

An Adaptive Fictitious Domain Method for elliptic problems

S. Berrone, A. Bonito, M. Verani

Abstract In the Fictitious Domain Method with Lagrange multiplier (FDM) the physical domain is embedded into a simpler but larger domain called the fictitious domain. The partial differential equation is extended to the fictitious domain using a Lagrange multiplier to enforce the prescribed boundary conditions on the physical domain while all the other data are extended to the fictitious domain. This leads to a saddle point system coupling the Lagrange multiplier and the extended solution of the original problem. At the discrete level, the Lagrange multiplier is approximated on subdivisions of the physical boundary while the extended solution is approximated on partitions of the fictitious domain. A significant advantage of the FDM is that no conformity between these two meshes is required. However, a restrictive compatibility condition between the mesh-sizes must be enforced to ensure that the discrete saddle point system is well-posed. In this paper, we present an adaptive fictitious domain method (AFDM) for the solution of elliptic problems in two dimensions. The method hinges upon two modules `ELLIPTIC` and `ENRICH` which iteratively increase the resolutions of the approximation of the extended solution and the multiplier, respectively. The adaptive algorithm AFDM is convergent without any compatibility condition between the two discrete spaces. We provide numerical experiments illustrating the performances of the proposed algorithm.

1 Introduction

In many engineering applications the efficient numerical solution of partial differential equations on deformable or complex geometries is of paramount importance. In this respect, one crucial issue is the construction of the computational grid. To face this problem, one can basically resort to two different types of approaches. In the first approach, a mesh is constructed on a sufficiently accurate approximation of the

S. Berrone

Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129, Torino, Italy e-mail: tefano.berrone@polito.it

A. Bonito

Department of Mathematics, Texas A&M University, College Station, Texas 77843-3368, USA, e-mail: bonito@math.tamu.edu

M. Verani

MOX-Dipartimento di Matematica, Politecnico di Milano, P.zza L. da Vinci, I-20132 Milano, Italy, e-mail: marco.verani@polimi.it

exact physical domain (see, e.g., isoparametric finite elements [8], or isogeometric analysis [9]), while in the second approach one embeds the physical domain into a simpler computational mesh whose elements can intersect the boundary of the given domain. Clearly, the mesh generation process is extremely simplified in the second approach, while the imposition of boundary conditions requires extra work. Among the huge variety of methods sharing the philosophy of the second approach, let us mention here the Immersed Boundary methods (see, e.g., [16] and the references therein), the Penalty Methods (see, e.g., the seminal work [2]), the Fictitious Domain/Embedding Domain Methods (see, e.g., [7] and the references therein). In this paper, we focus on the Fictitious Domain Method with Lagrange multiplier (FDM) introduced in [12, 11] (see also [1] for the pioneering work inspiring this approach). In this approach, the physical domain ω is embedded into a simpler and larger domain Ω (the fictitious domain), the right-hand side is extended to the fictitious domain and the boundary conditions on the boundary of the physical domain are appended through the use of a Lagrange multiplier. The FDM gives rise to a saddle point problem whose exact solution restricted to ω corresponds to the solution of the original problem. At the discrete level, the FDM allows the use of structured and uniform meshes in the fictitious domain, without requiring any conformity between the bulk mesh and the boundary of the physical domain. This represents a relevant computational advantage. However, there are two important issues to be taken into account to build numerical techniques that are able to take advantage of the crucial features of FDM: the choice of the discrete spaces for the approximation of the solution and the multiplier, and the construction of the extension of the right-hand side from the physical domain to the fictitious one. As pointed out in the analysis performed in [11] in the context of finite element approximation for the solution of elliptic problems with Dirichlet boundary conditions, the first condition (*inf-sup* condition) is essential to ensure existence and uniqueness of the discrete solution, while the second one influences the regularity of the extended continuous solution, thus impacting on the approximation properties of the discrete spaces (on this latter topic, see, e.g., [13]). The first condition turns out to introduce some restrictive compatibility conditions between the mesh-sizes of the fictitious domain grid and the subdivision of boundary of the physical domain (needed to approximate the Lagrange multiplier). The second issue can spoil, for example, the performance of the linear finite element method on uniform meshes whenever the original solution is sufficiently regular, e.g. H^2 regular, while the extended solution is less regular.

In view of the above discussion, the computational effectivity of FDM seems to be a non trivial issue. However, as shown in the present paper, a judicious use of adaptivity can allow to overcome the above two obstructions and recover the full potentiality of FDM, i.e. working with discrete spaces (and meshes) violating the compatibility conditions and recover even in presence of less regular extended solutions, the optimal performance of finite elements on uniform grids in presence of regular solutions. In particular, in this work we present an adaptive fictitious domain method, named AFDM, based on the use of linear finite elements for the approximation of the solution and piecewise constants for the approximation of the Lagrange multiplier. In the spirit of the algorithm provided in [6], the method hinges upon two modules, `ELLIPTIC` and `ENRICH` that iteratively modify the discrete spaces for the approximation of the extended solution and the multiplier. Our method is proved to be convergent regardless of the imposition of any compatibility condition between the two discrete spaces. Similar remarks has been already pointed out in different contexts by [10, 4] (see also [3] for an abstract discussion of inexact Uzawa methods) and we refer to [14] for the mathematical study of an adaptive algorithm for the Stokes system, which serves as a benchmark for saddle point problems. Moreover, preliminary numerical results show that AFDM is optimal with respect to the number of degrees of freedom employed to approximate the extended solution and the Lagrange multiplier. In two dimension, the optimality of the adaptive refinement strategy seems to require an adaptive strategy to generate the successive subdivision of the fictitious domain, while uniform or quasi-uniform subdivisions can be used for the boundary mesh.

The outline of the paper is as follows. In Section 2 we introduce the fictitious domain method, while in Section 3 we introduce the adaptive fictitious domain method. Finally, in Section 4 we numerically explore the convergence (and optimality) properties of our algorithm.

2 Fictitious Domains Method

Let ω be a bounded domain of \mathbb{R}^2 with boundary γ . To make the presentation simpler, we assume that ω is a polygon. We are interested in employing the fictitious domain method to solve the following model problem: let $f \in L^2(\omega)$, find $u \in H_0^1(\omega)$ such that

$$-\Delta u = f \quad \text{in } \omega, \quad (1)$$

$$u = 0 \quad \text{on } \gamma. \quad (2)$$

The fictitious domain formulation of problem (1)-(2) hinges on a square or rectangular domain Ω with boundary $\Gamma := \partial\Omega$ and such that $\omega \subset \subset \Omega$. It reads: for any L^2 -extension \tilde{f} of f to Ω , find $(\tilde{u}, \lambda) \in H_0^1(\Omega) \times H^{-\frac{1}{2}}(\gamma)$ such that

$$\int_{\Omega} \nabla \tilde{u} \cdot \nabla v - \langle \lambda, v \rangle_{\gamma} = \int_{\Omega} \tilde{f} v \, dx \quad \forall v \in H_0^1(\Omega), \quad (3)$$

$$\langle \mu, \tilde{u} \rangle_{\gamma} = 0 \quad \forall \mu \in H^{-\frac{1}{2}}(\gamma), \quad (4)$$

where for $v \in H_0^1(\Omega)$, its restriction to γ is understood in the sense of traces, and $\langle \cdot, \cdot \rangle_{\gamma}$ denotes the duality pairing between $H^{-\frac{1}{2}}(\gamma)$ and $H^{\frac{1}{2}}(\gamma)$ (recall that γ is a closed curve). Using an integration by parts formula, it is immediate to verify (see, e.g., [12]), that the fictitious domain formulation (3)-(4) is equivalent to original formulation (1)-(2) where

$$\lambda = \left[\frac{\partial \tilde{u}}{\partial n} \right]_{\gamma} \quad (5)$$

is the jump of $\frac{\partial \tilde{u}}{\partial n}$ across γ and n denotes the unit normal exterior to ω .

The regularity of \tilde{u} and λ depends on the extension chosen for f and the domain ω . In the worst case, $\tilde{u} \in H^{\frac{3}{2}-\varepsilon}(\Omega)$ for any $\varepsilon > 0$ and $\lambda \in L^2(\gamma)$ satisfies $\lambda \in H^{\frac{1}{2}}(\gamma_i)$, for every straight line γ_i composing γ (see [11]). In what follows we will drop the symbol \sim if no confusion arises.

In [11], the Babuska-Brezzi's theory is used to guarantee that problem (3)-(4) is well posed. In particular, the bilinear form $(u, v) \mapsto \int_{\Omega} \nabla u \cdot \nabla v$ is coercive on the set $\{v \in H_0^1(\Omega) : \langle \mu, v \rangle_{\gamma} = 0 \, \forall \mu \in H^{-1/2}(\gamma)\}$ and that the following inf-sup condition holds: there exists a constant $\kappa > 0$ such that

$$\inf_{\mu \in H^{-\frac{1}{2}}(\gamma)} \sup_{v \in H_0^1(\Omega)} \frac{\langle v, \mu \rangle_{\gamma}}{\|\mu\|_{H^{-\frac{1}{2}}(\gamma)} \|v\|_{H^1(\Omega)}} \geq \kappa. \quad (6)$$

3 Adaptive Fictitious Domain Method

In this section we present our adaptive fictitious domain method (AFDM) based on Uzawa iterations. In Section 3.1, we describe the infinite dimensional version of the algorithm, whereas in Section 3.2 we introduce its adaptive finite dimensional counterpart.

3.1 Infinite dimensional fictitious domain algorithm

We start this section by describing the infinite dimensional fictitious domain algorithm for solving (3)-(4). It consists of Uzawa-type successive iterations: Given $\alpha > 0$ and $\lambda_0 \in H^{-\frac{1}{2}}(\gamma)$ we seek, for $j \geq 1$,

$$u_j \in H_0^1(\Omega) : \quad \int_{\Omega} \nabla u_j \cdot \nabla v = \int_{\Omega} f v + \langle \lambda_{j-1}, v \rangle_{\gamma} \quad \forall v \in H_0^1(\Omega), \quad (7)$$

$$\lambda_j \in H^{-\frac{1}{2}}(\gamma) : \quad (\lambda_j, \mu)_{\gamma} = (\lambda_{j-1}, \mu)_{\gamma} - \alpha (u_j, \mu)_{\gamma} \quad \forall \mu \in H^{-\frac{1}{2}}(\gamma) \quad (8)$$

where we denote by $(\cdot, \cdot)_{\gamma}$ the scalar product in $H^{-\frac{1}{2}}(\gamma)$ and where we used the identification of $L^2(\gamma)$ and have with a slight abuse of notation $H^{\frac{1}{2}}(\gamma) \subset H^{-\frac{1}{2}}(\gamma)$.

The Schur complement operator $S : H^{-\frac{1}{2}}(\gamma) \rightarrow H^{\frac{1}{2}}(\gamma) \subset H^{-\frac{1}{2}}(\gamma)$ defined as

$$S\lambda = u_{\lambda}, \quad (9)$$

where $u_{\lambda} \in H_0^1(\Omega)$ is the solution to

$$\int_{\Omega} \nabla u_{\lambda} \cdot \nabla v = \langle \lambda, v \rangle_{\gamma} \quad \forall v \in H_0^1(\Omega).$$

The operator S is symmetric and positive definite [12] and is instrumental in the analysis of the Uzawa iterations. In fact, (7)-(8) can be written using S as

$$\lambda_j = (I - \alpha S)\lambda_{j-1} + \alpha u_f \quad \text{in } H^{-\frac{1}{2}}(\gamma), \quad (10)$$

where $u_f \in H_0^1(\Omega)$ is given by

$$\int_{\Omega} \nabla u_f \cdot \nabla v = \int_{\Omega} f v \quad \forall v \in H_0^1(\Omega).$$

It is immediate to verify that if $0 < \alpha < 2/\|S\|_{\mathcal{L}(H^{-\frac{1}{2}}(\gamma), H^{-\frac{1}{2}}(\gamma))}$, then

$$\beta := \|I - \alpha S\|_{\mathcal{L}(H^{-\frac{1}{2}}(\gamma), H^{-\frac{1}{2}}(\gamma))} < 1 \quad (11)$$

and thus the infinite dimensional fictitious domain algorithm is convergent.

3.2 Adaptive finite dimensional fictitious domain method

We now introduce our adaptive finite dimensional fictitious domain algorithm (AFDM) which iteratively builds a sequence of nested finite dimensional spaces to achieve a reduction of the approximation error between each iterative step. We start with an initial conforming subdivision \mathcal{T}_0 of Ω made of triangles and an initial subdivision of γ (made of segments). We assume that

$$\text{for each } T \in \mathcal{T}_0, \hat{T} \cap \gamma \text{ is connected,} \quad (12)$$

which is automatically satisfied upon assuming that the initial subdivision \mathcal{T}_0 is sufficiently fine to capture the interface and could be enforced via uniform refinements without affecting the asymptotic performances of the algorithm. From now on, $j \geq 0$ will always denote the AFDM iteration counter. We denote by \mathcal{T}_j and \mathcal{S}_j the j -th conforming partitions of Ω and γ made of triangles and segments, respectively. The diameters of the elements $T \in \mathcal{T}_j$ and $\ell \in \mathcal{S}_j$ are denoted $h_T := \text{diam}(T)$ and $h_\ell := \text{diam}(\ell)$, respectively. We emphasize that the two partitions built by the adaptive algorithm described below are mutually independent and in particular no compatibility conditions between the two partitions is required. This is a crucial difference from the results in [11] (see Remark 2 below). The shape regularity constant of a generic subdivision \mathcal{T} is $\max_{T \in \mathcal{T}} \frac{h_T}{\rho_T}$, where ρ_T is the diameter of the largest ball inside T . The shape regularity constant of a sequence of subdivision $\{\mathcal{T}_i\}_{i \geq 0}$ is

$$\sup_{i \geq 0} \max_{T \in \mathcal{T}_i} \frac{h_T}{\rho_T}.$$

Associated with conforming partitions \mathcal{T} of Ω and \mathcal{S} of γ , we introduce the finite dimensional spaces

$$\mathbb{V}_{\mathcal{T}} := \{v \in C^0(\overline{\Omega}) : v|_T \in \mathcal{P}^1(T) \forall T \in \mathcal{T}_j\} \cap H_0^1(\Omega)$$

and

$$\mathbb{M}_{\mathcal{S}} := \{w \in L^2(\gamma) : w|_\ell \in \mathcal{P}^0(\ell) \forall \ell \in \mathcal{S}\},$$

where for $k \in \mathbb{N}$, $\mathcal{P}^k(D)$ is the set of polynomials of degree k in D . In the following, we set $\mathbb{V}_j := \mathbb{V}_{\mathcal{T}_j}$ and $\mathbb{M}_j := \mathbb{M}_{\mathcal{S}_j}$.

The finite dimensional adaptive fictitious domain algorithm relies on two sub-routines described now.

3.2.1 The module ELLIPTIC

The module ELLIPTIC adaptively constructs approximations U_j of the exact solution u_j to (7). To describe this procedure, we let $u_j \in H_0^1(\Omega)$ satisfying

$$\int_{\Omega} \nabla u_j \cdot \nabla v = \int_{\Omega} f v + \int_{\gamma} \Lambda_{j-1} v \quad \forall v \in H_0^1(\Omega), \quad (13)$$

for a given $\Lambda_{j-1} \in \mathbb{M}_{j-1}$.

In contrast to (7) where $\lambda_{j-1} \in H^{-\frac{1}{2}}(\gamma)$, we note that Λ_{j-1} belongs to a finite dimensional subspace of $L^2(\gamma)$. Moreover, we observe that (13) is a weak formulation of

$$\begin{aligned}
-\Delta u_j &= f \quad \text{in } \omega, & -\Delta u_j &= f \quad \text{in } \Omega \setminus \overline{\omega}, \\
[u_j] &= 0 \quad \text{on } \gamma, & \left[\frac{\partial u_j}{\partial n} \right]_\gamma &= \Lambda_{j-1} \quad \text{on } \gamma, \\
u_j &= 0 \quad \text{on } \partial\Omega.
\end{aligned} \tag{14}$$

If ε_j stands for an adjustable error tolerance, then the module `ELLIPTIC`,

$$(\mathcal{T}_j, U_j) = \text{ELLIPTIC}(\mathcal{T}_{j-1}, \Lambda_{j-1}, \varepsilon_j),$$

constructs adaptively a refined mesh \mathcal{T}_j of \mathcal{T}_{j-1} such that the solution of the discrete elliptic problem

$$U_j \in \mathbb{V}_j : \quad \int_{\Omega} \nabla U_j \cdot \nabla V = \int_{\Omega} f V + \int_{\gamma} \Lambda_{j-1} V \quad \forall V \in \mathbb{V}_j, \tag{15}$$

approximate u_j within in the prescribed tolerance ε_j , i.e.

$$\|\nabla(u_j - U_j)\|_{L^2(\Omega)} \leq \varepsilon_j. \tag{16}$$

The adaptive `ELLIPTIC` module iterates a classical strategy of the type

$$\text{SOLVE} - \text{ESTIMATE} - \text{MARK} - \text{REFINE}$$

until condition (16) is satisfied (see, e.g., [15]).

The following upper bound for the error of any inner-iterate is instrumental in `ESTIMATE`. A corresponding lower bound is also valid and we refer to [5] for their proofs.

Proposition 3.1 (Upper bound for `ELLIPTIC`) *Let $u_j \in H_0^1(\Omega)$ be the solution to (13) and $U_j^i \in \mathbb{V}_i$ be the discrete Galerkin solution to (15) associated with any refinement \mathcal{T}^i of \mathcal{T}_{j-1} . Assume that the initial subdivision satisfy (12). Then, there exists a positive constant K^* only depending on \mathcal{T}^i through its shape-regularity constant such that the following a posteriori error estimate holds*

$$\|\nabla(u_j - U_j^i)\|_{L^2(\Omega)}^2 \leq K^* \sum_{T \in \mathcal{T}^i} \eta^i(T)^2 \tag{17}$$

where

$$\eta^i(T)^2 := h_T^2 \|R_T^i\|_{L^2(T)}^2 + h_T \sum_{e \text{ edge of } T} \|J_e^i\|_{L^2(e)}^2 + h_T \|\Lambda_{j-1}\|_{L^2(\hat{T} \cap \gamma)}^2,$$

with

$$R_T^i := (f + \Delta U_j^i)|_T, \quad J_e^i := \begin{cases} \left[\frac{\partial U_j^i}{\partial n} \right]_e - \Lambda_{j-1} & \text{on } e \cap \gamma, \\ \left[\frac{\partial U_j^i}{\partial n} \right]_e & \text{on } e \setminus \gamma. \end{cases} \tag{18}$$

Here $[\cdot]_e$ is the jump across the edge e and \hat{T} is the topological interior of T .

Remark 1 (Elliptic Estimator). We note that the non-standard terms containing Λ_{j-1} in the estimator $\eta^i(T)$ above both measure the discrepancy from the exact relation $\lambda = [\frac{\partial u}{\partial n}]_\gamma$, see (5). In particular, within an element T , ∇U_j^i is continuous so that

$$\|\Lambda_{j-1}\|_{L^2(\hat{T} \cap \gamma)} = \left\| \left[\frac{\partial U_j^i}{\partial m} \right]_{\hat{T} \cap \gamma} - \Lambda_{j-1} \right\|_{L^2(\hat{T} \cap \gamma)},$$

where $\left[\frac{\partial U_j^i}{\partial m} \right]_{\hat{T} \cap \gamma}$ denotes the jump of $\frac{\partial U_j^i}{\partial m}$ across $\hat{T} \cap \gamma$ and m denotes one of the normal to $\hat{T} \cap \gamma$.

3.3 The module ENRICH

Let \mathbb{V}_j^γ be the restriction of functions in \mathbb{V}_j . We denote by $\Pi_j : \mathbb{V}_j^\gamma \rightarrow \mathbb{M}_j$ the orthogonal L^2 -projection onto \mathbb{M}_j . The module

$$\mathcal{S}_j = \text{ENRICH}(\mathcal{S}_{j-1}, U_j, \varepsilon_j), \quad (19)$$

constructs a refinement \mathcal{S}_j of \mathcal{S}_{j-1} such that

$$\|(I - \Pi_j)U_j\|_{H^{-\frac{1}{2}}(\gamma)} \leq C\varepsilon_j \quad (20)$$

for a given C independent of j .

The quantity $\|(I - \Pi_j)U_j\|_{H^{-\frac{1}{2}}(\gamma)}$ is not computed exactly but estimated as follows. Standard interpolation error estimates together with a trace estimate and the stability of the L^2 -projection yield

$$\begin{aligned} \|(I - \Pi_j)U_j\|_{H^{-1/2}(\gamma)} &= \sup_{\phi \in H^{1/2}(\gamma)} \frac{\int_\gamma (I - \Pi_j)U_j \phi}{\|\phi\|_{H^{1/2}(\gamma)}} = \sup_{\phi \in H^{1/2}(\gamma)} \frac{\int_\gamma (I - \Pi_j)U_j (\phi - \Phi)}{\|\phi\|_{H^{1/2}(\gamma)}} \\ &\leq \sup_{\phi \in H^{1/2}(\gamma)} \frac{\|h_\ell^{1/2}(I - \Pi_j)U_j\|_{L^2(\gamma)} \|h_\ell^{-1/2}(\phi - \Phi)\|_{L^2(\gamma)}}{\|\phi\|_{H^{1/2}(\gamma)}} \\ &\leq C \|h_\ell^{1/2}(I - \Pi_j)U_j\|_{L^2(\gamma)} \leq C (\max_{\ell \in \mathcal{S}_j} h_\ell)^{1/2} \|(I - \Pi_j)U_j\|_{L^2(\gamma)} \\ &\leq C (\max_{\ell \in \mathcal{S}_j} h_\ell) \|U_j\|_{H^{1/2}(\gamma)} \leq C (\max_{\ell} h_\ell) \|U_j\|_{H^1(\Omega)}, \end{aligned}$$

where $\Phi \in \mathbb{M}_j$ and C is a generic constant independent of j . Now the stability estimate $\|U_j\|_{H^1(\Omega)} \leq C\|f\|_{L^2(\Omega)}$ imply

$$\|(I - \Pi_j)U_j\|_{H^{-1/2}(\gamma)} \leq C (\max_{\ell} h_\ell) \|f\|_{L^2(\Omega)}. \quad (21)$$

Based on this, the ENRICH routine refines recursively all the elements of \mathcal{S}_{j-1} until they all have a mesh-size smaller than $\frac{\varepsilon_j}{\|f\|_{L^2(\Omega)}}$. The resulting mesh is output of ENRICH since it satisfies (20).

3.4 The module UPDATE

The discrete Lagrange multiplier is updated by the module UPDATE

$$\Lambda_j = \text{UPDATE}(\mathcal{T}_j, \mathcal{S}_j, \Lambda_{j-1}, U_j, \alpha), \quad (22)$$

which computes according to (8)

$$\Lambda_j \in \mathbb{M}_j : \quad \Lambda_j = \Lambda_{j-1} - \alpha \Pi_j U_j . \quad (23)$$

We note that as $U_j \in \mathbb{V}_j$ its restriction to γ is an element of \mathbb{V}_j^γ (by construction).

3.5 The AFDM algorithm

We are now in a position to detail the iterative structure of AFDM. Each iteration of the algorithm consists of an inner solver employing `ELLIPTIC` in place of (7), followed by an application of the module `ENRICH` and by an update of the multiplier performed by the module `UPDATE`.

ADAPTIVE FICTITIOUS DOMAIN METHOD (AFDM)

Given the initial triangulations \mathcal{T}_0 and \mathcal{S}_0 , and the parameters α , $\varepsilon_0 > 0$, $0 < \zeta < 1$ set $j=1$ and iterate:

1. Select any function $\Lambda_0 \in \mathbb{M}_0$.
2. Update $\varepsilon_j \leftarrow \zeta \varepsilon_{j-1}$.
3. Compute $(\mathcal{T}_j, U_j) = \text{ELLIPTIC}(\mathcal{T}_{j-1}, \Lambda_{j-1}, \varepsilon_j)$.
4. Enrich $\mathcal{S}_j = \text{ENRICH}(\mathcal{S}_{j-1}, U_j, \varepsilon_j)$.
5. Compute $\Lambda_j = \text{UPDATE}(\mathcal{T}_j, \mathcal{S}_j, \Lambda_{j-1}, U_j, \alpha)$.
6. Update $j \leftarrow j+1$.
7. Go to step (2).

The algorithm AFDM is convergent [5] as reported in the theorem below.

Theorem 1 (Convergence). *Let $\alpha > 0$ be such that (11) holds and assume that the initial subdivision satisfy (12). Let $(\mathcal{T}_j, \mathcal{S}_j, U_j, \Lambda_j)$ be the sequence of meshes and subdivision produced by AFDM. There exist positive constants C_1 and $\delta < 1$ depending on the shape regularity constant of $\{\mathcal{T}_j\}_{j \geq 0}$ such that*

$$\|\nabla(u - U_j)\|_{L^2(\Omega)} + \|\lambda - \Lambda_j\|_{H^{-\frac{1}{2}}(\gamma)} \leq C_1 \delta^j . \quad (24)$$

Remark 2 (Compatibility condition). In [11] the authors obtain a priori error estimates for the finite element approximation of (3)-(4) under the assumption that the mesh size of the bulk triangulation is sufficiently large compared to the mesh size of the boundary triangulation, i.e. there holds a compatibility condition between the discrete spaces. This latter is a crucial requirement to prove the validity of a discrete inf-sup condition and thus the existence of the discrete solution. We emphasize that the convergence of our adaptive algorithm holds without enforcing any compatibility condition, thus possibly violating, the discrete inf-sup condition (see [10, 4] for similar results in different contexts).

The algorithm AFDM described above never ends until a stopping criterion is appended. In principle, we can resort to (24) to obtain an a priori estimate for the necessary number of iterations to reach a prescribed tolerance. However, this strategy is not implementable as the constant δ appearing in (24) is not accessible in practice. For this reason, we now provide an a-posteriori error estimate for the quantity $\|\nabla(u - U_j)\|_{L^2(\Omega)} + \|\lambda - \Lambda_{j-1}\|_{H^{-\frac{1}{2}}(\gamma)}$ which can be employed to stop AFDM while guarantying a prescribed approximation error.

Proposition 3.2 (Upper bound for AFDM) *Let $(u, \lambda) \in H_0^1(\Omega) \times H^{-\frac{1}{2}}(\gamma)$ be the solution to (3)-(4) and $\{(U_j, \Lambda_{j-1})\}$ be the sequence of approximations produced by AFDM. Assume that the initial subdivision satisfy (12). Then there exists a constant C_2 depending on the shape regularity constant of \mathcal{T}_j such that*

$$\|\nabla(u - U_j)\|_{L^2(\Omega)} + \|\lambda - \Lambda_{j-1}\|_{H^{-\frac{1}{2}}(\gamma)} \leq C_2 \left(\eta_{\mathcal{T}_j} + \eta_{\mathcal{T}_{j-1}} \right), \quad (25)$$

where

$$\eta_{\mathcal{T}_j} := \left(\sum_{T \in \mathcal{T}_j} \eta_j(T)^2 \right)^{1/2} \quad (26)$$

$$\eta_{\mathcal{T}_{j-1}} := \left(\sum_{\ell \in \mathcal{T}_{j-1}} h_\ell \|\nabla_\gamma U_j\|_{L^2(\ell)}^2 \right)^{1/2} \quad (27)$$

and

$$\eta_j(T)^2 := h_T^2 \|R_{T,j}\|_{L^2(T)}^2 + h_T \sum_{e \text{ edge in } T} \|J_{e,j}\|_{L^2(e)}^2 + h_T \|\Lambda_{j-1}\|_{L^2(\hat{T} \cap \gamma)}^2,$$

with

$$R_{T,j} := (f + \Delta U_j)|_T, \quad J_{e,j} := \begin{cases} [\frac{\partial U_j}{\partial n}]_e - \Lambda_{j-1} & \text{on } e \cap \gamma, \\ [\frac{\partial U_j}{\partial n}]_e & \text{on } e \setminus \gamma. \end{cases} \quad (28)$$

Remark 3 (Boundary indicator). In view of the results contained in [17], the indicator $\eta_{\mathcal{T}_{j-1}}$ measures the mismatch between the trace of U_j on γ and the exact homogeneous Dirichlet boundary condition (2).

4 Numerical Results

In this section we illustrate numerically the convergence properties of the AFDM algorithm and investigate numerically its optimality studied in [5]. Before presenting the numerical results we discuss some details regarding the implementation of AFDM.

4.1 Implementation issues

Data structures for representing the (two dimensional) triangular bulk mesh of the fictitious domain Ω and the (one dimensional) boundary mesh of γ are organized as binary trees starting from the 0-th level meshes \mathcal{T}_0 and \mathcal{S}_0 , respectively. The initial bulk mesh \mathcal{T}_0 is a regular mesh of the domain Ω constructed ignoring the conformity with γ . The initial boundary mesh \mathcal{S}_0 is made of the edges of γ , which is assumed to be a polygonal curve, see Figure 4.1(left). The refinement of the triangular bulk mesh is performed by employing the longest edge splitting. The refined elements are labeled as non-active and the newly created elements become active. Notice that additional refinements of neighboring elements may be needed in order to ensure conformity.

The refinement strategy advocated for the boundary mesh is more involved. On each boundary edge, we consider the points given by the intersection between the edge itself and the triangular mesh elements. We refer to these points as to boundary points. Whenever a refinement of a boundary edge is needed, we split the edge at the closest boundary point to the edge mid-point, see Figure 4.1(right). In Figure 4.1(bottom-left) we sketch 3 successive refinement of the horizontal portion of γ . In Figure 4.1(top-left), we report the meshes resulting from the first four steps of a uniform refinement of the boundary elements. In case no boundary nodes are available, we perform successive refinements of the triangle containing the edge until one boundary nodes becomes available. However the two children produced by refinement of the triangle are labeled as non-active unless needed by the bulk mesh refinement procedure.

In our implementation the elements of the mesh \mathcal{S} are always determined by the intersection between the boundary γ and a triangle (not necessarily active) belonging to the infinite binary tree with root \mathcal{T}_0 . In view of this construction, the levels of refinement to which the elements in \mathcal{S} and in \mathcal{T} belong to are completely independent. Clearly, the above construction of \mathcal{S} adds some geometric restrictions on the set of meshes we can deal with. However, the advantage at the computational level is quite remarkable because all the computer operations required by the refining process, as well as all the numerical computation of integrals involving the interaction between bulk and boundary objects (mesh elements or functions) can be always performed at the level of the single element (or of the binary tree stemming from it) *without* affecting the binary data structure of the neighboring elements.

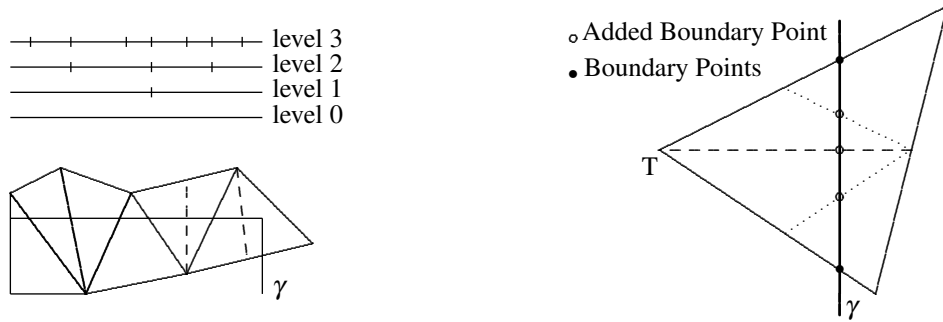


Fig. 1 Example of boundary refinement: (top-left) four different levels of refinement of the boundary mesh and (bottom-left) associated bulk mesh with local refinement (dotted line); (right) boundary points produced by the refinement of the bulk elements.

For the sake of presentation in the rest of this section we employ the following notation. We set $e_j^u = u - U_j$ and $e_j^\lambda = \lambda - \Lambda_j$. Moreover, we denote by $\|\cdot\|_0$ the $L^2(\Omega)$ or $L^2(\gamma)$ -norm depending on the context and by $|\cdot|_1$ the $H^1(\Omega)$ -seminorm.

4.2 Test case: L-shape domain

We consider the L-shaped domain $\omega = (-1, 1)^2 \setminus (-1, 0)^2$ with boundary γ . We are interested in the following model problem: we choose $f \in L^2(\omega)$ such that, after introducing polar coordinates (r, ϕ) , the solution to (1)-(2) is

$$u(r, \phi) = h(r)r^{2/3} \sin(2/3(\phi + \pi/2))$$

where

$$h(r) = \frac{w(3/4 - r)}{w(r - 1/4) + w(3/4 - r)}, \quad w(r) = \begin{cases} r^2 & \text{if } r > 0 \\ 0 & \text{else.} \end{cases}$$

The fictitious domain formulation of problem (3)-(4) is obtained by embedding ω in the square domain $\Omega = (-1, 1)^2$ with boundary $\Gamma = \partial\Omega$, see Figure 2, and extend f by zero outside ω . It is not difficult to see that $u \in H^{\frac{5}{3}-\varepsilon}(\omega)$, for any $\varepsilon > 0$ and the exact Lagrange multiplier is

$$\lambda = \frac{2}{3} h(r) r^{-1/3}.$$

In the following, we explore the convergence and optimality properties of AFDM algorithm. The AFDM algorithm is applied with the following parameters: $\alpha = 0.5$, $\zeta = 0.95$ and different values of ε_0 , namely $\varepsilon_0 = 1.0, 0.5, 0.25, 0.1$ (see Section 3 for the precise meaning of the parameters). In the sequel, we report the results obtained by AFDM. The outer iteration of AFDM is stopped when

$$\eta_{\mathcal{T}_j} + \eta_{\mathcal{S}_{j-1}} < \zeta^{45} \varepsilon_0.$$

In Figure 2 we display the initial and the final mesh together with two intermediate adaptively refined meshes, while in Figure 3 (left) we report the final discrete solution U plotted on the fictitious domain and in Figure 3 (right) we collect the graphs of final discrete Lagrange multiplier Λ together with the exact multiplier λ and the restriction of U on the boundary γ . A close inspection of the figures reveals that the algorithm AFDM correctly approximates the exact pair (u, λ) . As the value of β is unknown, an a priori choice for ξ is impossible. However, as already observed in [4], a practical choice for ξ seems to be $\xi = 0.95$. Indeed, larger values of ξ clearly guarantee the boundedness of the inner iterations of ELLIPTIC, but at the expense of an increased number of outer iterations. Values of ξ close to 0.95 ensure an optimal decay rate of the H^1 -error. This motivates the initial choice of setting $\xi = 0.95$ for running the numerical tests.

	$\ e_j^u\ _0$	$ e_j^u _1$	$\ e_{j-1}^\lambda\ _{-\frac{1}{2}, j-1}$	$\eta_{\mathcal{T}_j}$	$\eta_{\mathcal{S}_{j-1}}$
$\varepsilon_0=1$	-0.9073	-0.5257	-0.4279	-0.5201	-0.6892
$\varepsilon_0=0.5$	-0.9347	-0.5447	-0.4232	-0.5080	-0.7072
$\varepsilon_0=0.25$	-0.9019	-0.5368	-0.4104	-0.5062	-0.7491
$\varepsilon_0=0.1$	-0.8819	-0.5584	-0.3924	-0.4980	-0.7966

Table 1 Computed rates of convergence of the true errors and of the error estimators $\eta_{\mathcal{T}_j}$ and $\eta_{\mathcal{S}_{j-1}}$ with respect to the total number of dofs $\#\mathcal{T}_j + \#\mathcal{S}_j$ for different values of ε_0 .

	$\#\mathcal{T}_j + \#\mathcal{S}_j$	$\#\mathcal{T}_j$	$\#\mathcal{S}_j$
$\varepsilon_0=1$	-1.8951	-1.9415	-0.9895
$\varepsilon_0=0.5$	-1.9326	-1.9595	-0.9928
$\varepsilon_0=0.25$	-2.0156	-2.0319	-1.0240
$\varepsilon_0=0.1$	-2.2928	-2.3096	-1.0831

Table 2 Growing rate r of the total number of dofs (second column), bulk dofs (third) and boundary dofs (fourth) with respect to ε_j , i.e. r such that $\#(\cdot)_j \in \mathcal{O}(\varepsilon_j^r)$.

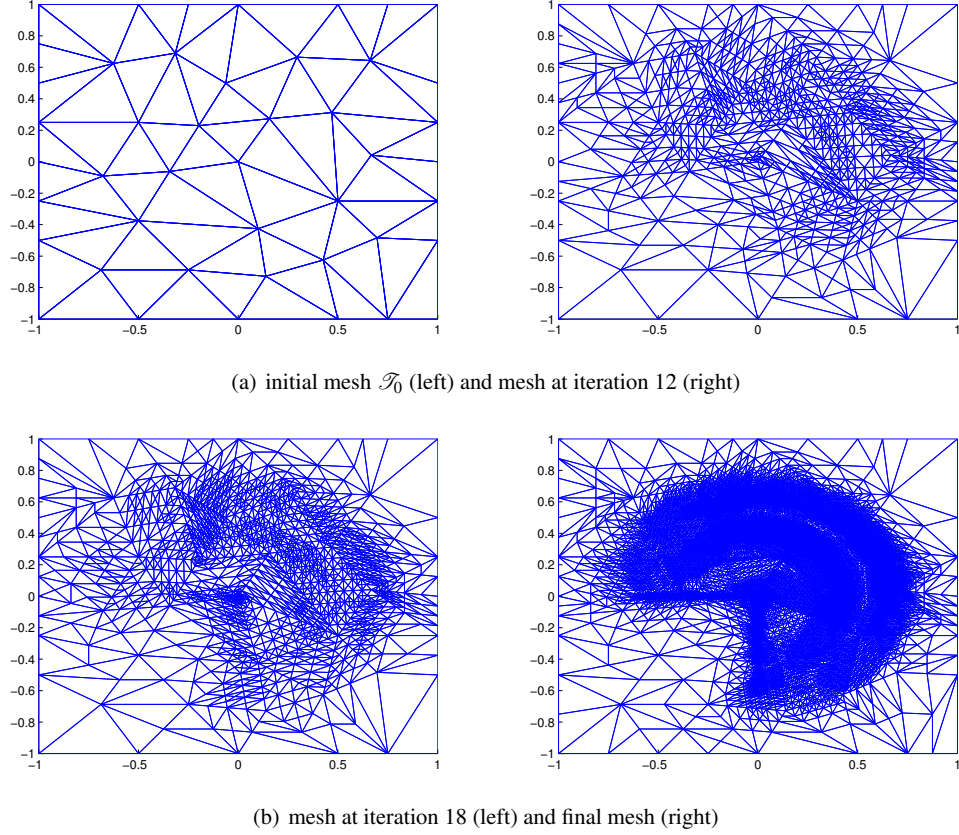


Fig. 2 Adaptive meshes produced by AFDM.

In the following, we further explore the optimality properties of AFDM and see their dependency on the parameter ε_0 . In particular, in Table 1 we report the rates of convergence with respect to the total number of degrees of freedom, of the true errors for e_j^u in $L^2(\Omega)$, $H^1(\Omega)$, the error for e_j^λ in an approximate $H^{-\frac{1}{2}}(\gamma)$ computed as a weighted L^2 norm

$$\|e_j^\lambda\|_{-\frac{1}{2}} \approx \|e_j^\lambda\|_{-\frac{1}{2},j} := \left(\sum_{\ell \in \mathcal{S}_j} h_\ell \|\lambda - \Lambda_j\|_{L^2(\ell)}^2 \right)^{1/2},$$

and the error estimators $\eta_{\mathcal{T}_j}$ and $\eta_{\mathcal{S}_{j-1}}$.

From the third and fourth columns we infer that the total error $\|\nabla(u - U_j)\|_{L^2(\Omega)} + \|\lambda - \Lambda_{j-1}\|_{H^{-\frac{1}{2}}(\gamma)}$ decreases to zero roughly as $(\#\mathcal{T}_j + \#\mathcal{S}_j)^{-0.5}$, the optimal decay for piecewise approximations of u . This is corroborated by the results in columns five and six which are thus in agreement with Proposition 3.2.

In Table 2 we collect the growing rates of the total number of dofs $\#\mathcal{T}_j + \#\mathcal{S}_j$, of the bulk dofs $\#\mathcal{T}_j$ and of the boundary dofs $\#\mathcal{S}_j$. The results shows that optimal convergence is dictated by the regularity of the

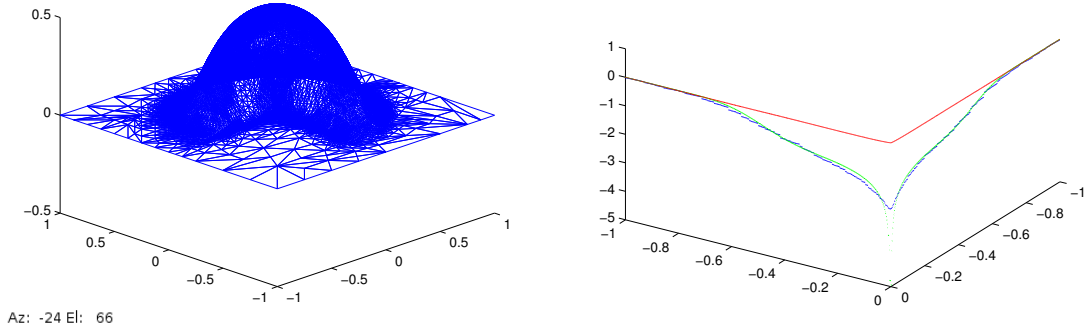


Fig. 3 Left: discrete solution U in the fictitious domain. Right: zoom on γ at the reentrant corner of the discrete solutions U (red), A (blue) and exact multiplier λ (green).

original non-extended solution u . Finally, collecting the results of Tables 1-2, we realize that both the total error and the error indicator $\eta_{\mathcal{T}_j} + \eta_{\mathcal{S}_j}$ decay as ε_j .

5 Conclusions

We introduced an Adaptive Fictitious Domain Method (AFDM). The core of the method is based on two modules, `ELLIPTIC` and `ENRICH` that iteratively modify the discrete spaces for the approximation of the extended solution and the multiplier, respectively. Numerical results show that AFDM is convergent, regardless of the imposition of any compatibility condition between the two discrete space. Moreover, preliminary tests seem to suggest the optimal behaviour of AFDM. This last topic requires further investigations, which is the topic of [5].

Acknowledgements This work has been partially supported by the Italian MIUR through PRIN research grant 2012HBLYE4_001 “Metodologie innovative nella modellistica differenziale numerica”, by INdAM-GNCS and by the National Science Foundation grant DMS-1254618.

References

1. I. Babuška. The finite element method with Lagrangian multipliers. *Numer. Math.*, 20:179–192, 1972/73.
2. I. Babuška. The finite element method with penalty. *Math. Comp.*, 27:221–228, 1973.
3. C. Bacuta. A unified approach for Uzawa algorithms. *SIAM J. Numer. Anal.*, 44(6):2633–2649, 2006.
4. E. Bänsch, P. Morin, and R. H. Nochetto. An adaptive Uzawa FEM for the Stokes problem: convergence without the inf-sup condition. *SIAM J. Numer. Anal.*, 40(4):1207–1229, 2002.
5. S. Berrone, A. Bonito, and M. Verani. An adaptive fictitious domain method for elliptic problems: convergence and optimality. in preparation, 2016.

6. A. Bonito, R. A. DeVore, and R. H. Nochetto. Adaptive finite element methods for elliptic problems with discontinuous coefficients. *SIAM J. Numer. Anal.*, 51(6):3106–3134, 2013.
7. C. Börgers and O. B. Widlund. On finite element domain imbedding methods. *SIAM J. Numer. Anal.*, 27(4):963–978, 1990.
8. P. G. Ciarlet. *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. Reprint of the 1978 original [North-Holland, Amsterdam; MR0520174 (58 #25001)].
9. J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.
10. S. Dahlke, W. Dahmen, and K. Urban. Adaptive wavelet methods for saddle point problems—optimal convergence rates. *SIAM J. Numer. Anal.*, 40(4):1230–1262, 2002.
11. V. Girault and R. Glowinski. Error analysis of a fictitious domain method applied to a Dirichlet problem. *Japan J. Indust. Appl. Math.*, 12(3):487–514, 1995.
12. R. Glowinski, T.-W. Pan, and J. Périaux. A fictitious domain method for Dirichlet problem and applications. *Comput. Methods Appl. Mech. Engrg.*, 111(3-4):283–303, 1994.
13. M. S. Mommer. A smoothness preserving fictitious domain method for elliptic boundary-value problems. *IMA J. Numer. Anal.*, 26(3):503–524, 2006.
14. M. S. Mommer and R. Stevenson. A goal-oriented adaptive finite element method with convergence rates. *SIAM J. Numer. Anal.*, 47(2):861–886, 2009.
15. R. H. Nochetto and A. Veiser. Primer of adaptive finite element methods. In *Multiscale and adaptivity: modeling, numerics and applications*, volume 2040 of *Lecture Notes in Math.*, pages 125–225. Springer, Heidelberg, 2012.
16. C. S. Peskin. The immersed boundary method. *Acta Numer.*, 11:479–517, 2002.
17. R. Sacchi and A. Veiser. Locally efficient and reliable a posteriori error estimators for Dirichlet problems. *Math. Models Methods Appl. Sci.*, 16(3):319–346, 2006.