
- [2] G. Caginalp and X. Chen. Phase field equations in the singular limit of sharp interface problems. In *On the Evolution of Phase Boundaries (Minneapolis, MN, 1990–91)*, volume 43 of *IMA Vol. Math. Appl.*, pages 1–27. Springer, New York, 1992.
- [3] A.H. Coppola-Owen and R. Codina. Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *Internat. J. Numer. Methods Fluids*, 49:1287–1304, 2005.
- [4] J.-S. Huh and J.A. Sethian. Exact subgrid interface correction schemes for elliptic interface problems. *Proceedings of the National Academy of Sciences of the United States of America*, 105(29):9874, 2008.
- [5] D.Q. Nguyen, R.P. Fedkiw, and M. Kang. A boundary condition capturing method for incompressible flame discontinuities. *J. Comput. Phys.*, 172:71–98, 2000.
- [6] N. Parolini. *Computational Fluid Dynamics for Naval Engineering Problems*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2004.
- [7] A. Quarteroni. *Numerical Models for Differential Problems*. Springer, Milan, 2009.
- [8] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, 1994.
- [9] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, second edition, 1999.
- [10] A.K. Tornberg. Multi-dimensional quadrature of singular and discontinuous functions. *BIT Numerical Mathematics*, 42(3):644–669, 2002.

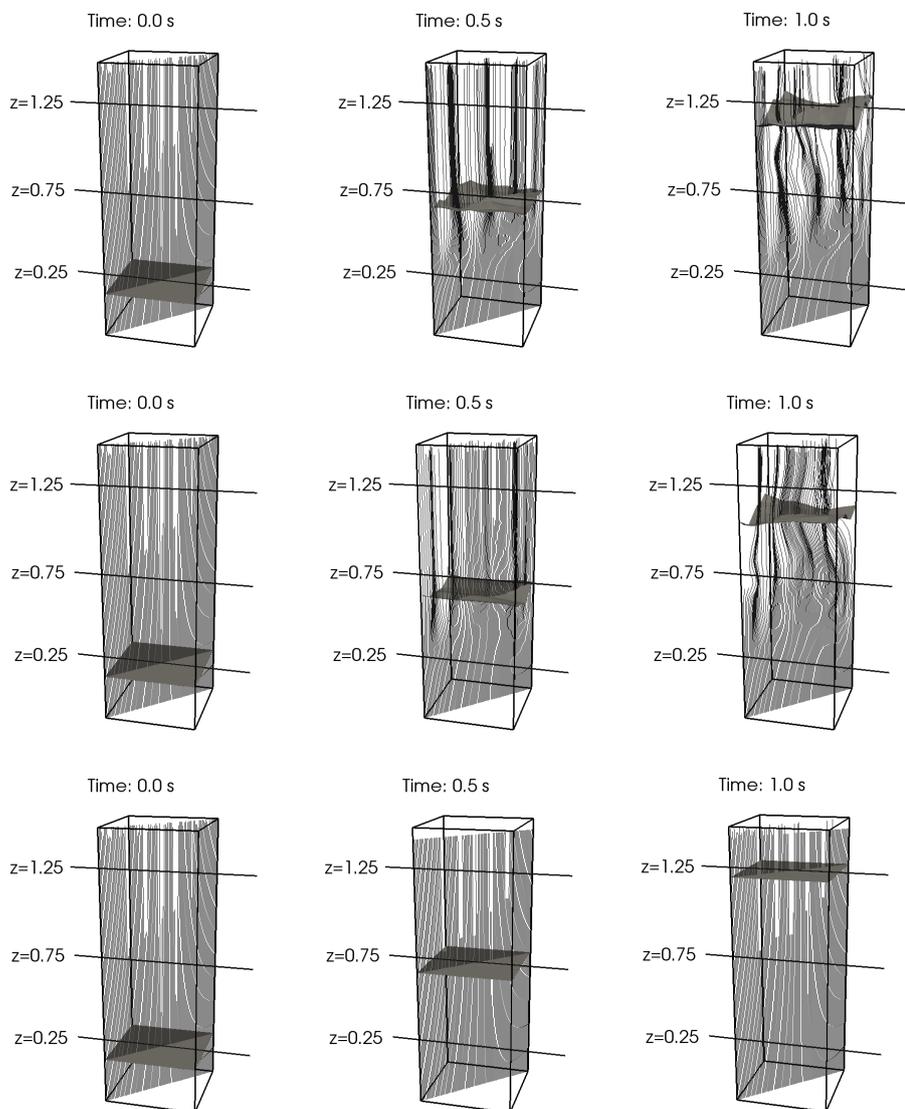


Figure 16: Streamlines of the solutions computed at different times for the benchmark problem: with standard numerical quadrature (top), with conforming quadrature (middle) and with conforming quadrature plus pressure correction (bottom).

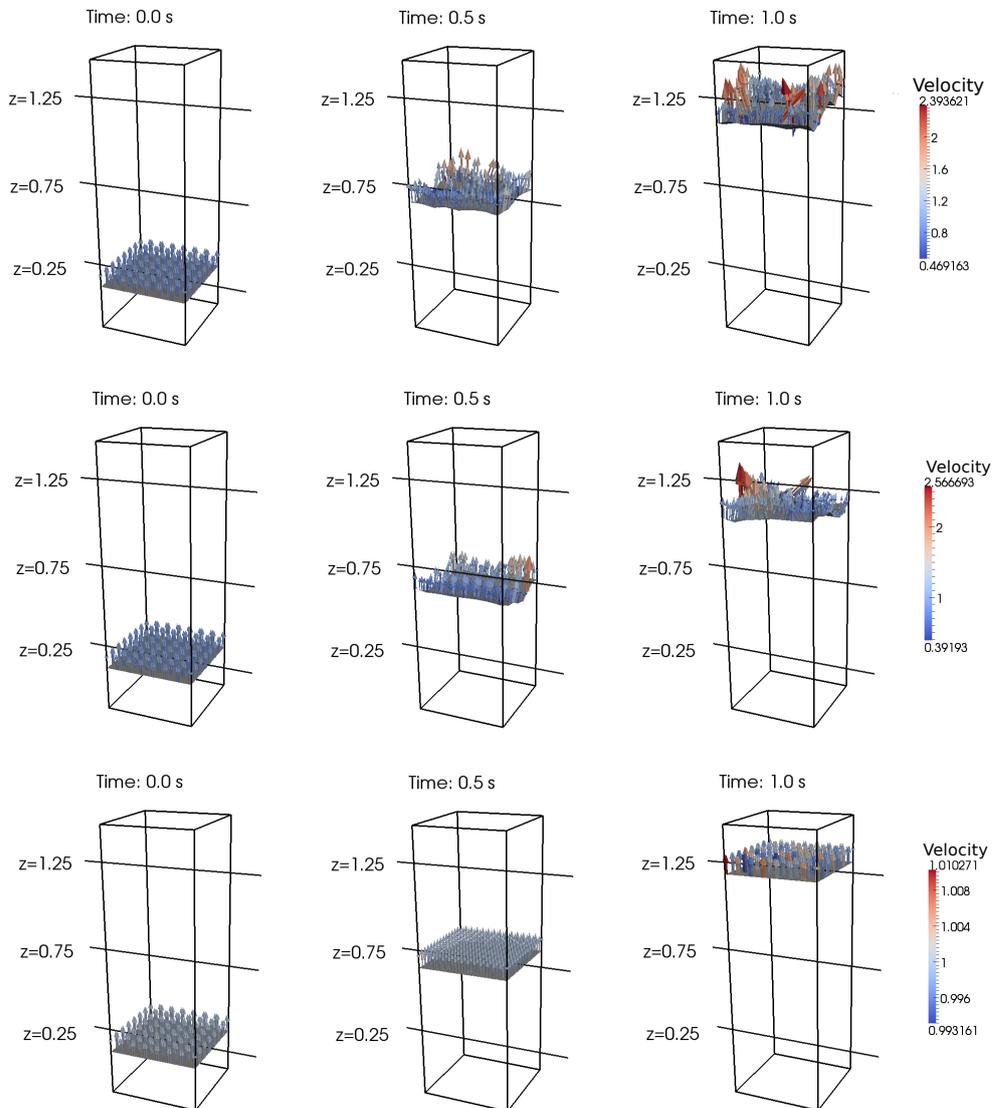


Figure 17: Free surface velocity for the solutions computed at different times for the benchmark problem: with standard numerical quadrature (top), with conforming quadrature (middle) and with conforming quadrature plus pressure correction (bottom).

# MOX Technical Reports, last issues

Dipartimento di Matematica “F. Brioschi”,  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 37/2010** MARCO DISCACCIATI, ALFIO QUARTERONI,  
SAMUEL QUINODOZ:  
*Numerical approximation of internal discontinuity interface problems*
- 36/2010** GIANNI ARIOLI, HANS KOCH:  
*Non-Symmetric low-index solutions for a symmetric boundary value problem*
- 35/2010** GIANNI ARIOLI, HANS KOCH:  
*Integration of dissipative PDEs: a case study*
- 34/2010** ANTONELLA ABBA', LUCA BONAVENTURA:  
*A mimetic finite difference method for Large Eddy Simulation of incompressible flow*
- 33/2010** GIOVANNI MIGLIORATI, ALFIO QUARTERONI:  
*Multilevel Schwarz Methods for Elliptic Partial Differential Equations*
- 32/2010** A. CRISTIANO I. MALOSSI, PABLO J. BLANCO,  
SIMONE DEPARIS, ALFIO QUARTERONI:  
*Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows*
- 31/2010** ANDREA MANZONI, ALFIO QUARTERONI, GIANLUIGI ROZZA:  
*Shape optimization for viscous flows by reduced basis methods and free-form deformation*
- 30/2010** PABLO J. BLANCO, MARCO DISCACCIATI,  
ALFIO QUARTERONI:  
*Modeling dimensionally-heterogeneous problems: analysis, approximation and applications*
- 29/2010** MATTEO LESINIGO, CARLO D'ANGELO,  
ALFIO QUARTERONI:  
*A multiscale Darcy-Brinkman model for fluid flow in fractured porous media*

**28/2010** PAOLO CROSETTO, PHILIPPE REYMOND, SIMONE DEPARIS,  
DIMITRIOS KONTAXAKIS, NIKOLAOS STERGIOPULOS,  
ALFIO QUARTERONI:  
*Fluid Structure Interaction Simulations of Physiological Blood Flow in  
the Aorta*