A LOCAL ADAPTIVE REFINEMENT METHOD WITH MULTIGRID PRECONDITIONNING ILLUSTRATED BY MULTIPHASE FLOWS SIMULATIONS.

Franck Boyer[1], Celine Lapuerta[2], Sebastian Minjeaud[2] and Bruno Piar[2]

Abstract. The aim of this paper is to describe some numerical aspects linked to incompressible three-phase flow simulations, thanks to Cahn-Hilliard type model. The numerical capture of transfer phenomenon in the neighborhood of the interface require a mesh thickness which become crippling in the case where it is applied to the whole computational domain. This suggests the use of a local refinement method which allows to dynamically focus on problematic areas. The notion of refinement pattern, introduced for Lagrange finite elements, allows to build a conceptual hierarchy of nested conformal approximation spaces which is then used to implement the so-called CHARMS local refinement methods. Properties of these methods are proved ensuring in particular the conformity of approximation spaces at every time of simulations. Furthermore, the multilevel structure obtained by this method, is used to construct multigrid preconditioners. Finally, after a validation on a model problem, the performance of the whole method is illustrated on an example of a liquid lens spreading between two stratified fluids.

Résumé. L'objectif de l'article est de décrire certains aspects numériques liés à la simulation d'écoulements incompressibles à trois phases non miscibles, à l'aide de modèles à interfaces diffuses de type Cahn-Hilliard. La capture numérique des phénomènes de transfert au voisinage des interfaces requiert une finesse de maillage qui devient rédhibitoire si elle est appliquée à l'ensemble du domaine de calcul. Ceci suggère l'utilisation de méthodes de raffinement local adaptatif qui permettent de se focaliser dynamiquement sur les zones sensibles. La notion de motif de raffinement, introduite pour des éléments finis de Lagrange, permet de construire une hiérarchie conceptuelle d'espaces d'approximation conformes emboîtés qui est alors utilisée pour mettre en oeuvre les techniques de raffinement local dites CHARMS. Les propriétés de la méthode sont prouvées assurant en particulier la conformité de l'espace d'approximation à tout instant des simulations. En outre, la structure multiniveaux obtenue par cette méthode est exploitée pour construire des préconditionneurs multigrilles. Enfin, après une validation sur un problème modèle, les performances de l'ensemble sont illustrées sur un exemple d'étalement d'une lentille piégée entre deux phases stratifiées.

[1] Laboratoire d'Analyse, Topologie et Probabilités (LATP), Faculté des Sciences et Techniques de Saint-Jérôme - Case Cour A, Université Paul Cézanne, Avenue Escadrille Normandie-Niemen, 13397 MARSEILLE Cedex 20, France.
e-mail: `fboyer@cmi.univ-mrs.fr`

[2] Institut de Radioprotection et de Sûreté Nucléaire (IRSN), DPAM/SEMIC/LIMSI, BP-3, 13115 Saint-Paul-lez-Durance, France.
e-mail: `celine.lapuerta@irsn.fr` & `sebastian.minjeaud@irsn.fr` & `bruno.piar@irsn.fr`

Contents

Introduction

During a hypothetical major accident in a nuclear reactor, the degradation of the core may produce multiphase flows where interfaces undergo extreme topological changes, e.g. break-up and coalescence. Because of their ability to capture interfaces implicitly, phase field models are attractive for the numerical simulation of such phenomena [5]. They replace sharp interfaces by thin but nonzero thickness transition regions where capillary effects and changes in fluid properties are smoothed. The implied burden on the numerical method lies in the required resolution of very thin moving internal layers. We present here a general adaptive local refinement strategy and an associated multigrid solver in order to tackle such issues. We illustrate the method on the ternary Cahn-Hilliard model taken from [5].

The challenge of local refinement can be expressed in that way: increase the spatial resolution of some part of the domain which are dynamically selected at each time step. Of course, the choice of these parts is not obvious, nevertheless, for the description of the general method, we assume that a refinement criterion is available so that we know the area to refine. We will give, in Section 3.3.3, the precise criterion we used for the resolution of the Cahn-Hilliard model.

To increase the spatial resolution in these selected areas, a solution is to use elements with smaller diameters, i.e. to split some cells into smaller ones. The main difficulty is to preserve, at the same time, the geometric conformity of the mesh and then its "good" geometric quality. The geometric conformity prevents the so-called hanging nodes which for different reasons are undesired in many applications. For instance, hanging nodes do not represent degree of freedom and are somewhat difficult to handle, because the local correlation pattern of the stiffness matrix is disturbed. When they exist, their taking into account may be carried out by many ways, for instance, by direct elimination of these "false" unknowns, or by adding constraint and using penalty methods or Lagrange multiplier methods. In these cases, numerical methods and schemes are modified.

Another method consists in eliminating non-conformity of the mesh by splitting cells until there is no non-conformal edge left. In two dimensions, for triangular meshes, we may for instance cite the red-green refinement method [2]. This technique consists in first using a "regular" splitting (called red refinement) of triangles in four congruent ones by connecting the midpoints of its edges. This refinement preserves geometric properties of triangles but creates non-conformal edges when a refined triangle and an unrefined one are adjacent. Therefore, a second type of refinement (called green refinement) is first used connecting one edge midpoint and the opposite

corner. This yields a simple bisection which is "irregular" but only used for solving conformity issues. Bey [4] and Zhang [24] proposed a generalisation of this method to three dimensions. Other methods based only on bisection have been introduced by Rivara [21, 22] or Mitchell [18] in two dimension and Bänsch [3] or Maubach [17] in three dimensions. All these methods depend on the choice of the element and on the dimension. Moreover, their implementation tends to be quite complex, notably in three dimensions.

A possible alternative considered in this paper is to adopt the point of view of basis functions refinement instead of cells refinement. In this approach [15], a stock of basis functions with increasing spatial resolution is assumed to be given through a nested sequence of approximation spaces $X_0 \subset \cdots \subset X_J$, $J \geqslant 1$. Local refinement is then performed by using multilevel approximation spaces containing few basis functions suitably selected in each approximation spaces X_j, $j \in [\![0, J]\!]$. In this framework, Krysl, Grinspun and Schröder in [15] (see also [10, 13, 16]) proposed procedures called CHARMS (Conforming Hierarchical Adaptive Refinement MethodS) which enable to (un)refine multilevel approximation spaces. We give here a precise construction of the nested sequence $X_0 \subset \cdots \subset X_J$ for suitable Lagrange finite element (e.g. $\mathbb{P}_k, \mathbb{Q}_k$, $k \geqslant 1$) using a unique refinement pattern and we study a slightly modified version of the quasi-hierarchical CHARMS adaptation technique (the "one-level-difference refinement rule" is used as a separate criterion, and is not involved in the proof of the properties of the procedure). Finally, we incorporate multigrid preconditioners [1, 8, 12, 23, 24] in the method. Attractive features follow from this methodology:

- there is no modification of the discrete problem due to the mesh adaptation,
- the cells are divided into cells of the same type, uniformly applying the same subdivision pattern,
- the possible geometric non-conformity of the adapted meshes are implicitly handled,
- there is no specific treatment due to particular Lagrange finite elements, e.g. $\mathbb{P}_k, \mathbb{Q}_k$, $k \geqslant 1$,
- all the procedure is independent of the space dimension,
- transfer operators between the different grids are not needed for time evolution problems provided that a suitable definition of elementary integration domains is used when assembling the system.

The detailed outline of the paper is the following. In Section 1, we introduce the notion of refinement pattern for suitable Lagrange elements (e.g. $\mathbb{P}_k$ and $\mathbb{Q}_k$) and we show how a hierarchy of nested conforming meshes can be built recursively by applying a unique refinement pattern to each cell of a given initial mesh. We investigate in details the parent-child relationship between basis functions of successive refinement levels. In particular, we show that all the coefficients in this linear relationship can be simply computed by considering the refinement pattern on the reference element. Consequently, the meshes hierarchy is never explicitly created since every necessary information is available on the reference element.

In Section 2, we describe the local adaptation procedure in the above framework and we establish its main properties. In particular, we give precise sufficient conditions ensuring that:

- the selected basis functions on the various levels are always linearly independent.
- no information is lost when refining a basis function: the refinement algorithm produces increasing approximation spaces sequences.
- the approximation space obtained by refining (resp. unrefining) a set of basis functions is independent of the order in which successive refinements (resp. unrefinements) are performed.

Some counterexamples illustrate the fact that these properties are not satisfied in the general case.

In the second part of Section 2, we show that the multilevel structure of the approximation spaces built following the above methodology can be exploited to derive efficient multigrid preconditioners. All the results in this section are illustrated by numerical results on a model problem.

Section 3 is devoted to illustrations to more complicated partial differential equations. We concentrate on the simulation of three-phase systems thanks to a diffuse-interface model of Cahn-Hilliard type. The model is a non-linear parabolic problem describing the evolution of thin interfaces. This is the reason why such a local adaptive approach is particularly fruitful. Interestingly enough, we focus on the fact that, even though the approximation spaces may vary from a time step to another, it is not needed to make use of transfer operators between the different grids when assembling the matrix of the system. In the conclusion of the paper, we give

even more complex simulation results we obtained by solving the complete three-phase flows model constituted by a coupling between the Cahn-Hilliard system and the Navier-Stokes equations, which is the main motivation of this study.

1. Multilevel Finite Element Spaces

1.1. Preliminary notation and definitions

This subsection introduces some classical notation, definitions and properties, following [9] and [20], about meshes and Lagrange finite elements. These would be useful in Sections 1.2 and 1.3.

Definition 1.1 (Lagrange Finite Element [20, §4.1]). A Lagrange finite element is a triple (K, Σ, P) where
- K is a compact, connected, Lipschitz subset of $\mathbb{R}^d$ ($d = 1$, 2 or 3),
- $\Sigma = \{\boldsymbol{a}_k \in K; 1 \leqslant k \leqslant N\}$ is a set of N distinct points belonging to K, called the Lagrange nodes,
- P is a vector space of functions $p : K \to \mathbb{R}$ such that Σ is P-unisolvent [20, Def 4.1-1], i.e.

$$\forall(\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N, \ \exists! p \in P, \ \forall k \in [\![1, N]\!], \ p(\boldsymbol{a}_k) = \alpha_k.$$

The element (K, Σ, P) is called a polygonal Lagrange finite element iff K is a polygon.

Definition 1.2 (Mesh [9, Def 1.49]). Let $\omega \subset \mathbb{R}^d$ ($d = 1$, 2 or 3) be a domain [9, Def 1.46]. A mesh $\mathcal{T}$ of ω is a set $\{K_e \subset \omega; 1 \leqslant e \leqslant N_e\}$ of compact, connected, Lipschitz subsets of ω with non-empty interior (called cells), such that

- $\overline{\omega} = \bigcup_{e=1}^{N_e} K_e$ and,
- $\mathring{K}_e \cap \mathring{K}_f = \emptyset$ for any pair of distinct cells (K_e, K_f).

Definition 1.3 (Finite element mesh generated by a reference element [9, §1.3.2]). Let $\mathcal{T} = \{K_e; 1 \leqslant e \leqslant N_e\}$ be a mesh of a domain ω. Let $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ be a polygonal Lagrange finite element, hereafter called reference element. We say that the mesh $\mathcal{T}$ is generated using the reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ iff:

For all $e \in [\![1, N_e]\!]$, there exists a C^1-diffeomorphism T_e from $\widehat{K}$ to K_e.

When the transformations T_e, $1 \leqslant e \leqslant N_e$, are affine, the mesh is said to be affine.

For all $e \in [\![1, N_e]\!]$, we can define a Lagrange finite element (K_e, Σ_e, P_e) by setting:
- $\Sigma_e = T_e(\widehat{\Sigma})$ and
- $P_e = \{p \circ T_e^{-1}; p \in \widehat{P}\}$.

Remark 1.4. We always assume that the reference element is polygonal. However, depending on the mapping T_e, a cell K_e may be non polygonal [9, Figure 1.13].

The following definitions are commonly used in order to build H$^1(\omega)$-conformal finite element approximation spaces X, i.e. such that $X \subset$ H$^1(\omega)$, see Proposition 1.10.

Definition 1.5 (C^0-class element [20, Def 5.1-2]). The polygonal Lagrange finite element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ is a C^0-class element iff
- $\widehat{P} \subset$ C$^0(\widehat{K})$ and
- for any face $\widehat{F}$ of $\widehat{K}$, $\widehat{\Sigma} \cap \widehat{F}$ is $\widehat{P}_{|\widehat{F}}$-unisolvent where $\widehat{P}_{|\widehat{F}} = \{\widehat{\varphi}_{|\widehat{F}}; \widehat{\varphi} \in \widehat{P}\}$.

Definition 1.6 (Compatibility requirements [20, Def 5.1-3]). The compatibility requirements for the Lagrange reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ hold iff for all faces $\widehat{F}_1$ and $\widehat{F}_2$ of $\widehat{K}$, for all affine invertible function $\widehat{A}$ such that $\widehat{F}_2 = \widehat{A}(\widehat{F}_1)$, we have,

- $\widehat{\Sigma} \cap \widehat{F}_2 = \widehat{A}(\widehat{\Sigma} \cap \widehat{F}_1)$ and
- $\{\widehat{\varphi}_{|\widehat{F}_1}; \widehat{\varphi} \in \widehat{P}_1\} = \{\widehat{\varphi} \circ \widehat{A}_{|\widehat{F}_1}; \widehat{\varphi} \in \widehat{P}_2\}$.

Definition 1.7 (Geometrically conformal mesh [9, Def 1.55])**.** A mesh $\mathcal{T} = \{K_e; 1 \leqslant e \leqslant N_e\}$ of a domain ω is said to be geometrically conformal iff for all cells K_e and K_f having a non-empty $(d\text{-}1)$-dimensional intersection, say $F = K_e \cap K_f$, $T_e^{-1}(F)$ and $T_f^{-1}(F)$ are faces of $\widehat{K}$, and there exists a bijective affine transformation $\widehat{A} : T_e^{-1}(F) \to T_f^{-1}(F)$ such that $\widehat{A} \circ T_e^{-1} = T_f^{-1}$ on F.

Remark 1.8. Definition 1.7 implies, in particular, that for any pair of distinct cells (K_e, K_f), the intersection $K_e \cap K_f$ is:

- either empty or a common vertex in dimension 1,
- either empty, or a common vertex, or a common face in dimension 2,
- either empty, or a common vertex, or a common edge, or a common face in dimension 3.

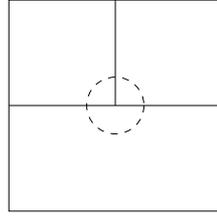An example of a geometrically non-conformal mesh is shown in Figure 1.

FIGURE 1. Example of a geometrically non-conformal mesh.

Remark 1.9. In case that the mesh is geometrically conformal, Definition 1.6 implies, in particular, that the Lagrange nodes on a common face belong to each element sharing the face. An example of incompatible node positions is given in Figure 2.
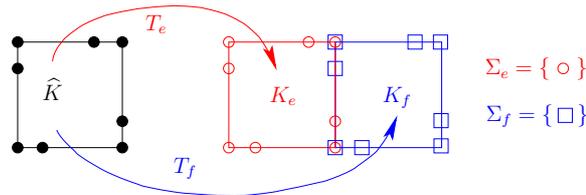

FIGURE 2. Example of incompatible node positions.

Let $\mathcal{T} = \{K_e; 1 \leqslant e \leqslant N_e\}$ be a mesh generated using a Lagrange reference finite element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$. Owing to Definition 1.3, we can associate to this mesh:

- a set of geometric mappings $\{T_e : \widehat{K} \to K_e; 1 \leqslant e \leqslant N_e\}$.
- a set of Lagrange finite elements $\{(K_e, \Sigma_e, P_e); 1 \leqslant e \leqslant N_e\}$.
- a set of Lagrange nodes $\Sigma = \bigcup_{e=1}^{N_e} \Sigma_e$. Let us denote by $N_{\text{dof}} = \#\Sigma$ the number of Lagrange nodes, so that we can write $\Sigma = \{\boldsymbol{a}_i; 1 \leqslant i \leqslant N_{\text{dof}}\}$.

- a set of basis functions $\{\varphi_i; 1 \leqslant i \leqslant N_{\text{dof}}\}$ defined as follows. First, for $e \in [\![1, N_e]\!]$ and $k \in [\![1, \widehat{N}]\!]$ where $\widehat{N} = \#\widehat{\Sigma}$, we denote by $I(e, k) \in \{1, \ldots, N_{\text{dof}}\}$ the corresponding index, in the global numerotation, of the k-th local Lagrange node in the e-th cell of $\mathcal{T}$. That is

$$\forall e \in [\![1, N_e]\!], \ \forall k \in [\![1, \widehat{N}]\!], \quad \boldsymbol{a}_{I(e,k)} = T_e(\widehat{\boldsymbol{a}}_k).$$

Furthermore, since $\widehat{\Sigma}$ is $\widehat{P}$-unisolvent, we have:

$$\forall k \in [\![1, \widehat{N}]\!], \ \exists \widehat{\varphi}_k \in \widehat{P}, \ \forall \ell \in [\![1, \widehat{N}]\!], \ \widehat{\varphi}_k(\widehat{\boldsymbol{a}}_\ell) = \delta_{k\ell}.$$

We can then associate to each node $\boldsymbol{a}_i$, $1 \leqslant i \leqslant N_{\text{dof}}$, a basis function φ_i defined elementwise by

$$\varphi_{i|K_e} = \begin{cases} \widehat{\varphi}_k \circ T_e^{-1} \text{if there exists } k \in [\![1, \widehat{N}]\!] \text{ such that } I(e, k) = i \\ 0 \text{ otherwise.} \end{cases}$$

Owing to this definition, we denote by $\text{supp}[\varphi_i]$ the following subset of ω:

$$\text{supp}[\varphi_i] = \bigcup_{e \in \mathcal{E}} K_e \text{ where } \mathcal{E} = \left\{e \in [\![1, N_e]\!]; \exists k \in [\![1, \widehat{N}]\!], i = I(e, k)\right\}.$$

- a $H^1(\omega)$-conformal approximation space [9, Prop 1.74]

$$X = \left\{v \in C^0(\omega); \forall e \in [\![1, N_e]\!], \ v_{|K_e} \in P_e\right\}.$$

The advantage of this construction is not only to produce a $H^1(\omega)$-conformal approximation space but most important to provide an explicit basis of this space. Indeed, we have the following result:

Proposition 1.10. *Let $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ be a C^0-class polygonal Lagrange reference element satisfying the compatibility requirements 1.6. Let $\mathcal{T} = \{K_e; 1 \leqslant e \leqslant N_e\}$ be a geometrically conformal mesh of ω generated using the reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$. Then, we have $\varphi_k(\boldsymbol{a}_\ell) = \delta_{k\ell}$, and*

$$\left\{\varphi_1, \ldots, \varphi_{N_{dof}}\right\} \ \text{is a basis of } X.$$

Proof. See [9, Prop 1.78] $\square$

1.2. Refinement pattern

Let us now define the notion of refinement pattern and the associated compatibility requirements that will be useful to generate geometric conformal uniformly refined meshes in Section 1.3.1.

Definition 1.11 (Refinement pattern)**.** Let

- $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ be a C^0-class polygonal Lagrange reference finite element satisfying the compatibility requirements 1.6,
- $\widehat{\mathcal{T}} = \{\widehat{K}_e^{[1]}; 1 \leqslant e \leqslant \widehat{N}_e^{[1]}\}$ be a geometrically conformal affine mesh of the interior of $\widehat{K}$ generated using the reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$ itself.

Then, we say that $(\widehat{K}, \widehat{\Sigma}, \widehat{P}, \widehat{\mathcal{T}})$ is a refinement pattern.

Owing to the previous subsection let us denote by

- $\{\widehat{T}_e : \widehat{K} \to \widehat{K}_e^{[1]}; 1 \leqslant e \leqslant \widehat{N}_e^{[1]}\}$ the set of geometric mappings,
- $\{(\widehat{K}_e^{[1]}, \widehat{\Sigma}_e^{[1]}, \widehat{P}_e^{[1]}); 1 \leqslant e \leqslant \widehat{N}_e^{[1]}\}$ the set of Lagrange finite elements,
- $\widehat{\Sigma}^{[1]} = \{\widehat{\boldsymbol{a}}_k^{[1]}; 1 \leqslant k \leqslant \widehat{N}^{[1]}\}$ the set of Lagrange nodes,

- $\{\widehat{\varphi}_k^{[1]}; 1 \leqslant k \leqslant \widehat{N}^{[1]}\}$ the set of basis functions,

associated to the mesh $\widehat{\mathcal{T}}$.

Definition 1.12 (Compatibility requirements). We say that a refinement pattern $(\widehat{K}, \widehat{\Sigma}, \widehat{P}, \widehat{\mathcal{T}})$ satisfies the compatibility requirements iff

- for all $e \in [\![1, \widehat{N}_{\mathrm{e}}^{[1]}]\!]$, $\{\varphi_{|\widehat{K}_e^{[1]}}; \varphi \in \widehat{P}\} \subset \widehat{P}_e^{[1]}$,
- for all faces $\widehat{F}_1$, $\widehat{F}_2$ of $\widehat{K}$, for all affine invertible function $\widehat{A}$ such that $\widehat{F}_2 = \widehat{A}(\widehat{F}_1)$, for all faces $\widehat{F}_1^{[1]} \subset \widehat{F}_1$ of an element $\widehat{K}_e^{[1]}$, $\widehat{A}(\widehat{F}_1^{[1]}) \subset \widehat{F}_2$ is exactly a face of an other element $\widehat{K}_f^{[1]}$,
- any node of $\widehat{\Sigma}$ is also a node of $\widehat{\Sigma}^{[1]}$, that is $\widehat{\Sigma} \subset \widehat{\Sigma}^{[1]}$.

Remark 1.13. The Definition 1.12 is used to avoid refinement patterns which would lead in the sequel to non-conformal meshes. An example is given with Figure 3.
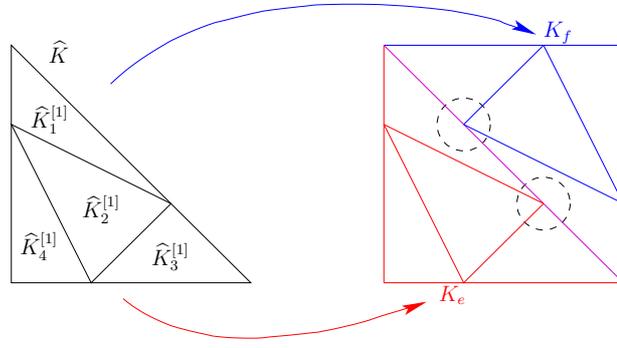

FIGURE 3. Example of an incompatible refinement pattern.

The compatibility requirements of refinement patterns 1.12 are also used to etablish the so-called refinement equation on the reference element which is the cornerstone of CHARMS method [15].

Proposition 1.14 (Refinement equation for the refinement pattern)**.** *Let $(\widehat{K}, \widehat{\Sigma}, \widehat{P}, \widehat{\mathcal{T}})$ be a refinement pattern satisfying the compatibility requirements 1.12. We have the following relationship*
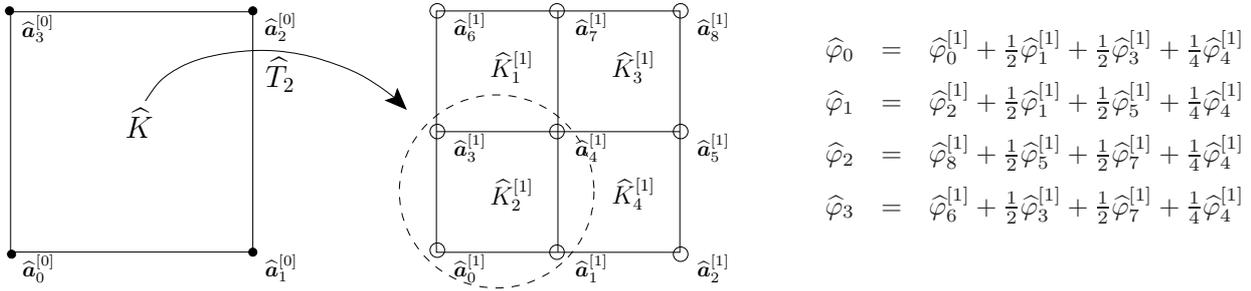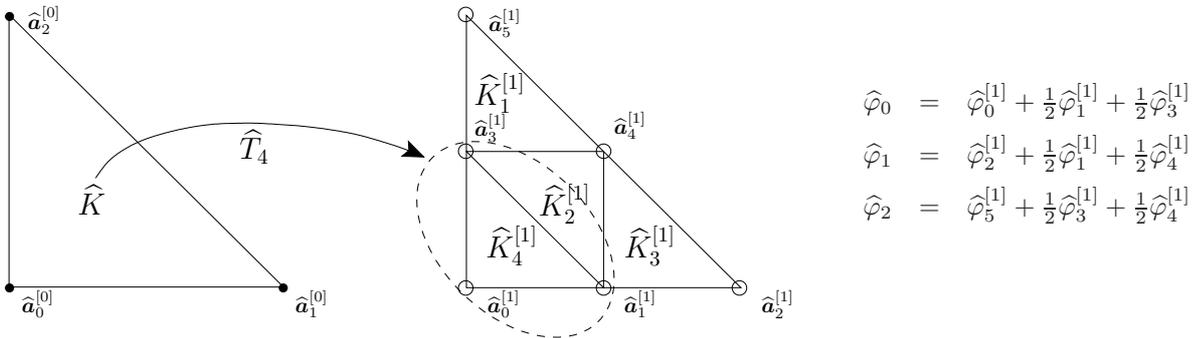
$$\forall k \in [\![1, \widehat{N}]\!], \quad \widehat{\varphi}_k = \sum_{\ell=1}^{\widehat{N}^{[1]}} \widehat{\beta}_{k\ell} \widehat{\varphi}_\ell^{[1]} \quad where \quad \widehat{\beta}_{k\ell} = \widehat{\varphi}_k(\widehat{a}_\ell^{[1]}).$$

Proof. Owing to Proposition 1.10, $\{\widehat{\varphi}_1^{[1]}, \ldots, \widehat{\varphi}_{\widehat{N}^{[1]}}^{[1]}\}$ is a basis of $\{v \in \mathrm{C}^0(\widehat{K}); \ \forall e \in [\![1, \widehat{N}_{\mathrm{e}}^{[1]}]\!], \ v_{|\widehat{K}_e^{[1]}} \in \widehat{P}_e^{[1]}\}$. However, for all $k \in \{1, \ldots, \widehat{N}\}$, the basis function $\widehat{\varphi}_k$ is in $\mathrm{C}^0(\widehat{K})$ and using compatibility requirements 1.12, we have $\forall e \in [\![1, \widehat{N}_{\mathrm{e}}^{[1]}]\!]$, $\widehat{\varphi}_{k|\widehat{K}_e^{[1]}} \in \widehat{P}_e^{[1]}$. Hence, there exist coefficients $\widehat{\beta}_{k\ell}$ such that

$$\widehat{\varphi}_k = \sum_{\ell=1}^{\widehat{N}^{[1]}} \widehat{\beta}_{k\ell} \widehat{\varphi}_\ell^{[1]}.$$

Finally, the coefficients $\widehat{\beta}_{kt}$ can be obtained thanks to the relationship $\widehat{\varphi}_\ell^{[1]}(\widehat{a}_t^{[1]}) = \delta_{\ell t}$. □

Notations of this section are illustrated by Figures 4 and 5 which show the complete refinement patterns and all the refinement equations of .$\mathbb{Q}_1$-square and $\mathbb{P}_1$-triangle elements.

$$\widehat{\varphi}_0 = \widehat{\varphi}_0^{[1]} + \tfrac{1}{2}\widehat{\varphi}_1^{[1]} + \tfrac{1}{2}\widehat{\varphi}_3^{[1]} + \tfrac{1}{4}\widehat{\varphi}_4^{[1]}$$

$$\widehat{\varphi}_1 = \widehat{\varphi}_2^{[1]} + \tfrac{1}{2}\widehat{\varphi}_1^{[1]} + \tfrac{1}{2}\widehat{\varphi}_5^{[1]} + \tfrac{1}{4}\widehat{\varphi}_4^{[1]}$$

$$\widehat{\varphi}_2 = \widehat{\varphi}_8^{[1]} + \tfrac{1}{2}\widehat{\varphi}_5^{[1]} + \tfrac{1}{2}\widehat{\varphi}_7^{[1]} + \tfrac{1}{4}\widehat{\varphi}_4^{[1]}$$

$$\widehat{\varphi}_3 = \widehat{\varphi}_6^{[1]} + \tfrac{1}{2}\widehat{\varphi}_3^{[1]} + \tfrac{1}{2}\widehat{\varphi}_7^{[1]} + \tfrac{1}{4}\widehat{\varphi}_4^{[1]}$$

FIGURE 4. $\mathbb{Q}_1$-square refinement pattern and refinement equations.

$$\widehat{\varphi}_0 = \widehat{\varphi}_0^{[1]} + \tfrac{1}{2}\widehat{\varphi}_1^{[1]} + \tfrac{1}{2}\widehat{\varphi}_3^{[1]}$$

$$\widehat{\varphi}_1 = \widehat{\varphi}_2^{[1]} + \tfrac{1}{2}\widehat{\varphi}_1^{[1]} + \tfrac{1}{2}\widehat{\varphi}_4^{[1]}$$

$$\widehat{\varphi}_2 = \widehat{\varphi}_5^{[1]} + \tfrac{1}{2}\widehat{\varphi}_3^{[1]} + \tfrac{1}{2}\widehat{\varphi}_4^{[1]}$$

FIGURE 5. $\mathbb{P}_1$-triangle refinement pattern and refinement equations.

Since the geometrical configuration is more complicated for the $\mathbb{Q}_2$-square element, we only give in Figure 6 the non-zero coefficients in the refinement equation associated to three coarse basis functions. More precisely, on each picture, the coarse basis function represented by a black bullet is the linear combination of the finer basis functions represented by circles with the coefficients mentionned nearby. The other refinement equations can be readily obtained by symmetry.
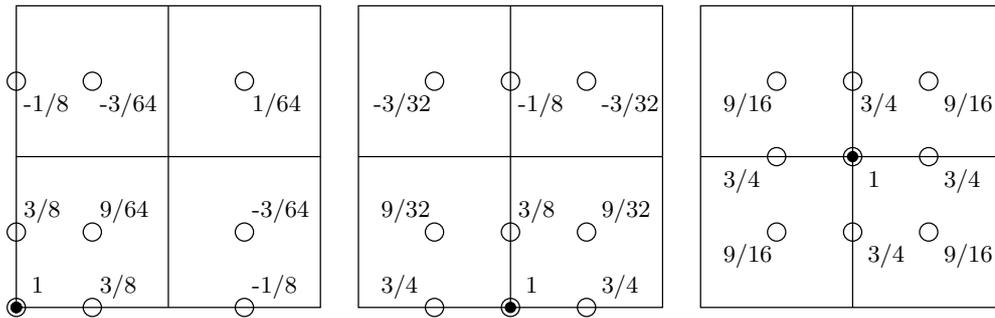
FIGURE 6. $\mathbb{Q}_2$-square refinement coefficients.

1.3. Multilevel finite element approximation spaces

Let Ω be a bounded domain of $\mathbb{R}^d$, $d = 1, 2$ or 3. The purpose of this section is to give an automatic way to construct $\mathrm{H}^1(\Omega)$-conformal multilevel finite element approximation spaces from an initial geometrically conformal mesh $\mathcal{T}_0$ and a given refinement pattern. Let $J \in \mathbb{N}^*$. First, we construct a hierarchy of nested

meshes $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_J$ from $\mathcal{T}_0$ by applying uniformly and recursively the refinement pattern. Then, a multilevel approximation space is obtained by selecting some basis functions associated to each mesh $\mathcal{T}_j$, $0 \leqslant j \leqslant J$ in a way that guarantees linear independence of the selected basis functions.

1.3.1. *Hierarchy of nested* $\mathrm{H}^1(\Omega)$-*conformal approximation spaces. Parent-child relationship*

Let $(\widehat{K}, \widehat{\Sigma}, \widehat{P}, \widehat{\mathcal{T}})$ be a refinement pattern and $j \in [\![0, J-1]\!]$. In this subsection, we assume that a geometrically conformal mesh $\mathcal{T}_j = \{K_e^{[j]}; 1 \leqslant e \leqslant N_\mathrm{e}^{[j]}\}$ of Ω is given, that this mesh $\mathcal{T}_j$ is generated using the reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$, and then we explain how we can build the mesh $\mathcal{T}_{j+1}$.

In the sequel, all mathematical objects associated to the mesh $\mathcal{T}_j$ will be marked with the sign j as follows:

- $T_e^{[j]}$ is the geometric mapping used to generate $K_e^{[j]}$, i.e. $K_e^{[j]} = T_e^{[j]}(\widehat{K})$,
- $\{\boldsymbol{a}_1^{[j]}, \ldots, \boldsymbol{a}_{N_\mathrm{dof}^{[j]}}^{[j]}\}$ is the set of the Lagrange nodes of the mesh $\mathcal{T}_j$, called level-$[j]$ nodes,
- $\mathrm{B}_j = \{\varphi_1^{[j]}, \ldots, \varphi_{N_\mathrm{dof}^{[j]}}^{[j]}\}$ is the set of basis functions of the mesh $\mathcal{T}_j$, called level-$[j]$ basis functions,
- $X_j = \{v \in \mathrm{C}^0(\Omega); \quad \forall e \in [\![1, N_\mathrm{e}^{[j]}]\!], \ v_{|K_e^{[j]}} \in P_e^{[j]}\}$ is the $\mathrm{H}^1(\Omega)$-conformal approximation space associated to $\mathcal{T}_j$.

Owing to Proposition 1.10, the following result holds:

$$\mathrm{B}_j \text{ is a basis of the } \mathrm{H}^1(\Omega)\text{-conformal approximation space } X_j.$$

Definition 1.15. We define the set $\mathcal{T}_{j+1}$ as follows:

$$\mathcal{T}_{j+1} = \left\{ K_{ef}^{[j+1]} = T_e^{[j]}(\widehat{K}_f^{[1]}); 1 \leqslant e \leqslant N_\mathrm{e}^{[j]}, 1 \leqslant f \leqslant \widehat{N}_\mathrm{e}^{[1]} \right\}.$$

Proposition 1.16. *The set $\mathcal{T}_{j+1}$ is a geometrically conformal mesh of Ω generated by the reference element $\widehat{K}$.*

Sketch of the proof. For the sake of simplicity, we only give here a sketch of the proof. It is straightforward to see that $\mathcal{T}_{j+1}$ is a mesh of Ω. Hence, we only have to prove that it is geometrically conformal. Let $K_{ef}^{[j+1]}$ and $K_{e'f'}^{[j+1]}$ be two cells which have a non empty $(d-1)$-dimensional intersection, say $F = K_{ef}^{[j+1]} \cap K_{e'f'}^{[j+1]}$. The proof is based on different arguments depending on whether $e = e'$ or $e \neq e'$. In the first case, we use the geometric conformity of $\widehat{\mathcal{T}}$ whereas in the second case the result is deduced from the geometrical conformity of $\mathcal{T}_j$ and compatibility requirements 1.12. $\square$

The last point of the compatibility requirements 1.12 obviously ensures that a level-$[j]$ node is also a level-$[j+1]$ node. That is

$$\forall k \in [\![1, N_\mathrm{dof}^{[j]}]\!], \ \exists \ell \in [\![1, N_\mathrm{dof}^{[j+1]}]\!], \ \boldsymbol{a}_k^{[j]} = \boldsymbol{a}_\ell^{[j+1]}. \tag{1}$$

We can now prove that our construction leads to embedded approximation spaces:

Proposition 1.17. *It holds that $X_j \subset X_{j+1}$.*

Proof. Let $v \in X_j$, $e \in \{1, \ldots, N_\mathrm{dof}^{[j]}\}$ and $f \in \{1, \ldots, \widehat{N}_\mathrm{e}^{[1]}\}$. By definition, $v_{|K_e^{[j]}} \in P_e^{[j]}$. This is equivalent to $v \circ T_e^{[j]} \in \widehat{P}$. Using the compatibility requirements 1.12, we have $v \circ T_e^{[j]}{}_{|\widehat{K}_f^{[1]}} \in \widehat{P}_f^{[1]}$. Hence, we get $v \circ T_e^{[j]} \circ \widehat{T}_f \in \widehat{P}$, which exactly means that $v_{|K_{ef}^{[j+1]}} \in P_{ef}^{[j+1]}$, and the claim is proved. $\square$

For the following result, we have to introduce relevant indexation of level-$[j]$ and level-$[j+1]$ nodes. Note that level-$[j]$ and level-$[j+1]$ nodes belonging to $K_e^{[j]}$ are, by definition, image of nodes of $\widehat{\Sigma}$ and $\widehat{\Sigma}^{[1]}$ respectively under the mapping $T_e^{[j]}$. Hence, we denote by:

- $I^{[j]}(e,k)$ the index of the level-$[j]$ node which is the image of $\widehat{a}_k$ under the mapping $T_e^{[j]}$. That is

$$\forall (e,k) \in [\![1, N_{\mathrm{e}}^{[j]}]\!] \times [\![1, \widehat{N}]\!], \quad a_{I^{[j]}(e,k)}^{[j]} = T_e^{[j]}(\widehat{a}_k).$$

- $I^{[j,1]}(e,\ell)$ the index of the level-$[j+1]$ node which is the image of $\widehat{a}_\ell^{[1]}$ under the mapping $T_e^{[j]}$. That is

$$\forall (e,\ell) \in [\![1, N_{\mathrm{e}}^{[j]}]\!] \times [\![1, \widehat{N}^{[1]}]\!], \quad a_{I^{[j,1]}(e,\ell)}^{[j+1]} = T_e^{[j]}(\widehat{a}_\ell^{[1]}).$$

Proposition 1.18 (Refinement equation). *The following relationship holds:*

$$\forall i \in [\![1, N_{dof}^{[j]}]\!], \quad \varphi_i^{[j]} = \sum_{t=1}^{N_{dof}^{[j+1]}} \beta_{it}^{[j]} \varphi_t^{[j+1]} \tag{RE}$$

where the coefficients $\beta_{it}^{[j]}$ are given by: $\forall (i,t) \in [\![1, N_{dof}^{[j]}]\!] \times [\![1, N_{dof}^{[j+1]}]\!]$,

$$\beta_{it}^{[j]} = \begin{cases} \widehat{\beta}_{k\ell} & \textit{if } \exists (e,k,\ell) \in [\![1, N_{\mathrm{e}}^{[j]}]\!] \times [\![1, \widehat{N}]\!] \times [\![1, \widehat{N}^{[1]}]\!] \textit{ s.t. } i = I^{[j]}(e,k) \textit{ and } t = I^{[j,1]}(e,\ell), \\ 0 & \textit{otherwise.} \end{cases}$$

Remark 1.19. Note that the coefficients $\widehat{\beta}_{k\ell}$ only depend on the refinement pattern. Hence, these coefficients can be computed beforehand. They are in small number and therefore, requiring a low memory cost, they can be stored. Thus, the above refinement equations can be deduced without computation of any coefficient. In practice, coefficients $\beta_{it}^{[j]}$ are obtained thanks to a loop on level-$[j]$ cells included in $\mathrm{supp}[\varphi_i^{[j]}]$ by setting: for all $e \in [\![1, N_{\mathrm{e}}^{[j]}]\!]$ such that $K_e^{[j]} \subset \mathrm{supp}[\varphi_i^{[j]}]$, for all $(k,\ell) \in [\![1, \widehat{N}]\!] \times [\![1, \widehat{N}^{[1]}]\!]$,

$$\beta_{I^{[j]}(e,k) I^{[j,1]}(e,\ell)}^{[j]} = \widehat{\beta}_{k\ell},$$

and other coefficients are zero. Remark that, such a loop may lead to consider several times the same pair of indices $(I^{[j]}(e,k), I^{[j,1]}(e,\ell))$ for distinct e,k,ℓ. Proposition 1.18 ensures that corresponding coefficients $\widehat{\beta}_{k\ell}$ are the same.

Proof of Proposition 1.18. Let $i \in [\![1, N_{\mathrm{dof}}^{[j]}]\!]$. The basis function $\varphi_i^{[j]}$ belongs to X_j. Since $X_j \subset X_{j+1}$ and B_{j+1} is a basis of X_{j+1}, the existence of the coefficients $\beta_{it}^{[j]}$ is straightforward.

Let $(i,t) \in [\![1, N_{\mathrm{dof}}^{[j]}]\!] \times [\![1, N_{\mathrm{dof}}^{[j+1]}]\!]$. We have:

$$\varphi_i^{[j]} = \sum_{t=1}^{N_{\mathrm{dof}}^{[j+1]}} \beta_{it}^{[j]} \varphi_t^{[j+1]}. \tag{2}$$

- Case 1 : there exists $(e,k,\ell) \in [\![1, N_{\mathrm{e}}^{[j]}]\!] \times [\![1, \widehat{N}]\!] \times [\![1, \widehat{N}^{[1]}]\!]$ such that $i = I^{[j]}(e,k)$ and $t = I^{[j,1]}(e,\ell)$.
  The restriction of (2) to $K_e^{[j]}$ yields:

$$\widehat{\varphi}_k \circ \left(T_e^{[j]}\right)^{-1} = \sum_{l=1}^{\widehat{N}^{[1]}} \beta_{i I^{[j,1]}(e,l)}^{[j]} \varphi_{I^{[j,1]}(e,l)}^{[j+1]}.$$

Owing to Proposition 1.14, we have:

$$\sum_{l=1}^{\widehat{N}^{[1]}} \widehat{\beta}_{kl} \widehat{\varphi}_l^{[1]} \circ \left(T_e^{[j]}\right)^{-1} = \sum_{l=1}^{\widehat{N}^{[1]}} \beta_{i I^{[j,1]}(e,l)}^{[j]} \widehat{\varphi}_l^{[1]} \circ \left(T_e^{[j]}\right)^{-1}.$$

Evaluating this equality at $T_e^{[j]}(\widehat{\boldsymbol{a}}_\ell^{[1]})$, it follows that $\beta_{iI^{[j,1]}(e,\ell)}^{[j]} = \widehat{\beta}_{k\ell}$. This is exactly $\beta_{it}^{[j]} = \widehat{\beta}_{k\ell}$.

- Case 2: $\forall(e,k,\ell) \in [\![1, N_e^{[j]}]\!] \times [\![1, \widehat{N}]\!] \times [\![1, \widehat{N}^{[1]}]\!]$, $i \neq I^{[j]}(e,k)$ or $t \neq I^{[j,1]}(e,\ell)$.

  Let $e \in [\![1, N_e^{[j]}]\!]$ and $\ell \in [\![1, \widehat{N}^{[1]}]\!]$ such that $t = I^{[j,1]}(e,\ell)$. We necessarily have, by assumption, $\forall k \in [\![1, \widehat{N}]\!]$, $i \neq I^{[j]}(e,k)$. Hence, we get:

$$0 = \sum_{s=1}^{N_{\text{dof}}^{[j+1]}} \beta_{is}^{[j]} \varphi_s^{[j+1]}{}_{|K_e^{[j]}} = \sum_{v=1}^{\widehat{N}^{[1]}} \beta_{iI^{[j,1]}(e,v)}^{[j]} \widehat{\varphi}_v^{[1]} \circ (T_e^{[j]})^{-1}.$$

Evaluating this equality at $T^{[j]}(\widehat{\boldsymbol{a}}_\ell^{[1]})$ yields $\beta_{iI^{[j,1]}(e,\ell)}^{[j]} = 0$. This is exactly, $\beta_{it}^{[j]} = 0$.

$\square$

The refinement equation (RE) introduces a relationship between a level-$[j]$ basis function and some level-$[j+1]$ basis functions which are called its children.

Definition 1.20 (Parent-child relationship for basis functions). In the case where $\beta_{it}^{[j]} \neq 0$, we say that:
- the level-$[j]$ basis function $\varphi_i^{[j]}$ is a parent of the level-$[j+1]$ basis function $\varphi_t^{[j+1]}$,
- the level-$[j+1]$ basis function $\varphi_t^{[j+1]}$ is a child of the level-$[j]$ basis function $\varphi_i^{[j]}$.

For this reason, the refinement equation (RE) is also called the parent-child relationship. Along the same lines, we can define a parent-child relationship for cells.

Definition 1.21 (Parent-child relationship for cells). Let $e \in [\![1, N_e^{[j]}]\!]$.
- For all $f \in \{1, \ldots, \widehat{N}_e^{[1]}\}$, we say that the level-$[j+1]$ cell $K_{ef}^{[j+1]}$ is a child cell of the level-$[j]$ cell $K_e^{[j]}$.
- Conversely, we say that the level-$[j]$ cell $K_e^{[j]}$ is the parent cell of each level-$[j+1]$ cell $K_{ef}^{[j+1]}$, for $f \in \{1, \ldots, \widehat{N}_e^{[1]}\}$.

Remark 1.22. A cell has at most one parent cell whereas a basis function may have several parents. Nevertheless, let us identify some basis functions which have only one parent.

Proposition 1.23 (Private Child). *Let* $(k,\ell) \in [\![1, N_{dof}^{[j]}]\!] \times [\![1, N_{dof}^{[j+1]}]\!]$ *such that* $\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_\ell^{[j+1]}$ *Then,*

$$\varphi_k^{[j]} \text{ is the unique parent of } \varphi_\ell^{[j+1]}.$$

Proof. For $1 \leqslant i \leqslant N_{\text{dof}}^{[j]}$, the parent-child relationship yields:

$$\varphi_i^{[j]}(\boldsymbol{a}_k^{[j]}) = \sum_{t=1}^{N_{\text{dof}}^{[j+1]}} \beta_{it}^{[j]} \varphi_t^{[j+1]}(\boldsymbol{a}_\ell^{[j+1]}).$$

This is exactly:

$$\delta_{ik} = \beta_{i\ell}^{[j]}.$$

Thus, the basis function $\varphi_\ell^{[j+1]}$ has a unique parent which is $\varphi_k^{[j]}$. $\square$

Summary 1.24. Let $(\widehat{K}, \widehat{\Sigma}, \widehat{P}, \widehat{\mathcal{T}})$ be a refinement pattern. Let $\mathcal{T}_0$ be a geometrically conformal mesh of Ω generated using the reference element $(\widehat{K}, \widehat{\Sigma}, \widehat{P})$. By applying uniformly the refinement pattern as described in this section, we are able to construct:
- a hierarchy of nested meshes, $\mathcal{T}_0$, $\mathcal{T}_1$, $\ldots$, $\mathcal{T}_J$ (Figures 7, 8, 9),
- a hierarchy of nested $H^1(\Omega)$-conformal finite element approximation spaces $X_0 \subset X_1 \subset \cdots \subset X_J$,

	Mesh	Basis Function Set	Approximation Space	
Level 0	$\mathcal{T}_0$	$B_0 = \{\varphi_k^{[0]}; k = 1, \ldots, N_{\text{dof}}^{[0]}\}$	$X_0 \quad =$	span B_0
Level 1	$\mathcal{T}_1$	$B_1 = \{\varphi_k^{[1]}; k = 1, \ldots, N_{\text{dof}}^{[1]}\}$	$X_1 \quad =$	span B_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Level J	$\mathcal{T}_J$	$B_J = \{\varphi_k^{[J]}; k = 1, \ldots, N_{\text{dof}}^{[J]}\}$	$X_J \quad =$	span B_J

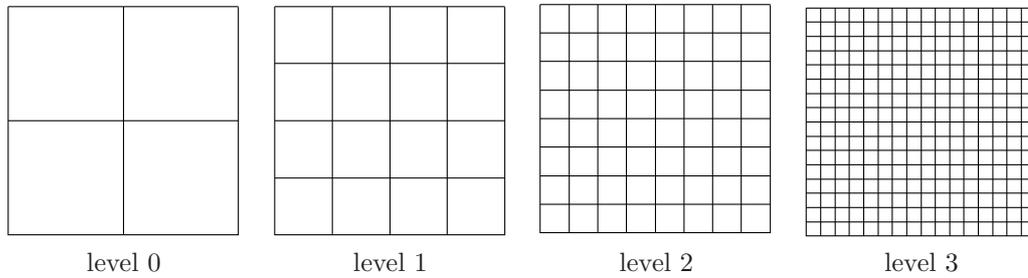TABLE 1. Conceptual hierarchy of nested conformal FE spaces.

level 0   level 1   level 2   level 3

FIGURE 7. Square-$\mathbb{Q}_1$. Nested meshes $\mathcal{T}_j$.

- basis function sets B_0, B_1, ..., B_J, spanning the above approximation spaces such that two consecutive sets are linked by refinement equations,

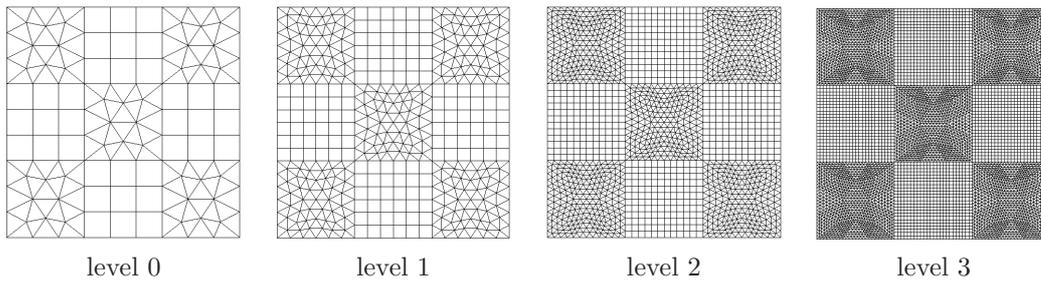Table 1 gives a summary of notations used in the sequel.

level 0   level 1   level 2   level 3

FIGURE 8. Tri/Quadr-angle-$\mathbb{P}_1/\mathbb{Q}_1$. Nested meshes $\mathcal{T}_j$.

Note that the hierarchy of nested meshes is never explicitly created. This conceptual structure is introduced in order to explain the refinement method but in practice, the refinement pattern can be applied only locally where basis functions have to be effectively refined.

level 0   level 1   level 2   level 3
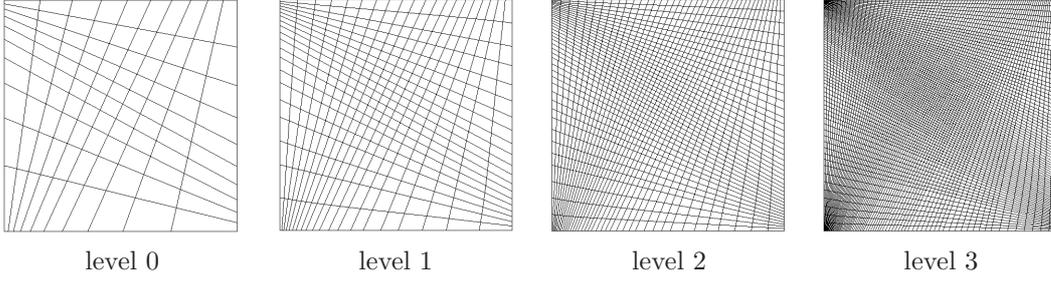
FIGURE 9. Quadrangle-$\mathbb{Q}_1$. Nested meshes $\mathcal{T}_j$.

1.3.2. *Multilevel basis and multilevel approximation spaces*

We assume that a structure as presented in Summary 1.24 is given and we use the same notation (Table 1). The aim of this subsection is to explain how we can select some basis functions in $\bigcup_{j=0}^{J} \mathrm{B}_j$ in order to ensure the linear independence of the selected family.

Proposition 1.25. *Let $\mathcal{B}$ be a subset of $\bigcup_{j=0}^{J} \mathrm{B}_j$. In the case where two nodes associated to distinct basis functions of $\mathcal{B}$ do not have the same location, i.e.*

$$\left(\begin{array}{l} \forall (j,j') \in [\![0,J]\!]^2, \ \forall (k,k') \in [\![1,N_{dof}^{[j]}]\!] \times [\![1,N_{dof}^{[j']}]\!] \ such\ that\ \boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k'}^{[j']}, \\ \qquad\qquad\qquad\qquad (\varphi_k^{[j]} \in \mathcal{B}, \ \varphi_{k'}^{[j']} \in \mathcal{B}) \Longrightarrow (j = j', \ k = k'), \end{array} \right) \qquad (\mathcal{P}_{\mathrm{LI}})$$

then $\mathcal{B}$ is linearly independent.

Remark 1.26. Note that $(\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k'}^{[j']}$ and $j = j') \Longrightarrow k = k'$.

Proof. The property $(\mathcal{P}_{\mathrm{LI}})$ implies that

$$\left(\begin{array}{l} \forall (j,j') \in [\![0,J]\!]^2, \ \forall (k,k') \in [\![1,N_{\mathrm{dof}}^{[j]}]\!] \times [\![1,N_{\mathrm{dof}}^{[j']}]\!] \ such\ that\ j' > j, \\ \qquad\qquad\qquad (\varphi_k^{[j]} \in \mathcal{B}, \ \varphi_{k'}^{[j']} \in \mathcal{B}) \Longrightarrow \varphi_{k'}^{[j']}(\boldsymbol{a}_k^{[j]}) = 0, \end{array} \right) \qquad (3)$$

because, owing to (1), $\boldsymbol{a}_k^{[j]}$ is also a level-$[j']$ node and, by $(\mathcal{P}_{\mathrm{LI}})$, this node is certainly different from $\boldsymbol{a}_{k'}^{[j']}$ (otherwise $j' = j$). Consider a linear combination of basis functions φ belonging to $\mathcal{B}$ such that

$$\sum_{\varphi \in \mathcal{B}} \lambda_\varphi \varphi = 0, \qquad (4)$$

and assume that $\mathcal{E} = \{\varphi \in \mathcal{B}; \lambda_\varphi \neq 0\}$ is not empty. We can then define

$$j_m = \min\{j \in [\![0,J]\!]; \exists k \in [\![1,N_{\mathrm{dof}}^{[j]}]\!] \ such\ that\ \varphi_k^{[j]} \in \mathcal{E}\},$$

and select a $k_m \in [\![1,N_{\mathrm{dof}}^{[j_m]}]\!]$ such that $\varphi_{k_m}^{[j_m]} \in \mathcal{E}$. Let $j \in [\![0,J]\!]$ and $k \in [\![1,N_{\mathrm{dof}}^{[j]}]\!]$ such that $\varphi_k^{[j]} \in \mathcal{E}$ and $(j,k) \neq (j_m,k_m)$. Owing to (3), we have $\varphi_k^{[j]}(\boldsymbol{a}_{k_m}^{[j_m]}) = 0$. Hence, evaluating the linear combination (4) at $\boldsymbol{a}_{k_m}^{[j_m]}$ yields $\lambda_{\varphi_{k_m}^{[j_m]}} = 0$. This is a contradiction and the claim is proved. $\qquad\square$

Owing to Proposition 1.25, we give the following definition of a multilevel basis.

Definition 1.27 (Multilevel basis and multilevel approximation space)**.** We call a multilevel level basis a subset $\mathcal{B}$ of $\bigcup_{j=0}^{J} B_j$ satisfying the property $(\mathcal{P}_{\mathrm{LI}})$. By Proposition 1.25, this set is actually independent. A space spanned by a multilevel basis is called multilevel approximation space.

Remark 1.28. Let $\mathcal{V} = \mathrm{span}\,\mathcal{B}$ be a multilevel approximation space and $u \in \mathcal{V}$. The coordinates of the expansion of u in the multilevel basis $\mathcal{B}$ are not necessarily the values of u at the nodes associated to the corresponding basis function, since two basis functions of different levels may have overlapping supports.

2. Adaptation procedure and multigrid preconditioners

The adaptation consists in adding or removing some basis functions of a given multilevel basis $\mathcal{B}^{\star}$ in order to produce a new multilevel basis $\mathcal{B}$ whose spatial resolution will be better suited to the problem. The main points are to ensure that the Un/Refinement algorithm will actually produce a linearly independent family of basis functions, and that no information is loss during the refinement process. Section 2.1 is devoted to the proofs of such properties. Then, in Section 2.2, we show how to also use the multilevel structure obtained by the adaptation algorithm in order to build a multigrid preconditioning algorithm which let us solve a multilevel linear system with a moderate computational cost.

2.1. **Adaptation**

2.1.1. *Refinement/Unrefinement procedures*

Given a multilevel basis $\mathcal{B}$, let us first introduce the notion of $\mathcal{B}$-refined basis functions.

Definition 2.1 ($\mathcal{B}$-refined basis functions)**.** Let $\mathcal{B}$ a multilevel basis. Let $j \in [\![0, J]\!]$ and $k \in [\![1, N_{\mathrm{dof}}^{[j]}]\!]$. The basis function $\varphi_k^{[j]}$ is said to be $\mathcal{B}$-refined iff:

$$\exists j' \in [\![j+1, J]\!], \ \exists k' \in [\![1, N_{\mathrm{dof}}^{[j']}]\!], \ \text{such that } \varphi_{k'}^{[j']} \in \mathcal{B} \text{ and } \boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k'}^{[j']}.$$

Moreover, if the above condition holds for $j' = j+1$ then we say that the basis function $\varphi_k^{[j]}$ is $\mathcal{B}$-refined only once.

Remark 2.2. Owing to Property $(\mathcal{P}_{\mathrm{LI}})$, notice that:
- the indices j' and k' are necessarily unique.
- a $\mathcal{B}$-refined basis function does not belong to $\mathcal{B}$.

Remark also that, by Proposition 1.23, if $\varphi_k^{[j]}$ is only once $\mathcal{B}$-refined then $\varphi_k^{[j]}$ is the unique parent of $\varphi_{k'}^{[j']}$.

Let us give the following lemma which will be useful in the following proofs.

Lemma 2.3. *Let* $j \in [\![0, J]\!]$ *and* $j' \in [\![0, J-1]\!]$ *such that* $j' \leqslant j$. *Let* $(k, \ell, \ell') \in [\![1, N_{dof}^{[j]}]\!] \times [\![1, N_{dof}^{[j+1]}]\!] \times [\![1, N_{dof}^{[j']}]\!]$. *If* $\boldsymbol{a}_\ell^{[j+1]} = \boldsymbol{a}_{\ell'}^{[j']}$ *and* $\varphi_k^{[j]}$ *is a parent of* $\varphi_\ell^{[j+1]}$ *then the node* $\boldsymbol{a}_k^{[j]}$ *is necessarily at the same position that the nodes* $\boldsymbol{a}_{\ell'}^{[j']}$ *and* $\boldsymbol{a}_\ell^{[j+1]}$.

Proof. Since $j' \leqslant j$, owing to (1) and a straightforward recurrence, there exists $t \in [\![1, N_{\mathrm{dof}}^{[j]}]\!]$ such that $\boldsymbol{a}_t^{[j]} = \boldsymbol{a}_\ell^{[j+1]}$. We can apply Proposition 1.23 which proves that $\varphi_t^{[j]}$ is the unique parent of $\varphi_\ell^{[j+1]}$. Hence, $\varphi_k^{[j]} = \varphi_t^{[j]}$ and then, $k = t$. $\qquad\square$

We can now describe the refinement and unrefinement procedure.

Algorithm 2.4 (Quasi-Hierarchical Un/Refinement)**.** *Let* $\mathcal{B}^{\star}$ *be a multilevel basis.*

- *Refinement: Let $\varphi_k^{[j]}$ be a basis function belonging to $\mathcal{B}^\star$.*

  Refining the given basis function $\varphi_k^{[j]} \in \mathcal{B}^\star$ consists in producing a new multilevel basis $\mathcal{B}$ by
  - *removing this basis function $\varphi_k^{[j]}$, and*
  - *adding all its children $\varphi_\ell^{[j+1]}$ which are not $\mathcal{B}^\star$-refined.*

  This can be written in a compact way as follows:

$$\mathcal{B} = \mathcal{B}^\star \backslash \{\varphi_k^{[j]}\} \cup \{\text{children of } \varphi_k^{[j]} \text{ not } \mathcal{B}^\star\text{-refined}\}.$$

- *Unrefinement: Let $\varphi_k^{[j]}$ be an only once $\mathcal{B}^\star$-refined basis function without $\mathcal{B}^\star$-refined children.*

  Unrefining the given basis function $\varphi_k^{[j]} \notin \mathcal{B}^\star$ consists in producing a new multilevel basis $\mathcal{B}$ by
  - *adding this basis function $\varphi_k^{[j]}$, and*
  - *removing those children of $\varphi_k^{[j]}$ which have no other $\mathcal{B}^\star$-refined parent .*

  This can be written in a compact way as follows:

$$\mathcal{B} = \mathcal{B}^\star \backslash \{\text{children of } \varphi_k^{[j]} \text{ without other } \mathcal{B}^\star\text{-refined parent}\} \cup \{\varphi_k^{[j]}\}.$$

Proof. As claimed in the above algorithm, we have to prove that refinement and unrefinement procedures produce actually a multilevel basis. That is to say that $\mathcal{B}$ satisfies the property $(\mathcal{P}_{\mathrm{LI}})$. Indeed, let $\varphi_k^{[j]}$, $\varphi_{k'}^{[j']}$ two basis functions belonging to $\mathcal{B}$ such that $\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k'}^{[j']}$. We have to show that $j = j'$.

- Refinement: Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of a basis function belonging to $\mathcal{B}^\star$, say $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$, so that we have

$$\mathcal{B} = \mathcal{B}^\star \backslash \{\varphi_{k_0}^{[j_0]}\} \cup \{\text{children of } \varphi_{k_0}^{[j_0]} \text{ not } \mathcal{B}^\star\text{-refined}\}.$$

  - If $\varphi_k^{[j]}$ and $\varphi_{k'}^{[j']}$ belong both to $\mathcal{B}^\star$ then, since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LI}})$, we readily find $j = j'$.
  - Otherwise, assume for instance that $\varphi_k^{[j]}$ does not belong to $\mathcal{B}^\star$. By definition of $\mathcal{B}$, $\varphi_k^{[j]}$ is then a child of $\varphi_{k_0}^{[j_0]}$ which is not $\mathcal{B}^\star$-refined, say $\varphi_{\ell_0}^{[j_0+1]}$, i.e. $k = \ell_0$, $j = j_0 + 1$.
    * If $\varphi_{k'}^{[j']} \in \mathcal{B}^\star$ then, since $\varphi_{\ell_0}^{[j_0+1]}$ is not $\mathcal{B}^\star$-refined, we have $j_0 + 1 \geqslant j'$. Assume that $j_0 + 1 > j'$. Lemma 2.3 yields $\boldsymbol{a}_{k_0}^{[j_0]} = \boldsymbol{a}_{k'}^{[j']}$. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LI}})$, we obtain $j_0 = j'$ and $k_0 = k'$, but $\varphi_{k_0}^{[j_0]} \notin \mathcal{B}$ and $\varphi_{k'}^{[j']} \in \mathcal{B}$. This is a contradiction and we get $j = j_0 + 1 = j'$.
    * Otherwise, $\varphi_{k'}^{[j']}$ is a child of $\varphi_{k_0}^{[j_0]}$, say $\varphi_{\ell_0'}^{[j_0+1]}$, i.e. $k' = \ell_0'$ and $j = j_0 + 1$. Hence, we have $j = j'$.
- Unrefinement: Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of a only once $\mathcal{B}^\star$-refined basis function, say $\varphi_{k_0}^{[j_0]}$, so that we have

$$\mathcal{B} = \mathcal{B}^\star \backslash \{\text{children of } \varphi_{k_0}^{[j_0]} \text{ without other } \mathcal{B}^\star\text{-refined parent}\} \cup \{\varphi_{k_0}^{[j_0]}\}.$$

  - If $\varphi_k^{[j]}$ and $\varphi_{k'}^{[j']}$ belong to $\mathcal{B}^\star$ then, since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LI}})$, we get $j = j'$.
  - Otherwise, assume for instance that $\varphi_k^{[j]}$ does not belong to $\mathcal{B}^\star$. By definition of $\mathcal{B}$, we have $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$, i.e. $k = k_0$, $j = j_0$. Arguing by contradiction, we assume that $j' \neq j$. We have $j' \neq j_0$ and so, $\varphi_{k'}^{[j']} \in \mathcal{B}^\star$. However, $\varphi_{k_0}^{[j_0]}$ is an only once $\mathcal{B}^\star$-refined basis function, so there exists $\ell_0 \in [\![1, N_{\mathrm{dof}}^{[j_0]}]\!]$ such that $\varphi_{\ell_0}^{[j_0+1]} \in \mathcal{B}^\star$ and $\boldsymbol{a}_{\ell_0}^{[j_0+1]} = \boldsymbol{a}_{k_0}^{[j_0]}$. Owing to Proposition 1.23, $\boldsymbol{a}_{\ell_0}^{[j_0+1]} = \boldsymbol{a}_{k_0}^{[j_0]}$ implies that $\varphi_{k_0}^{[j_0]}$ is the unique parent of $\varphi_{\ell_0}^{[j_0+1]}$. Therefore, since $\varphi_{\ell_0}^{[j_0+1]}$ is a child of $\varphi_{k_0}^{[j_0]}$ without other $\mathcal{B}^\star$-refined parent, we get $\varphi_{\ell_0}^{[j_0+1]} \notin \mathcal{B}$. However, since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LI}})$ and since

$\boldsymbol{a}_{\ell_0}^{[j_0+1]} = \boldsymbol{a}_{k'}^{[j']}$, we have $j' = j_0 + 1$ and $k' = \ell_0$. Finally, we get $\varphi_{k'}^{[j']} \notin \mathcal{B}$. This is a contradiction and we obtain $j = j'$.

$\square$

Remark 2.5. This algorithm is consistent with Definition 2.1. Indeed, owing to Proposition 1.23,

- if $\mathcal{B}$ is obtained from $\mathcal{B}^{\star}$ by the refinement of the basis function $\varphi_k^{[j]}$, then $\varphi_k^{[j]}$ is a $\mathcal{B}$-refined basis function in sense of Definition 2.1 ;
- if $\mathcal{B}$ is obtained from $\mathcal{B}^{\star}$ by the unrefinement of the only once $\mathcal{B}^{\star}$-refined basis function $\varphi_k^{[j]}$, then $\varphi_k^{[j]}$ is no longer a $\mathcal{B}$-refined basis function in the sense of Definition 2.1.
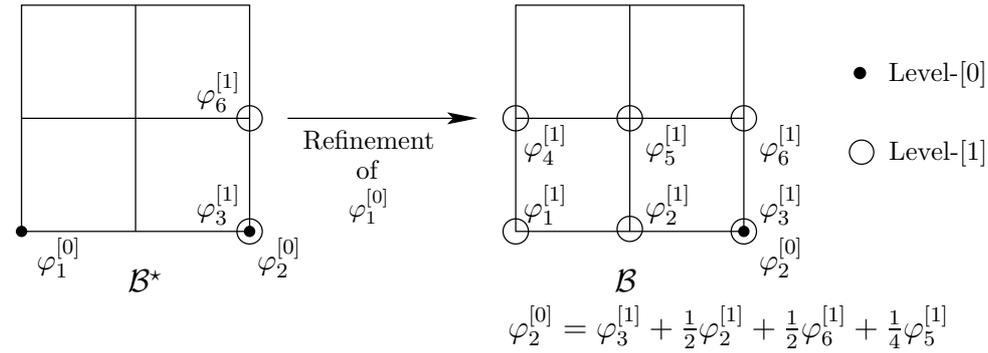

$$\varphi_2^{[0]} = \varphi_3^{[1]} + \tfrac{1}{2}\varphi_2^{[1]} + \tfrac{1}{2}\varphi_6^{[1]} + \tfrac{1}{4}\varphi_5^{[1]}$$

FIGURE 10. Refinement do not preserve the linear independence of multilevel set not satifying ($\mathcal{P}_{\text{LI}}$).

Remark 2.6. Refinement and unrefinement procedure described in Algorithm 2.4 do not preserve in general the linear independence of multilevel basis function set $\bigcup_{j=0}^{J} \widetilde{\mathcal{B}}_j$ (with $\widetilde{\mathcal{B}}_j \subset \mathrm{B}_j$) which do not satisfy the property ($\mathcal{P}_{\text{LI}}$). An example is given in Figure 10. The family $\mathcal{B}^{\star}$ represented on the left hand-side is linearly independent (but do no satisfy ($\mathcal{P}_{\text{LI}}$)) whereas the family $\mathcal{B}$, on the right hand-side, obtained from $\mathcal{B}^{\star}$ by refinement of $\varphi_1^{[0]}$, is not a linear independent family.

2.1.2. *Conservation of information*

A desirable property of a refinement procedure is that it does not involve a loss of information. It means that, if $\mathcal{B}$ is obtained from $\mathcal{B}^{\star}$ by refinement of a basis function, then span $\mathcal{B}^{\star} \subset$ span $\mathcal{B}$, i.e. the refined basis $\mathcal{B}$ allows for any function in the original basis $\mathcal{B}^{\star}$ to be reproduced exactly. However, the refinement procedure described in Algorithm 2.4 is not lossless. An example is given in Figure 11. Nevertheless, we can prove the following results.

Proposition 2.7. *Let $\mathcal{B}$ be a multilevel basis satisfying the following property:*

> *A child of a $\mathcal{B}$-refined basis function either belongs to $\mathcal{B}$ or is itself $\mathcal{B}$-refined.* ($\mathcal{P}_{\text{LO}}$)

Then, all $\mathcal{B}$-refined basis functions belong to span $\mathcal{B}$.

Proof. By a recurrence on the level j of basis functions, we prove the following statement (H_j) for all $j \in [\![0, J]\!]$:

> All level-$[j]$ $\mathcal{B}$-refined basis functions belong to span $\mathcal{B}$. (H_j)

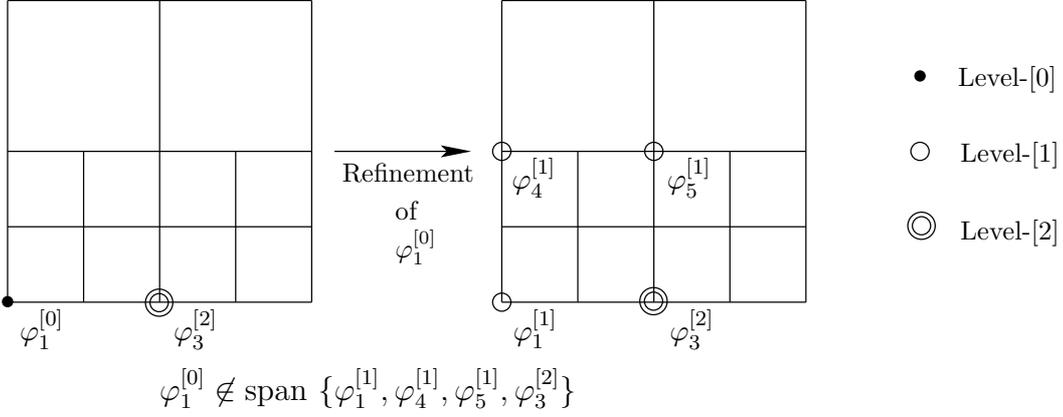A level-$[J]$ basis function cannot be $\mathcal{B}$-refined. Hence, the statement (H_J) is straightforward.

$$\varphi_1^{[0]} \notin \text{span } \{\varphi_1^{[1]}, \varphi_4^{[1]}, \varphi_5^{[1]}, \varphi_3^{[2]}\}$$

FIGURE 11. Refinement of multilevel basis is not lossless.

Let $j \in [\![0, J-1]\!]$. Assume that the statement (H_{j+1}) holds and let $\varphi_k^{[j]}$ be a level-$[j]$ $\mathcal{B}$-refined basis function. Owing to Proposition 1.18, we have

$$\varphi_k^{[j]} = \sum_{\substack{\ell \mid \varphi_\ell^{[j+1]} \text{ is} \\ \text{a child of } \varphi_k^{[j]}}} \beta_{k\ell}^{[j]} \varphi_\ell^{[j+1]}.$$

Moreover, property $(\mathcal{P}_{\text{LO}})$ implies that all basis functions $\varphi_\ell^{[j+1]}$ involving in the above sum are either in $\mathcal{B}$ or $\mathcal{B}$-refined. In the last case, the recurrence assumption (H_{j+1}) yields $\varphi_\ell^{[j+1]} \in \text{span } \mathcal{B}$. Hence, $\varphi_k^{[j]} \in \text{span } \mathcal{B}$ and the recurrence is established. □

Theorem 2.8. *Let $\mathcal{B}$ be a multilevel basis satisfying property $(\mathcal{P}_{\text{LO}})$ and obtained from the multilevel basis $\mathcal{B}^\star$ by the refinement procedure (Algorithm 2.4), then*

$$\text{span } \mathcal{B}^\star \subset \text{span } \mathcal{B}.$$

Proof. Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of the basis function $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$. Let $\varphi_k^{[j]} \in \mathcal{B}^\star$.

- If $\varphi_k^{[j]} \neq \varphi_{k_0}^{[j_0]}$ then $\varphi_k^{[j]} \in \mathcal{B}$,
- else $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$ which is $\mathcal{B}$-refined. Proposition 2.7 ensure that $\varphi_k^{[j]} \in \text{span } \mathcal{B}$.

Then span $\mathcal{B}^\star \subset \text{span } \mathcal{B}$. □

Moreover, the refinement and unrefinement procedures preserve the property $(\mathcal{P}_{\text{LO}})$. Let us begin with the following lemma.

Lemma 2.9. *Let $\mathcal{B}^\star$ and $\mathcal{B}$ be two multilevel basis. Let $\varphi_{k^\star}^{[j^\star]}$ be a $\mathcal{B}^\star$-refined basis function and $\varphi_k^{[j]}$ be a $\mathcal{B}$-refined basis function.*

1) *If $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of a basis function $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$ then*
   (i) *$\varphi_{k^\star}^{[j^\star]}$ is also $\mathcal{B}$-refined,*
   (ii) *$\varphi_k^{[j]}$ is either $\mathcal{B}^\star$-refined or equal to $\varphi_{k_0}^{[j_0]}$.*
2) *If $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of an only once $\mathcal{B}^\star$-refined basis function $\varphi_{k_0}^{[j_0]}$ then*
   (i) *$\varphi_{k^\star}^{[j^\star]}$ is either $\mathcal{B}$-refined or is equal to $\varphi_{k_0}^{[j_0]}$,*
   (ii) *$\varphi_k^{[j]}$ is also $\mathcal{B}^\star$-refined.*

(iii) *Moreover, if we assume that $\varphi_{k^\star}^{[j^\star]}$ is only once $\mathcal{B}^\star$-refined then $\varphi_{k^\star}^{[j^\star]}$ is either also only once $\mathcal{B}$-refined or is equal to $\varphi_{k_0}^{[j_0]}$.*

Proof. Since $\varphi_{k^\star}^{[j^\star]}$ is $\mathcal{B}^\star$-refined, there exists $j^{\star\prime} > j^\star$ and $k^{\star\prime} \in [\![1, N_{\mathrm{dof}}^{[j^{\star\prime}]}]\!]$ such that $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]} \in \mathcal{B}^\star$ and $\boldsymbol{a}_{k^\star}^{[j^\star]} = \boldsymbol{a}_{k^{\star\prime}}^{[j^{\star\prime}]}$. Along the same lines, since $\varphi_k^{[j]}$ is $\mathcal{B}$-refined, there exists $j' > j$ and $k' \in [\![1, N_{\mathrm{dof}}^{[j']}]\!]$ such that $\varphi_{k'}^{[j']} \in \mathcal{B}$ and $\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k'}^{[j']}$.

1) Refinement of $\varphi_{k_0}^{[j_0]}$.
   (i) If $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]} \in \mathcal{B}$ then $\varphi_{k^\star}^{[j^\star]}$ is $\mathcal{B}$-refined; else $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]} = \varphi_{k_0}^{[j_0]}$ and since $\varphi_{k_0}^{[j_0]}$ is $\mathcal{B}$-refined, a fortiori $\varphi_{k^\star}^{[j^\star]}$ is $\mathcal{B}$-refined.
   (ii) If $\varphi_{k'}^{[j']} \in \mathcal{B}^\star$ then $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined and the claim is proved. Otherwise, $\varphi_{k'}^{[j']}$ is a child of $\varphi_{k_0}^{[j_0]}$, say $\varphi_{\ell_0}^{[j_0+1]}$, i.e. $j' = j_0 + 1$, $k' = \ell_0$. We have $\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{\ell_0}^{[j_0+1]}$ and $j_0 + 1 > j$, then owing to Lemma 2.3, $\boldsymbol{a}_k^{[j]} = \boldsymbol{a}_{k_0}^{[j_0]}$. If $j_0 = j$ then $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$; else $j_0 > j$, since $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$, $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined.
2) Unrefinement of $\varphi_{k_0}^{[j_0]}$.
   (i) If $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]} \in \mathcal{B}$ then $\varphi_{k^\star}^{[j^\star]}$ is $\mathcal{B}$-refined and the claim is proved. Otherwise, $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]}$ is a child of $\varphi_{k_0}^{[j_0]}$, say $\varphi_{\ell_0}^{[j_0+1]}$, i.e. $j^{\star\prime} = j_0 + 1$, $k^{\star\prime} = \ell_0$. We have $\boldsymbol{a}_{k^\star}^{[j^\star]} = \boldsymbol{a}_{\ell_0}^{[j_0+1]}$ and $j_0 + 1 > j^\star$, then owing to Lemma 2.3, $\boldsymbol{a}_{k^\star}^{[j^\star]} = \boldsymbol{a}_{k_0}^{[j_0]}$. If $j_0 = j^\star$ then $\varphi_{k^\star}^{[j^\star]} = \varphi_{k_0}^{[j_0]}$; else $j_0 > j^\star$, since $\varphi_{k_0}^{[j_0]} \in \mathcal{B}$, $\varphi_{k^\star}^{[j^\star]}$ is $\mathcal{B}$-refined.
   (ii) If $\varphi_{k'}^{[j']} \in \mathcal{B}^\star$ then $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined; else $\varphi_{k'}^{[j']} = \varphi_{k_0}^{[j_0]}$ and since $\varphi_{k_0}^{[j_0]}$ is $\mathcal{B}^\star$-refined, a fortiori $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined.
   (iii) Here, we assume that $j^{\star\prime} = j^\star + 1$. If $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]} \in \mathcal{B}$ then $\varphi_{k^\star}^{[j^\star]}$ is only once $\mathcal{B}$-refined and the claim is proved. Otherwise, $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]}$ is a child of $\varphi_{k_0}^{[j_0]}$. However, owing to Remark 2.2, $\varphi_{k^\star}^{[j^\star]}$ is the unique parent of $\varphi_{k^{\star\prime}}^{[j^{\star\prime}]}$. Hence, we get $\varphi_{k^\star}^{[j^\star]} = \varphi_{k_0}^{[j_0]}$.

$\square$

Proposition 2.10. *The refinement and unrefinement procedures of multilevel basis (in the sense of Algorithm 2.4) preserves the property* $(\mathcal{P}_{\mathrm{LO}})$.

Proof. Assume that the multilevel basis $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LO}})$. Let $\varphi_k^{[j]}$ be a $\mathcal{B}$-refined basis function and $\varphi_\ell^{[j+1]}$ be a child of $\varphi_k^{[j]}$. We have to show that $\varphi_\ell^{[j+1]}$ either belong to $\mathcal{B}$ or is $\mathcal{B}$-refined.

- Refinement: Assume that the multilevel basis $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of a basis function $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$. By Lemma 2.9 property 1) (ii), $\varphi_k^{[j]}$ is either $\mathcal{B}^\star$-refined or equal to $\varphi_{k_0}^{[j_0]}$. Consider the two cases:
   - If $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LO}})$, only the two following cases are possible:
     * $\varphi_\ell^{[j+1]}$ is $\mathcal{B}^\star$-refined. Owing to Lemma 2.9 property 1)(i), this implies that $\varphi_\ell^{[j+1]}$ is $\mathcal{B}$-refined.
     * $\varphi_\ell^{[j+1]} \in \mathcal{B}^\star$. And then, $\varphi_\ell^{[j+1]} \in \mathcal{B}$ or $\varphi_\ell^{[j+1]} = \varphi_{k_0}^{[j_0]}$ which is $\mathcal{B}$-refined.
   - If $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$, all its children are either in $\mathcal{B}$ or are $\mathcal{B}^\star$-refined. Owing to Lemma 2.9 property 1)(i), they are either in $\mathcal{B}$ or $\mathcal{B}$-refined.
- Unrefinement: Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of an only once refined basis function $\varphi_{k_0}^{[j_0]}$ of $\mathcal{B}^\star$. By Lemma 2.9 property 2) (ii), $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{LO}})$, only the two following cases are possible:
   - $\varphi_\ell^{[j+1]}$ is $\mathcal{B}^\star$-refined. Owing to Lemma 2.9 property 2)(i), this implies that $\varphi_\ell^{[j+1]}$ is $\mathcal{B}$-refined or $\varphi_\ell^{[j+1]} = \varphi_{k_0}^{[j_0]} \in \mathcal{B}$.
   - $\varphi_\ell^{[j+1]} \in \mathcal{B}^\star$. And then, we have $\varphi_\ell^{[j+1]} \in \mathcal{B}$ or $\varphi_\ell^{[j+1]}$ is a child of $\varphi_{k_0}^{[j_0]}$ with no other $\mathcal{B}^\star$-refined parent. The last case is impossible. Indeed, $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined and then, by unicity, we would

have $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$. However, $\varphi_{k_0}^{[j_0]}$ is not a $\mathcal{B}$-refined function. This is a contradiction and we get $\varphi_\ell^{[j+1]} \in \mathcal{B}$.

$\square$

Remark 2.11. Note that the properties $(\mathcal{P}_{\mathrm{LI}})$ or $(\mathcal{P}_{\mathrm{LO}})$ are not so restrictive since they are preserved by Un/Refinement procedures and since it is straightforward to see that they are satisfied by the coarse basis B_0 which is used, in practice, for starting the adaptation algorithm.

2.1.3. *Adaptation procedure*

Proposition 2.12. *Let $\mathcal{B}^\star$ be a multilevel basis.*

1) *Let $\mathcal{E}^\star \subset \mathcal{B}^\star$. It is possible to refine successively all basis functions belonging to $\mathcal{E}^\star$, hence producing a multilevel basis $\mathcal{B}$ which is independent of the order in which the basis functions were refined.*
   We say that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of the set of basis functions $\mathcal{E}^\star$.

2) *Let $\mathcal{F}^\star$ be a set of only once $\mathcal{B}^\star$-refined basis functions which have no $\mathcal{B}^\star$-refined child. It is possible to unrefine successively all basis functions belonging to $\mathcal{F}^\star$, hence producing a multilevel basis $\mathcal{B}$ which is independent of the order in which the basis functions were unrefined.*
   We say that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of the set of basis functions $\mathcal{F}^\star$.

Proof. In the two cases, we first have to prove that successive (un)refinements are possible and then that the obtained multilevel basis $\mathcal{B}$ is independent of the order in which the basis functions are (un)refined.

1) – Let $\varphi \in \mathcal{E}^\star$. The set $\mathcal{E}^\star \backslash \{\varphi\}$ is included in the multilevel basis produced by the refinement of φ since, in this procedure, only φ is removed from $\mathcal{B}^\star$. Hence, all basis functions in $\mathcal{E}^\star \backslash \{\varphi\}$ can be then refined.
    – It is sufficient to prove that $\mathcal{B}$ is independent of the order in which the basis functions are refined in the case where $\#\mathcal{E}^\star = 2$, say $\mathcal{E}^\star = \{\varphi, \psi\}$. Denote by $\overline{\mathcal{B}}$ the multilevel basis obtained by the refinement of φ in $\mathcal{B}^\star$ and then by $\mathcal{B}$ the multilevel basis obtained by the refinement of ψ in $\overline{\mathcal{B}}$. By definition, we have

$$\overline{\mathcal{B}} = \mathcal{B}^\star \backslash \{\varphi\} \cup \{\text{children of } \varphi \text{ not } \mathcal{B}^\star\text{-refined}\} \tag{5}$$

and

$$\mathcal{B} = \overline{\mathcal{B}} \backslash \{\psi\} \cup \{\text{children of } \psi \text{ not } \overline{\mathcal{B}}\text{-refined}\} \tag{6}$$

Applying Lemma 2.9 property 1) (i) and (ii), we get

$$\{\overline{\mathcal{B}}\text{-refined basis functions}\} = \{\mathcal{B}^\star\text{-refined basis functions}\} \cup \{\varphi\} \tag{7}$$

Hence, combining (6) and (7), we obtain

$$\mathcal{B} = \overline{\mathcal{B}} \backslash \{\psi\} \cup \{\text{children of } \psi \text{ not } \mathcal{B}^\star\text{-refined and different from } \varphi\} \tag{8}$$

Combining (5) and (8), we get

$$\mathcal{B} = \left(\mathcal{B}^\star \cup \{\text{children of } \varphi \text{ and } \psi \text{ which are not } \mathcal{B}^\star\text{-refined}\}\right) \backslash \{\varphi, \psi\}.$$

This expression shows that $\mathcal{B}$ do not depend of the order in which the basis functions φ and ψ were refined.

2) – Let $\varphi \in \mathcal{F}^\star$. Denote by $\overline{\mathcal{B}}$ the multilevel basis obtained by the unrefinement of φ in $\mathcal{B}^\star$. We have to prove that all basis functions of $\mathcal{F}^\star \backslash \{\varphi\}$ can be unrefined in $\overline{\mathcal{B}}$, i.e. that any basis function of $\mathcal{F}^\star \backslash \{\varphi\}$ is only once $\overline{\mathcal{B}}$-refined and has no $\overline{\mathcal{B}}$-refined children. Let $\psi \in \mathcal{F}^\star \backslash \{\varphi\}$. Owing to Lemma 2.9 property 2) (iii), since ψ is only once $\mathcal{B}^\star$-refined basis function different from φ, ψ is only once $\overline{\mathcal{B}}$-refined. Furthermore, if ψ has a $\overline{\mathcal{B}}$-refined child then applying Lemma 2.9 property 2) (i), this child is also $\mathcal{B}^\star$-refined. This a contradiction and the claim is proved.

– It is sufficient to prove that $\mathcal{B}$ is independent of the order in which the basis functions are unrefined in the case where $\#\mathcal{F}^\star = 2$, say $\mathcal{F}^\star = \{\varphi, \psi\}$. Denote by $\overline{\mathcal{B}}$ the multilevel basis obtained from $\mathcal{B}^\star$ by the unrefinement of φ and then by $\mathcal{B}$ the multilevel basis obtained from $\overline{\mathcal{B}}$ by the unrefinement of ψ. By definition, we have

$$\overline{\mathcal{B}} = \mathcal{B}^\star \backslash \{\text{children of } \varphi \text{ which have no other } \mathcal{B}^\star\text{-refined parent}\} \cup \{\varphi\} \tag{9}$$

and
$$\mathcal{B} = \overline{\mathcal{B}} \backslash \{\text{children of } \psi \text{ which have no other } \overline{\mathcal{B}}\text{-refined parent}\} \cup \{\psi\} \tag{10}$$

Applying Lemma 2.9 property 2) (i) and (ii), we get

$$\{\mathcal{B}^\star\text{-refined basis functions}\} = \{\overline{\mathcal{B}}\text{-refined basis functions}\} \cup \{\varphi\} \tag{11}$$

Hence, combining (10) and (11), we obtain

$$\mathcal{B} = \overline{\mathcal{B}} \backslash \{\text{children of } \psi \text{ which have no } \mathcal{B}^\star\text{-refined parent except possibly } \varphi\} \cup \{\psi\} \tag{12}$$

Combining (9) and (12), we get

$$\mathcal{B} = \mathcal{B}^\star \backslash \{\text{children of } \varphi \text{ and } \psi \text{ which have no } \mathcal{B}^\star\text{-refined parents except possibly } \varphi \text{ and } \psi\} \cup \{\varphi, \psi\}.$$

This expression shows that $\mathcal{B}$ do not depend of the order in which the basis functions φ and ψ were refined.

$\square$

With this definition at hand, we can give the refinement algorithm.

Algorithm 2.13 (Adaptation procedure). *Let $\mathcal{B}^\star$ a multilevel basis. Assume that, thanks to a refinement criterion, we are given the set $\mathcal{E}^\star \subset \mathcal{B}^\star$ of basis functions to refine and the set $\mathcal{F}^\star$ of only once $\mathcal{B}^\star$-refined basis functions (without $\mathcal{B}^\star$-refined children) to unrefine. The adaptation procedure consists in the two following steps:*

1) *Refine the set $\mathcal{E}^\star$, thus producing a new multilevel basis $\overline{\mathcal{B}}$.*
2) *Unrefine the set of basis functions of $\mathcal{F}^\star$ which are still only once $\overline{\mathcal{B}}$-refined basis functions without $\overline{\mathcal{B}}$-refined children.*

2.1.4. *One-level-difference rule*

In practice, the following criterion is used to ensure the common rule of "one-level-difference refinement" [15].

Criterion 2.14 (One-level-difference refinement rule). *Let $\mathcal{B}^\star$ a multilevel basis. A basis function of $\mathcal{B}^\star$ may be refined only if all its parents are $\mathcal{B}^\star$-refined.*

Typically, this criterion is used to avoid an important difference of refinement level between "neighboring" basis functions. An example of refinement sequence forbidden by this criterion is given in Figure 12.

This criterion can be taken into account by adding a recursive step in the refinement procedure: to refine a basis function, we first refine all its parents which are not $\mathcal{B}^\star$-refined and then we refine the basis function itself. This is illustrated, for instance, by the spreading of the level-[1] refined area between the first two pictures in Figure 17.

2.2. **Multigrid preconditioner**

In this section, we assume that we have the following genuine variational problem to solve: Find $u \in \mathcal{V}_h$ such that

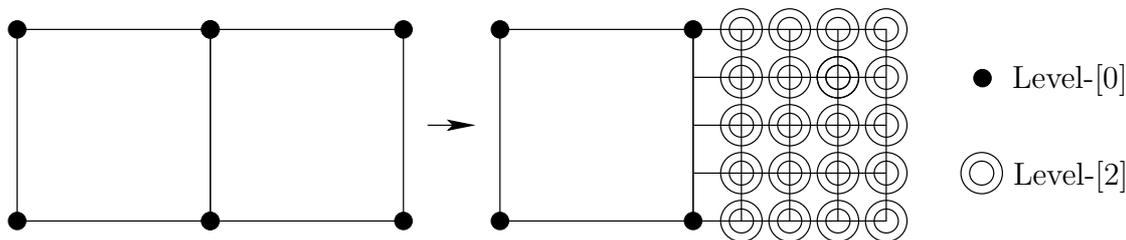$$\forall v \in \mathcal{V}_h, \ a(u, v) = l(v),$$

FIGURE 12. Refinement sequence forbidden by the One-level-difference rule.

where $\mathcal{V}_h = \text{span } \mathcal{B}$ is a multilevel approximation space, $a : \mathrm{H}^1(\Omega) \times \mathrm{H}^1(\Omega) \to \mathbb{R}$ is a bilinear continuous and coercive form and $b : \mathrm{H}^1(\Omega) \to \mathbb{R}$ is a linear continuous form. We denote by A_J the stiffness matrix associated to this problem, that is

$$A_J = [a(\varphi, \psi)]_{\varphi, \psi \in \mathcal{B}}.$$

The multilevel space $\mathcal{V}_h$ is built to achieve a given level of accuracy without increasing too much the number of degrees of freedom in the discrete problem. In fact, we are going to describe how to naturally take advantage of the multilevel structure of the approximation space in order to build a sequence of nested multilevel grids finally leading to a multigrid preconditioner.

2.2.1. *Coarsening*

From a given "fine" multilevel basis $\mathcal{B}_F$, the following algorithm is used to construct a "coarser" multilevel basis $\mathcal{B}_C$.

Algorithm 2.15 (Coarsening). *Let $\mathcal{B}_F$ a multilevel basis. Let $j_M = \max\{j \in [\![0, J]\!]; \mathcal{B}_F \cap \mathrm{B}_j \neq \emptyset\}$ be the highest refinement level in $\mathcal{B}_F$. A "coarser" multilevel basis $\mathcal{B}_C$, denoted by $\mathcal{B}_C = \text{coarsen}(\mathcal{B}_F)$, is obtained from $\mathcal{B}_F$ by unrefinement (Algorithm 2.4) of the set of $\mathcal{B}_F$-refined basis functions of level $[j_M - 1]$.*

The following proposition gives an equivalent formulation of the above algorithm.

Proposition 2.16. *Assume that $\mathcal{B}_F$ is a multilevel basis satisfying the following property:*

> *Any level-$[j]$ basis function, $j \geqslant 1$, which either belongs to $\mathcal{B}$ or is $\mathcal{B}$-refined, has at least one $\mathcal{B}$-refined parent .*   ($\mathcal{P}_{\mathrm{HI}}$)

Let $j_M = \max\{j \in [\![0, J]\!]; \mathcal{B}_F \cap \mathrm{B}_j \neq \emptyset\}$. The multilevel basis $\mathcal{B}_C = \text{coarsen}(\mathcal{B}_F)$ defined in Algorithm 2.15 can be obtained by the following equivalent algorithm:
> - *remove all level-$[j_M]$ basis functions of $\mathcal{B}_F$,*
> - *add all $\mathcal{B}_F$-refined basis functions of level $[j_M - 1]$.*

Proof. Note first that any step in Algorithm 2.15 consists in an unrefinement of a level-$[j_M - 1]$ basis function, this implies that added basis functions are certainly on level-$[j_M - 1]$ and that removed basis functions are certainly on level-$[j_M]$. Hence, the set of added and removed basis functions are disjoint. In Algorithm 2.15, a basis function which is removed (or added) by an unrefinement procedure can not be added (or respectively removed) by an other unrefinement procedure. Furthermore, remark that by definition of j_M, there is no $\mathcal{B}_F$-refined function on level-$[j_M]$. Hence, since they have no $\mathcal{B}_F$-refined children, all $\mathcal{B}_F$-refined basis functions can actually be unrefined. Unrefinement of a basis functions involved that it is added and then the set of added basis functions in Algorithm 2.15 is exactly the set of $\mathcal{B}_F$-refined basis functions. It remains to show that all level-$[j_M]$ basis functions of $\mathcal{B}_F$ are removed. Arguing by contradiction, assume that a level-$[j_M]$ basis function belongs to $\text{coarsen}(\mathcal{B}_F)$. Since unrefinement preserves the property $(\mathcal{P}_{\mathrm{HI}})$ (Proposition 2.17), this basis function has at least one $\text{coarsen}(\mathcal{B}_F)$-refined parent. Owing to Lemma 2.9 property 2) (ii), this is also a $\mathcal{B}_F$- refined basis function (of level-$[j_M - 1]$). This is a contradiction and the claim is proved.   □

The property $(\mathcal{P}_{\mathrm{HI}})$ ensures the desirable fact that all level-$[j_M]$ basis functions of $\mathcal{B}_F$ are removed. This additional property is not so restrictive since it is preserved by the refinement and unrefinement procedures.

Proposition 2.17. *The refinement and unrefinement procedures described in Algorithm 2.4 preserve the property $(\mathcal{P}_{\mathrm{HI}})$.*

Proof. Assume that $\mathcal{B}^\star$ is a multilevel basis satisfying the property $(\mathcal{P}_{\mathrm{HI}})$ and let $j \in [\![1, J]\!]$.

- Consider first a basis function $\varphi_k^{[j]}$ which belongs to $\mathcal{B}$.
  - Refinement: Assume that the multilevel basis $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of a basis function $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$.
    * Case 1: $\varphi_k^{[j]} \in \mathcal{B}^\star$. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{HI}})$, $\varphi_k^{[j]}$ has at least one $\mathcal{B}^\star$-refined parent. Owing Lemma 2.9 property 1), this parent is $\mathcal{B}$-refined.
    * Case 2: $\varphi_k^{[j]}$ is a child of $\varphi_{k_0}^{[j_0]}$ and $\varphi_{k_0}^{[j_0]}$ is $\mathcal{B}$-refined.
  - Unrefinement: Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of an only once refined basis function $\varphi_{k_0}^{[j_0]}$ of $\mathcal{B}^\star$.
    * Case 1: $\varphi_k^{[j]} \in \mathcal{B}^\star \backslash \{\text{children of } \varphi_k^{[j]} \text{ without other } \mathcal{B}^\star\text{-refined parent}\}$.
      Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{HI}})$, $\varphi_k^{[j]}$ has at least one $\mathcal{B}^\star$-refined parent. Owing Lemma 2.9 property 2), either this parent is $\mathcal{B}$-refined or this is $\varphi_{k_0}^{[j_0]}$.
      In the first case, the proof is finished, in the second case $\varphi_k^{[j]}$ is a child of $\varphi_{k_0}^{[j_0]}$ which belongs to $\mathcal{B}^\star \backslash \{\text{children of } \varphi_k^{[j]} \text{ without other } \mathcal{B}^\star\text{-refined parent}\}$. Therefore, $\varphi_k^{[j]}$ has an other $\mathcal{B}^\star$-refined parent. This parent is then $\mathcal{B}$-refined because it is not $\varphi_{k_0}^{[j_0]}$.
    * Case 2: $\varphi_k^{[j]} = \varphi_{k_0}^{[j_0]}$, $\varphi_{k_0}^{[j_0]}$ is $\mathcal{B}^\star$-refined. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{HI}})$, $\varphi_{k_0}^{[j_0]}$ has at least one $\mathcal{B}^\star$ refined parent. Owing Lemma 2.9 property 2), this parent is $\mathcal{B}$-refined because it can not be $\varphi_{k_0}^{[j_0]}$.
- Consider now a basis function $\varphi_k^{[j]}$ which is $\mathcal{B}$-refined.
  - Refinement: Assume that the multilevel basis $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the refinement of a basis function $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$. By Lemma 2.9 property 1) (ii), $\varphi_k^{[j]}$ is either $\mathcal{B}^\star$-refined or equal to $\varphi_{k_0}^{[j_0]} \in \mathcal{B}^\star$. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{HI}})$, in the two cases, $\varphi_k^{[j]}$ has at least one $\mathcal{B}^\star$-refined parent. Owing to Lemma 2.9 property 1)(i), this parent is also $\mathcal{B}$-refined.
  - Unrefinement: Assume that $\mathcal{B}$ is obtained from $\mathcal{B}^\star$ by the unrefinement of an only once refined basis function $\varphi_{k_0}^{[j_0]}$ of $\mathcal{B}^\star$ without $\mathcal{B}^\star$-refined children. By Lemma 2.9 property 2) (ii), $\varphi_k^{[j]}$ is $\mathcal{B}^\star$-refined. Since $\mathcal{B}^\star$ satisfies the property $(\mathcal{P}_{\mathrm{HI}})$, $\varphi_k^{[j]}$ has at least one $\mathcal{B}^\star$-refined parent. Owing to Lemma 2.9 property 2) (i), either this parent is $\mathcal{B}$-refined or is equal to $\varphi_{k_0}^{[j_0]}$. The last case is impossible because $\varphi_{k_0}^{[j_0]}$ has no $\mathcal{B}^\star$-refined children. $\square$

The last main property of the coarsening procedure, is that it produced nested vector spaces.

Proposition 2.18. *Let $\mathcal{B}_F$ be a multilevel basis satisfying the property $(\mathcal{P}_{\mathrm{LO}})$ and $\mathcal{B}_C = \mathrm{coarsen}(\mathcal{B}_F)$. The following embedding holds:*

$$\mathrm{span}\, \mathcal{B}_C \subset \mathrm{span}\, \mathcal{B}_F.$$

Proof. Let $\varphi \in \mathcal{B}_C$. If $\varphi \notin \mathcal{B}_F$ then φ has been unrefined and so this is a $\mathcal{B}_F$-refined basis function of level $[j_M - 1]$. Owing Proposition 2.7, $\varphi \in \mathrm{span}\, \mathcal{B}_F$. $\square$

Remark 2.19. Let us notice that, thanks to the refinement equation, it is straightforward to build the matrix representation in the corresponding basis of the natural embedding from $\mathrm{span}\, \mathcal{B}_C$ to $\mathrm{span}\, \mathcal{B}_F$. Indeed, if $\varphi \in \mathcal{B}_C$, we have

- either $\varphi \in \mathcal{B}_F$
- or all the children of φ belong to $\mathcal{B}_F$ and then, the refinement equation yields an expression of φ as a linear combination of elements in $\mathcal{B}_F$.

Remark 2.20. Note that properties $(\mathcal{P}_{\mathrm{LO}})$ and $(\mathcal{P}_{\mathrm{HI}})$ used in this section are not so restrictive since they are preserved by Un/Refinement procedures and since it is straightforward to see that they are satisfied by the coarse basis B_0 which is used, in practice, to start the adaptation algorithm. (see also Remark 2.11)

2.2.2. *Multigrid framework*

We recursively define a sequence $\{V_0, \dots, V_J\}$ of nested spaces built upon $\mathcal{V}_h$ as follows:

- we first take $\mathcal{B}_J = \mathcal{B}$ and $V_J = \mathrm{span}\, \mathcal{B}_J = \mathcal{V}_h$,
- then, for $k = J, \dots, 1$, we define a coarser multilevel basis $\mathcal{B}_{k-1}$ from $\mathcal{B}_k$ by:

$$\mathcal{B}_{k-1} = \mathrm{coarsen}(\mathcal{B}_k),$$

and the corresponding multilevel approximation space:

$$V_{k-1} = \mathrm{span}\, \mathcal{B}_{k-1}.$$

Owing to Proposition 2.18, we have:

$$V_0 \subset V_1 \subset \cdots \subset V_J.$$

Note that the auxiliary sequence $V_0 \subset \cdots \subset V_J$ introduced here usually do not reflect the dynamic refinement process, although a such sequence can always be a posteriori deduced from any multilevel approximation space V_J. An example with four refinement level is given in Figure 13.
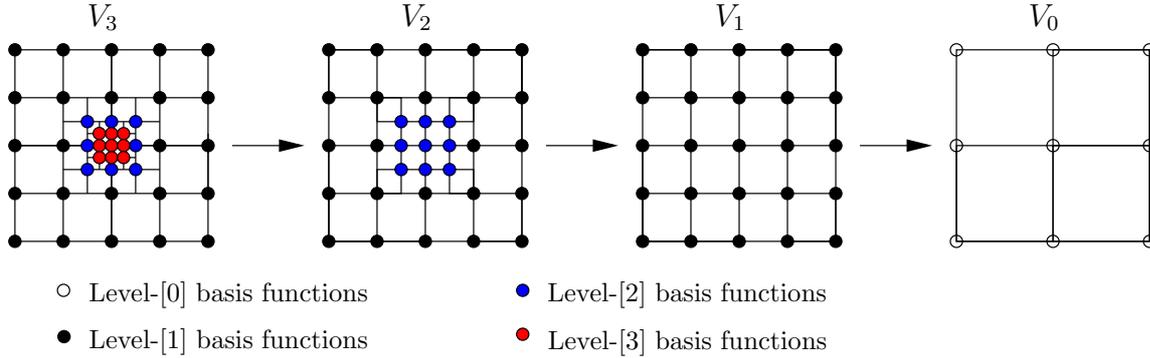

○ Level-[0] basis functions   ● Level-[2] basis functions

● Level-[1] basis functions   ● Level-[3] basis functions

FIGURE 13. Example of coarsening : from V_3 to V_0.

We do not need to know explicitly any information about the spaces V_k, except the intergrids operators. Owing to Remark 2.19, it is staightforward to construct the matrix representation, denoted by I_{k-1}^k, of the natural embedding from V_{k-1} to V_k in the basis $\mathcal{B}_{k-1}$ and $\mathcal{B}_k$. Then, intergrids operators are defined in the following way, for all $k \in [\![0, J]\!]$:

$$I_k = I_{J-1}^J I_{J-2}^{J-1} \cdots I_k^{k+1}$$

We can also define approximate operators on each space V_k, for all $k \in [\![0, J]\!]$ by:

$$A_k = I_k^t A_J I_k$$

At last, in the sequel, we used Jacobi and Gauss-Seidel smoothers defined for all $k \in [\![0, J]\!]$ as follows:

- Jacobi: $S_k = D_k$ where D_k is the diagonal part of A_k.
- Gauss-Seidel: $S_k = T_k$ where T_k is the superior triangular part of A_k.

2.2.3. *Multigrid Algorithm*

We used the two following multigrid preconditioners:

- Additive version [8]:

$$P_a = \sum_{k=0}^{J} I_k S_k I_k^t$$

  where S_k, $k \in [\![0, J]\!]$, is the Jacobi smoother.
- Multiplicative version [12]: This correspond to the classical V-cycle. In this section S_k is the Gauss-Seidel smoother. We define recursively, for all $k \in [\![0, J]\!]$, the linear operator $MG_k : \mathbb{R}^{\# \mathcal{B}_k} \to \mathbb{R}^{\# \mathcal{B}_k}$. We first set $MG_0 = A_0^{-1}$ and for all $k \in [\![1, J]\!]$, we define $MG_k(f_k)$, $f_k \in \mathbb{R}^{\# \mathcal{B}_k}$ by the following steps:
  - (0) $v_k \longleftarrow 0$ Initialisation
  - (1) $v_k \longleftarrow v_k + S_k(f_k - A_k v_k)$,   Pre-smoothing step
  - (2) $v_k \longleftarrow v_k + I_{k-1} MG_{k-1}(I_{k-1}^t (f_k - A_k v_k))$,   Coarse Grid Correction
  - (3) $v_k \longleftarrow v_k + S_k(f_k - A_k v_k)$,   Post-smoothing step
    Set $MG_k(f_k) = v_k$.

  The multiplicative preconditioner is then

$$P_m = MG_J.$$

2.3. **Validation on a stationnary model problem**

We first validate the local refinement method and the multigrid preconditioner on the following stationnary model problem.

The practical implementation has been performed using the software object-oriented component library PELICANS [19], developed at the "Institut de Radioprotection et de Sûreté Nucléaire (IRSN)" and distributed under the CeCILL-C license agreement (an adaptation of LGPL to the French law).

2.3.1. *Continuous model problem*

Let $\Omega = [0, 1]^d$. Consider the Laplace problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\Delta u & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial \Omega. \end{cases} \tag{13}$$

The source term f is chosen so that the exact solution u is:

$$\forall \mathbf{x} \in \mathbb{R}^d, \ u(\mathbf{x}) = H_\varepsilon \Big(R - |\mathbf{x} - \mathbf{x}_C| \Big),$$

where $R \in \mathbb{R}$, $\varepsilon \in \mathbb{R}_+^*$, $\mathbf{x}_C \in \mathbb{R}^d$ are parameters and $H_\varepsilon : \mathbb{R} \longrightarrow \mathbb{R}$ is defined by

$$\forall x \in \mathbb{R}, \ H_\varepsilon(x) = \begin{cases} 0 & \text{if } x < -\varepsilon, \\ \dfrac{1}{2} \left[1 + \dfrac{x}{\varepsilon} + \dfrac{1}{\pi} sin\left(\pi \dfrac{x}{\varepsilon}\right) \right] & \text{if } |x| \leqslant \varepsilon, \\ 1 & \text{if } x > \varepsilon. \end{cases}$$

The function H_ε is represented in Figure 14 and the interpretation of parameters R, ε, $\mathbf{x}_C$ is explained in Figure 15 which represents the exact solution u when $d = 1$.

For numerical simulations given in the sequel, we have set:

$$\Omega = [0, 1]^d, \quad \mathbf{x}_C = (0.5, 0.5), \quad R = 0.3 \quad \text{and} \quad \varepsilon = 0.1$$
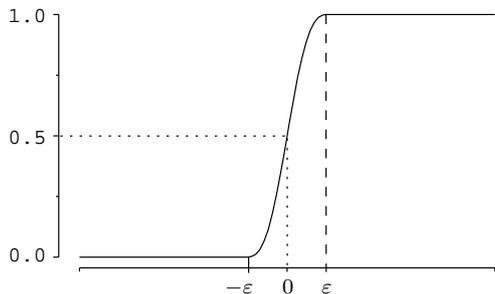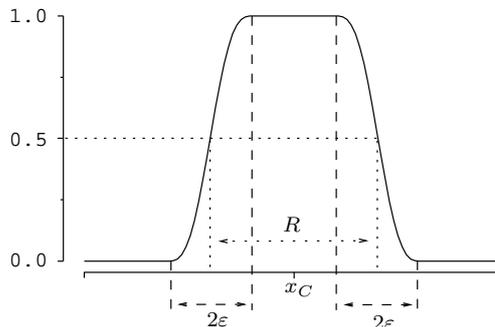
FIGURE 14. Function H_ε.

FIGURE 15. Exact solution of Problem 13, $d = 1$.

2.3.2. *Initial meshes and refinement criterion*
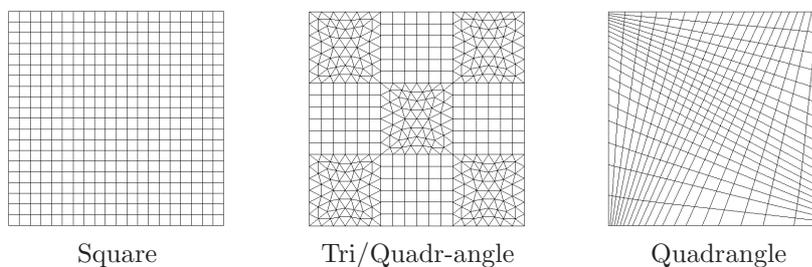

Square   Tri/Quadr-angle   Quadrangle

FIGURE 16. Initial meshes.

In order to show the capability of the local refinement method, we used several types of meshes: square, triangle, general quadrangle, in two dimension and cube in three dimension. Figure 16 shows the initial meshes, i.e. before refinement steps, which have been used for the validation in two dimension. In particular, note that the method allows to combine several kind of geometric elements (triangle $\mathbb{P}_1$ and square $\mathbb{Q}_1$ in the example here) provided that the corresponding reference elements, and refinement patterns are compatible. In three dimension, the initial mesh is a cubic regular mesh obtained by dividing all sides in 15 segments. In this section, the shape of the exact solution is known. This is the reason why a geometric refinement criterion is used. It consists in choosing a priori a number of refinement steps and the location of the nodes associated to the basis functions to refine. Here the following criterion is used:

Criterion 2.21 (Geometric criterion). $\varphi_k^{[j]}$ *will be refined iff*

$$R - \varepsilon < \left| \boldsymbol{a}_k^{[j]} - \mathbf{x}_C \right| < R + \varepsilon$$

2.3.3. *Local refinement and multigrid preconditioner*

For each type of initial mesh, we performed six computations increasing the number of refinement steps from one to six and following the geometric refinement criterion 2.21. In Figures 17, 18, 19 and 20, for smaller number of refinement steps (one, two and three steps), we represent the refined meshes and the plot of the function $u_h(1 - u_h)$ where u_h is the computed approximate solution. Hence, the colored zone in the figures represents the computed "interface", that is the areas where local refinement takes place. Note that meshes in our method are obtained as support of basis functions involved in the formulation of the problem (see Section 3.3.2). In particular, we remark the effect of the "one-level-difference" refinement rule. Indeed, for each type of mesh, the

"one level refinement" areas spreads during the second refinement step. This is due to the fact that during this refinement step, few parents of level-[1] basis functions are refined to enable the refinement of their childs.
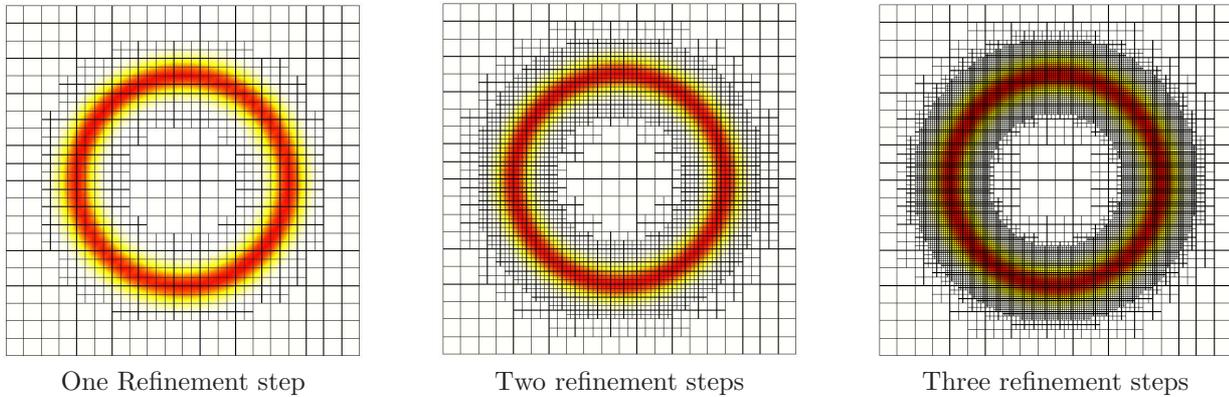

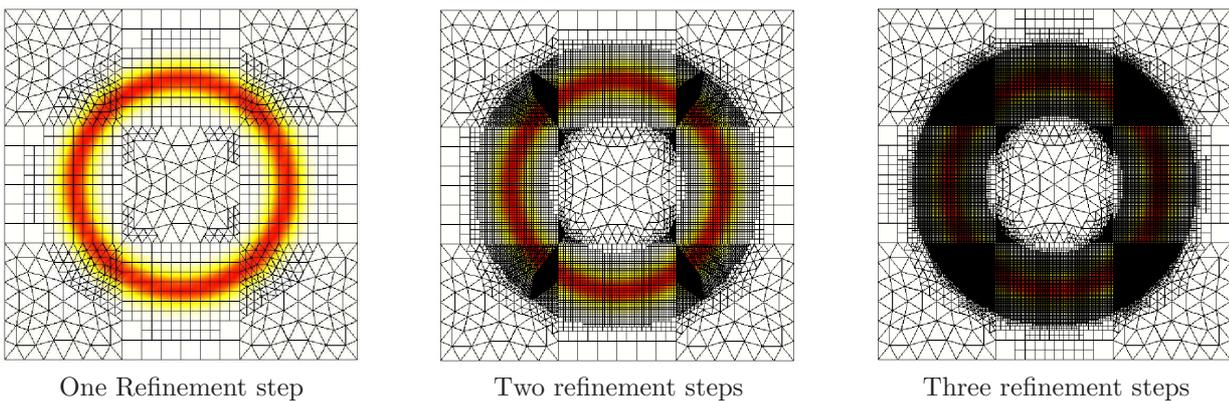| One Refinement step | Two refinement steps | Three refinement steps |

FIGURE 17. Square-$\mathbb{Q}_1$. Refined meshes.

| One Refinement step | Two refinement steps | Three refinement steps |

FIGURE 18. Tri/Quadr-angle-$\mathbb{P}_1$/$\mathbb{Q}_1$. Refined meshes.

Table 2 shows the convergence rates in the L^2-norm, computed with respect to the mesh size of the finest refinement level used in the computation. We obtain a convergence rate in the L^2-norm equal to 2 for first order finite element ($\mathbb{P}_1$ and $\mathbb{Q}_1$) and equal to 3 for second order finite element ($\mathbb{P}_2$ and $\mathbb{Q}_2$) as expected.

In order to illustrate the capability of multigrid preconditioners, we represent in Tables 3, 4 and 5 the number of necessary iterations of the conjugate gradient solver as a function of the number of unknowns. In Table 3, we show the results for two dimensional first order elements, in Table 4, for two dimensional second order elements and in Table 5 for three dimensional first order elements. Various preconditioners are compared : the classical Incomplete LU factorization (ILU0) and the additive (P_a) and multiplicative (P_m) versions of the multigrid algorithm. Note that in all those computations we say that convergence in the conjugate gradient method is achieved as soon as the relative L^∞-norm of the residual is less than 10^{-10}. We limit the total number of iterations to 600, and we denote by "–" in Tables 3, 4 and 5 the case where the convergence of the conjugate gradient method is not achieved before 600 iterations.

Without preconditionning, the number of iterations rapidly exceeds the limit when increasing the number of unknowns. The use of preconditioners enable to reduce the number of iterations and we observe in particular that the number of necessary iterations using (P_m) and (P_a) is almost independent of the size of the problem and is significantly smaller than using, for example, (ILU0) preconditioner.
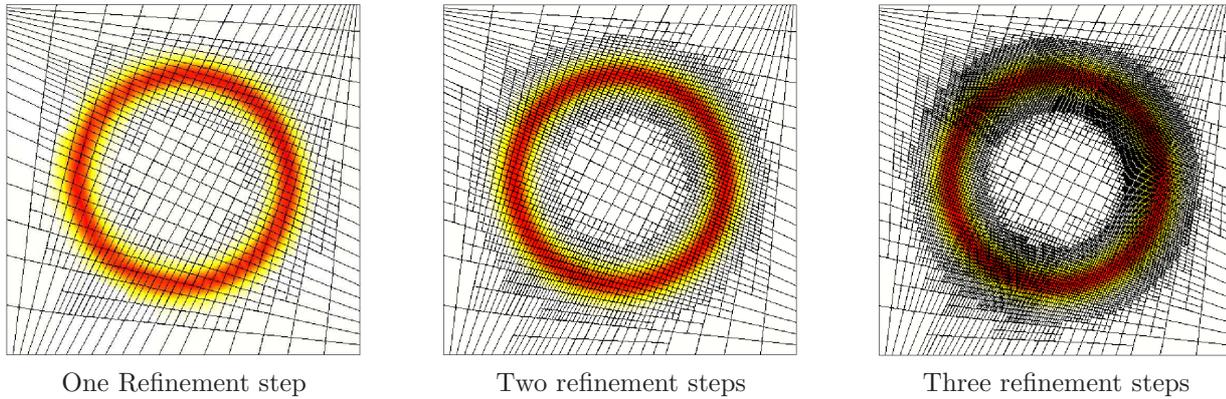
One Refinement step   Two refinement steps   Three refinement steps

FIGURE 19. Quadrangle-$\mathbb{Q}_1$. Refined meshes.
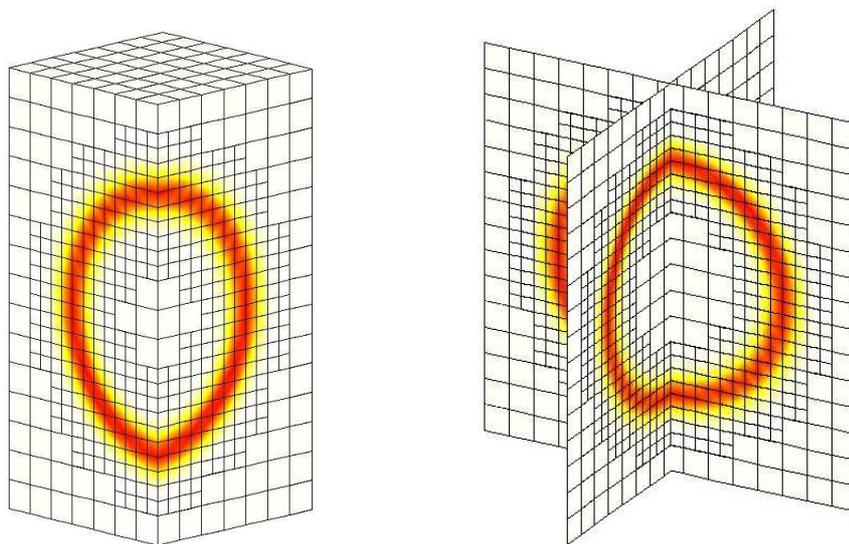
FIGURE 20. Refined mesh in 3D. One refinement step.

3. APPLICATION TO A TERNARY CAHN-HILLIARD SYSTEM

We are interested in the numerical computation of the solution of a three-phase Cahn-Hilliard model proposed in [5]. After a brief introduction of this model, we describe the time and space discretization of the problem that we are using. Then, a numerical example is given using local adaptive refinement and multigrid preconditioner.

3.1. **Introduction**

The Cahn-Hilliard approach consists in assuming that the interfaces between phases in the system have a small but positive thickness ε. Each phase i is represented by a smooth function c_i called the order parameter (which is taken to be the volumic fraction of the component in the mixture). Since the mixture is supposed to be perfect, the three unknowns c_1, c_2 and c_3 are linked though the relationship

$$c_1 + c_2 + c_3 = 1. \tag{14}$$

Dimension	Mesh-Finite Element	Convergence rate
	Square-$\mathbb{Q}_1$	1.99
2D	Quadrangle-$\mathbb{Q}_1$	1.99
	Tri/Quadr-angle-$\mathbb{P}_1/\mathbb{Q}_1$	1.99
3D	Cube-$\mathbb{Q}_1$	1.99
	Square-$\mathbb{Q}_2$	2.99
2D	Quadrangle-$\mathbb{Q}_2$	2.90
	Tri/Quadr-angle-$\mathbb{P}_2/\mathbb{Q}_2$	2.97

TABLE 2. Convergence rates in the L^2-norm, computed with respect to the mesh size of the finest refinement level used in the computation.

Refinement level		1	2	3	4	5	6
	Number of unknowns	893	3053	11021	41757	161233	633629
	No preconditioner	53	99	197	372	–	–
Square-$\mathbb{Q}_1$	ILU0	31	61	114	221	457	–
	P_a	18	23	27	29	33	34
	P_m	8	8	9	9	10	10
	Number of unknowns	935	3217	11609	43593	168347	660710
	No preconditioner	176	357	–	–	–	–
Quadrangle-$\mathbb{Q}_1$	ILU0	45	84	183	388	–	–
	P_a	46	67	83	97	111	120
	P_m	15	16	20	21	23	24
	Number of unknowns	869	2821	9893	37577	144969	569757
	No preconditioner	68	131	279	579	–	–
Tri/Quadr-angle-$\mathbb{P}_1/\mathbb{Q}_1$	ILU0	36	67	131	272	543	–
	P_a	21	29	35	40	43	47
	P_m	9	10	10	11	12	12

TABLE 3. Two dimensional test cases. Number of iterations in the conjugate gradient solver as a function of the number of unknowns.

The model we consider has been fully derived and studied in [5] as a generalization of the two-phase Cahn-Hilliard model. In the diphasic case, the free energy of the mixture depends on two parameters: the interface width ε and the surface tension σ. It can be written as follows:

$$\mathcal{F}_{\sigma,\varepsilon}^{\mathrm{diph}}(c) = \int_\Omega 12\frac{\sigma}{\varepsilon}c^2(1-c)^2 + \frac{3}{4}\sigma\varepsilon\,|\nabla c|^2\,dx.$$

Therefore, in [5], the authors have postulated that the three-phase free energy can be written as follows:

$$\mathcal{F}_{\boldsymbol{\Sigma},\varepsilon}^{\mathrm{triph}}(c_1,c_2,c_3) = \int_\Omega \frac{12}{\varepsilon}F(c_1,c_2,c_3) + \frac{3}{8}\varepsilon\Sigma_1|\nabla c_1|^2 + \frac{3}{8}\varepsilon\Sigma_2|\nabla c_2|^2 + \frac{3}{8}\varepsilon\Sigma_3|\nabla c_3|^2\,dx. \qquad (15)$$

The triple of constant parameters $\boldsymbol{\Sigma} = (\Sigma_1, \Sigma_2, \Sigma_3)$ and the bulk energy F have been determined so that the model fits with the prescribed surface tension σ_{12}, σ_{13} and σ_{23} and is "consistent" with the two-component situation (see below).

Refinement level		1	2	3	4
Square-$\mathbb{Q}_2$	Number of unknowns	3653	12405	44485	166253
	No preconditioner	202	404	–	–
	ILU0	79	135	259	508
	P_a	36	56	69	77
	P_m	16	18	20	20
Quadrangle-$\mathbb{Q}_2$	Number of unknowns	3827	13129	46881	175126
	No preconditioner	595	–	–	–
	ILU0	112	195	389	–
	P_a	83	127	158	178
	P_m	27	31	33	34
Tri/Quadr-angle-$\mathbb{P}_2/\mathbb{Q}_2$	Number of unknowns	3569	11441	40293	151013
	No preconditioner	180	361	–	–
	ILU0	80	152	309	–
	P_a	38	54	73	82
	P_m	17	18	20	21

TABLE 4. Two dimensional test cases. Number of iterations in the conjugate gradient solver as a function of the number of unknowns.

Refinement level		1	2	3
Cube-$\mathbb{Q}_1$	Number of unknowns	9942	63329	459063
	No preconditioner	45	224	–
	ILU0	28	55	109
	P_a	21	31	35
	P_m	10	13	14

TABLE 5. Three dimensional test case. Number of iterations in the conjugate gradient solver as a function of the number of unknowns.

The evolution of the system is then driven by the gradient of the total free energy $\mathcal{F}_{\mathbf{\Sigma},\varepsilon}^{\mathrm{triph}}$ and the time evolution of $\mathbf{c} = (c_1, c_2, c_3)$ is governed by the following system of equations:

$$
\begin{cases}
\dfrac{\partial c_i}{\partial t} = \nabla \cdot \left(\dfrac{M_0}{\Sigma_i} \nabla \mu_i \right), & \text{for } i = 1, 2, 3 \\
\mu_i = f_i^F(\mathbf{c}) - \dfrac{3}{4}\varepsilon\Sigma_i\Delta c_i , & \text{for } i = 1, 2, 3
\end{cases}
\tag{16}
$$

where M_0 is a constant (it could also depend on $\mathbf{c}$) and

$$
f_i^F(\mathbf{c}) = \frac{4\Sigma_T}{\varepsilon} \sum_{j \neq i} \left(\frac{1}{\Sigma_j} \left(\partial_i F(\mathbf{c}) - \partial_j F(\mathbf{c}) \right) \right) \text{ with } \Sigma_T \text{ defined by } \frac{3}{\Sigma_T} = \frac{1}{\Sigma_1} + \frac{1}{\Sigma_2} + \frac{1}{\Sigma_3}.
$$

This choice of f_i^F, obtained by the use of a Lagrange multipliers technique, enforces the condition (14) all along the time. Thus, one of the unknowns can be arbitrarily eliminated from the system (16) which is then equivalent

to:

$$
\begin{cases}
\dfrac{\partial c_i}{\partial t} = \nabla \cdot \left(\dfrac{M_0}{\Sigma_i} \nabla \mu_i \right), & \text{for } i = 1, 2 \\[2mm]
c_3 = 1 - c_1 - c_2 \\[2mm]
\mu_i = f_i^F(\mathbf{c}) - \dfrac{3}{4} \varepsilon \Sigma_i \Delta c_i, & \text{for } i = 1, 2 \\[2mm]
\mu_3 = - \left(\dfrac{\Sigma_3}{\Sigma_1} \mu_1 + \dfrac{\Sigma_3}{\Sigma_2} \mu_2 \right).
\end{cases}
\tag{17}
$$

For simplicity, we will later omit to write the equations giving c_3 and μ_3 as functions of c_1, c_2 and μ_1, μ_2 respectively. The consistency (or algebrical consistency) of the model defined by (15) and (16) with the diphasic systems corresponding to the given surface tensions $\sigma_{12}, \sigma_{13}, \sigma_{23}$ respectively means that the following properties hold:

- When the component i is not present, that is $c_i \equiv 0$, the total free energy $\mathcal{F}_{\Sigma,\varepsilon}^{\mathrm{triph}}(c_1, c_2, c_3)$ of the system has to be exactly equal to the total free energy $\mathcal{F}_{\sigma_{jk},\varepsilon}^{\mathrm{diph}}(c_j)$ of the diphasic system corresponding to the two other phases.
- When the component i is not present in the mixture at the initial time, the component i must not appear during the time evolution of the system.

Finally, it is shown in [5] that the model defined by (15) and (16) is *algebraically consistent* with the diphasic systems of surface tensions $\sigma_{12}, \sigma_{13}, \sigma_{23}$ respectively if and only if we have

$$
\forall i \in \{1, 2, 3\}, \quad \Sigma_i = \sigma_{ij} + \sigma_{ik} - \sigma_{jk},
\tag{18}
$$

and there exists a smooth function G such that

$$
\forall \mathbf{c} \in \mathbb{R}^3 \text{ such that } c_1 + c_2 + c_3 = 1, \quad F(\mathbf{c}) = \sigma_{12} c_1^2 c_2^2 + \sigma_{13} c_1^2 c_3^2 + \sigma_{23} c_2^2 c_3^2 + c_1 c_2 c_3 (\Sigma_1 c_1 + \Sigma_2 c_2 + \Sigma_3 c_3) \\
+ c_1^2 c_2^2 c_3^2 \, G(\mathbf{c}).
$$

In the physical literature, the coefficient $S_i = -\Sigma_i$ defined by (18) is well known and called *the spreading coefficient* of the phase i at the interface between phases j and k. If S_i is positive (that is $\Sigma_i < 0$), the spreading is said to be *total* and if S_i is negative, it is said to be *partial*.

Notice that, in the following study, the coefficients Σ_i are not assumed to be positive, so that the model presented above lets us cope with some total spreading situations. However, in order to have a well-posed system, we assume that the following condition holds:

$$
\Sigma_1 \Sigma_2 + \Sigma_1 \Sigma_3 + \Sigma_2 \Sigma_3 > 0,
\tag{19}
$$

In this paper we will consider Cahn-Hilliard potentials with the following form

$$
F(\mathbf{c}) = \underbrace{\sigma_{12} c_1^2 c_2^2 + \sigma_{13} c_1^2 c_3^2 + \sigma_{23} c_2^2 c_3^2 + c_1 c_2 c_3 (\Sigma_1 c_1 + \Sigma_2 c_2 + \Sigma_3 c_3)}_{F_0(\mathbf{c})} + \underbrace{3 \Lambda c_1^2 c_2^2 c_3^2}_{P(\mathbf{c})}
$$

Under these assumptions, there exists a unique weak solution for the problem (16) (in 2D and with a slightly modified potential in 3D).

3.2. **Discretization procedure**

In this section, we present the time and space discretizations of problem (17) that we used. We also state two results ensuring existence of the discrete solution and convergence of this solution towards the weak solution of the continuous problem.

3.2.1. *Time discretization*

Let $N \in \mathbb{N}^*$. The temporal interval $[0, t_f]$ is uniformly discretized with a fixed time step $\Delta t = \dfrac{t_f}{N}$. For $n \in [\![0, N]\!]$, we define $t_n = n\Delta t$. We use a semi-implicit discretization in time with a special care for nonlinear terms. The scheme is written as follows,

$$
\begin{cases}
\dfrac{c_i^{n+1} - c_i^n}{\Delta t} = \nabla \cdot \left(\dfrac{M_0}{\Sigma_i} \nabla \mu_i^{n+1} \right), \text{ for } i = 1, 2, \\[3mm]
\mu_i^{n+1} = D_i^F(\mathbf{c}^{n+1}, \mathbf{c}^n) - \dfrac{3}{4}\varepsilon\Sigma_i \left[\alpha\Delta c_i^n + (1 - \alpha)\Delta c_i^{n+1} \right], \text{ for } i = 1, 2,
\end{cases}
$$

where $0 \leqslant \alpha < \dfrac{1}{2}$ and $D_i^F(\mathbf{c}^{n+1}, \mathbf{c}^n) = \dfrac{4\Sigma_T}{\varepsilon} \sum\limits_{j \neq i} \left(\dfrac{1}{\Sigma_j} \left(d_i^F(\mathbf{c}^{n+1}, \mathbf{c}^n) - d_j^F(\mathbf{c}^{n+1}, \mathbf{c}^n) \right) \right).$

The functions d_i^F represent a semi-implicit discretization of $\partial_{c_i}F$. Since $F = F_0 + P$, we choose a discretization of the form $d_i^F = d_i^{F_0} + d_i^P$ where $d_i^{F_0}$ and d_i^P represent the discretization of $\partial_{c_i}F_0$ and $\partial_{c_i}P$ respectively.

We choose a fully implicit discretization of $\partial_{c_i}F_0$. It corresponds to the following definition of $\mathbf{d}^{F_0}$: for $i \in \{1, 2, 3\}$,

$$
d_i^{F_0}(\mathbf{c}^{n+1}, \mathbf{c}^n) = \partial_i F_0(\mathbf{c}^{n+1}). \tag{20}
$$

For $\partial_{c_i}P$, we choose a semi-implicit discretization: for $i \in \{1, 2, 3\}$,

$$
d_i^P(\mathbf{c}^{n+1}, \mathbf{c}^n) = 2\Lambda c_i^{n+1} \left[(c_j^n)^2(c_k^n)^2 + \dfrac{1}{2}(c_j^{n+1})^2(c_k^n)^2 + \dfrac{1}{2}(c_j^n)^2(c_k^{n+1})^2 + (c_j^{n+1})^2(c_k^{n+1})^2 \right]. \tag{21}
$$

We add the following Neumann boundary conditions for each of the unknown which ensures, in particular, that the volume of each phase is conserved along the time evolution:

$$
\nabla c_i^{n+1} \cdot n = M_0 \nabla \mu_i^{n+1} \cdot n = 0, \quad \text{ on } \partial\Omega.
$$

Other boundary conditions of interest will be presented in [7].

3.2.2. *Space discretization of problem* (17)

We first give the variational formulation of the semi-discrete problem.

Problem 3.1 (Variational formulation). Given $(c_1^n, c_2^n) \in (\mathrm{H}^1(\Omega))^2$, find $(c_1^{n+1}, c_2^{n+1}, \mu_1^{n+1}, \mu_2^{n+1}) \in (\mathrm{H}^1(\Omega))^4$ so that $\forall \nu \in \mathrm{H}^1(\Omega)$, we have, for $i = 1$ and 2,

$$
\begin{cases}
\displaystyle\int_\Omega \dfrac{c_i^{n+1} - c_i^n}{\Delta t}\nu \, dx = -\int_\Omega \dfrac{M_0}{\Sigma_i}\nabla\mu_i^{n+1} \cdot \nabla\nu \, dx, \\[4mm]
\displaystyle\int_\Omega \mu_i^{n+1}\nu \, dx = \int_\Omega D_i^F(\mathbf{c}^{n+1}, \mathbf{c}^n)\nu \, dx + \int_\Omega \dfrac{3}{4}\Sigma_i\varepsilon \left[\alpha\nabla c_i^n + (1 - \alpha)\nabla c_i^{n+1} \right] \cdot \nabla\nu \, dx.
\end{cases}
$$

where $\mathbf{c}^n = (c_1^n, c_2^n, 1 - c_1^n - c_2^n)$.

For the Galerkin approximation in space we introduce $\mathcal{V}_h$ a finite element (possibly multilevel) approximation subspace of $\mathrm{H}^1(\Omega)$.

Problem 3.2 (Galerkin formulation). Given $(c_{1h}^n, c_{2h}^n) \in (\mathcal{V}_h)^2$, find $(c_{1h}^{n+1}, c_{2h}^{n+1}, \mu_{1h}^{n+1}, \mu_{2h}^{n+1}) \in (\mathcal{V}_h)^4$ such that $\forall \nu_h \in \mathcal{V}_h$, we have, for $i = 1, 2$,

$$
\begin{cases}
\displaystyle\int_\Omega \dfrac{c_{ih}^{n+1} - c_{ih}^n}{\Delta t}\nu_h \, dx = -\int_\Omega \dfrac{M_0}{\Sigma_i}\nabla\mu_{ih}^{n+1} \cdot \nabla\nu_h \, dx, \\[4mm]
\displaystyle\int_\Omega \mu_{ih}^{n+1}\nu_h \, dx = \int_\Omega D_h^F(\mathbf{c}_h^{n+1}, \mathbf{c}_h^n)\nu_h \, dx + \int_\Omega \dfrac{3}{4}\Sigma_i\varepsilon \left[\alpha\nabla c_{ih}^n + (1 - \alpha)\nabla c_{ih}^{n+1} \right] \cdot \nabla\nu_h \, dx.
\end{cases}
$$

where $\mathbf{c}_h^n = (c_{1h}^n, c_{2h}^n, 1 - c_{1h}^n - c_{2h}^n)$.

Remark 3.3. Note that we use the same approximation space $\mathcal{V}_h$ for each of the unknowns $c_{1h}^{n+1}, c_{2h}^{n+1}, \mu_{1h}^{n+1}$ and μ_{2h}^{n+1}. Moreover, remark that in the formulation of Problem 3.2 and in Theorems 3.4 and 3.5, we assume that the approximation space $\mathcal{V}_h$ does not depend of n. Because of adaptivity, this is not the case in practice (see Section 3.3.1).

3.2.3. *Existence - Convergence*

Theorem 3.4. *Let* $(c_{1h}^n, c_{2h}^n) \in (\mathcal{V}_h)^2$. *We assume that, for all* $i \in \{1, 2, 3\}$, $\Sigma_i > 0$. *There exists a solution* $(c_{1h}^{n+1}, c_{2h}^{n+1}, \mu_{1h}^{n+1}, \mu_{2h}^{n+1}) \in (\mathcal{V}_h)^4$ *for Problem 3.2.*

We define

$$
\begin{cases}
c_{ih}^N(t, \cdot) = \dfrac{t_{n+1} - t}{\Delta t} c_{ih}^n(\cdot) + \dfrac{t - t_n}{\Delta t} c_{ih}^{n+1}(\cdot) & \text{for } t \in [t_n, t_{n+1}], \\[2mm]
\mu_{ih}^N(t, \cdot) = \mu_{ih}^{n+1}(\cdot) & \text{for } t \in [t_n, t_{n+1}].
\end{cases}
$$

The following convergence result holds.

Theorem 3.5. *Let* (c_1, c_2, μ_1, μ_2) *be the weak solution of the continuous problem* (17). *We assume that, for all* $i \in \{1, 2, 3\}$, $\Sigma_i > 0$ *and that the finite element approximation spaces are such that*

- $\displaystyle \inf_{\nu_h \in \mathcal{V}_h} |\nu - \nu_h|_{\mathrm{H}^1(\Omega)} \underset{h \to 0}{\to} 0$, *for any* $\nu \in \mathrm{H}^1(\Omega)$.

- *There exists a constant* C *such that* $\forall \nu \in \mathrm{H}^1(\Omega)$, $\left| \Pi_0^{\mathcal{V}_h} \nu \right|_{\mathrm{H}^1(\Omega)} \leqslant C |\nu|_{\mathrm{H}^1(\Omega)}$ *where* $\Pi_0^{\mathcal{V}_h}$ *is the* $\mathrm{L}^2(\Omega)$- *projection on* $\mathcal{V}_h$.

Then, for $i = 1$ *and* 2,

$$
\begin{aligned}
c_{ih}^N &\to c_i \ \text{in } \mathrm{C}^0([0, T], \mathrm{L}^2(\Omega)) \ \text{strong}, \\
c_{ih}^N &\rightharpoonup c_i \ \text{in } \mathrm{L}^\infty([0, T], \mathrm{H}^1(\Omega)) \, \text{weak-*}, \\
\mu_{ih}^N &\rightharpoonup \mu_i \ \text{in } \mathrm{L}^2_{loc}([0, T], \mathrm{H}^1(\Omega)) \ \text{weak}.
\end{aligned}
$$

when $(\Delta t, h) \to (0, 0)$.

These two theorems are not proved here; complete proofs will be presented in [7]. The conerstone of these proofs is a suitable discrete energy estimate. In the case where all Σ_i are not assumed to be positive, this estimate is not satisfied for the implicit time discretization (20) of F_0. Thus, other semi-implicit discretizations of this term (of the same kind that the given discretization (21) of P) are introduced in [7] to ensure the validity of a such estimate and then to prove the convergence theorem in the case where we only assume that the condition (19) is fulfilled.

3.3. **Practical issues**

3.3.1. *Newton linearization method*

In practice, for the numerical resolution of the problem we choose for $\mathcal{V}_h$ a multilevel approximation space as described in Section 1.3.2. Because of adaptivity, the unknowns are discretized in two different multilevel spaces at time step n and at time step $n + 1$, say $\mathcal{V}_h^n = \mathrm{span}\, \mathcal{B}^n$ and $\mathcal{V}_h^{n+1} = \mathrm{span}\, \mathcal{B}^{n+1}$. Hence, the problem we have to solve is in fact:

Given $(c_{1h}^n, c_{2h}^n) \in (\mathcal{V}_h^n)^2$, find $(c_{1h}^{n+1}, c_{2h}^{n+1}, \mu_{1h}^{n+1}, \mu_{2h}^{n+1}) \in (\mathcal{V}_h^{n+1})^4$ such that $\forall \nu_h \in \mathcal{V}_h^{n+1}$, we have, for $i = 1, 2$,

$$
\begin{cases}
\displaystyle \int_\Omega \dfrac{c_{ih}^{n+1} - c_{ih}^n}{\Delta t} \nu_h \, dx = - \int_\Omega \dfrac{M_0}{\Sigma_i} \nabla \mu_{ih}^{n+1} \cdot \nabla \nu_h \, dx, \\[3mm]
\displaystyle \int_\Omega \mu_{ih}^{n+1} \nu_h \, dx = \int_\Omega D_i^F(\mathbf{c}_h^{n+1}, \mathbf{c}_h^n) \nu_h \, dx + \int_\Omega \dfrac{3}{4} \Sigma_i \varepsilon \left[\alpha \nabla c_{ih}^n + (1 - \alpha) \nabla c_{ih}^{n+1} \right] \cdot \nabla \nu_h \, dx.
\end{cases}
$$

This problem is solved by the Newton-Raphson linearization method. The algorithm is initialized by a L^2-projection of $(c_{1h}^n, c_{2h}^n, \mu_{1h}^n, \mu_{2h}^n)$ on $\mathcal{V}_h^{n+1}$ and at each time step we have to solve the following linear system:

$$\begin{pmatrix} \mathbf{J}^{c_1 c_1} & \mathbf{J}^{c_1 \mu_1} & \mathbf{J}^{c_1 c_2} & 0 \\ \mathbf{J}^{\mu_1 c_1} & \mathbf{J}^{\mu_1 \mu_1} & 0 & 0 \\ \mathbf{J}^{c_2 c_1} & 0 & \mathbf{J}^{c_2 c_2} & \mathbf{J}^{c_2 \mu_2} \\ 0 & 0 & \mathbf{J}^{\mu_2 c_2} & \mathbf{J}^{\mu_2 \mu_2} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{1h}^{(k+1)} - \mathbf{c}_{1h}^{(k)} \\ \boldsymbol{\mu}_{1h}^{(k+1)} - \boldsymbol{\mu}_{1h}^{(k)} \\ \mathbf{c}_{2h}^{(k+1)} - \mathbf{c}_{2h}^{(k)} \\ \boldsymbol{\mu}_{2h}^{(k+1)} - \boldsymbol{\mu}_{2h}^{(k)} \end{pmatrix} = \begin{pmatrix} -\mathcal{R}^{c_1} \\ -\mathcal{R}^{\mu_1} \\ -\mathcal{R}^{c_2} \\ -\mathcal{R}^{\mu_2} \end{pmatrix}, \tag{22}$$

where $(\mathbf{c}_{1h}^{(k)}, \mathbf{c}_{2h}^{(k)}, \boldsymbol{\mu}_{1h}^{(k)}, \boldsymbol{\mu}_{2h}^{(k)})$ is the k-th iterate in the Newton iterative method and the matrix blocks are defined as follows: $\forall (I, J) \in [\![1, N_{\text{dof}}^{n+1}]\!]^2$,

$$[\mathbf{J}^{c_i c_i}]_{IJ} = \int_\Omega -\frac{3}{4}\Sigma_i \varepsilon(1-\alpha)\nabla\nu_J \cdot \nabla\nu_I \, dx - \int_\Omega \frac{\partial D_i}{\partial c_i}(\mathbf{c}_h^{(k)}, \mathbf{c}_h^n)\nu_J\nu_I \, dx,$$

$$[\mathbf{J}^{c_i c_j}]_{IJ} = \int_\Omega -\frac{\partial D_i}{\partial c_j}(\mathbf{c}_h^{(k)}, \mathbf{c}_h^n)\nu_J\nu_I \, dx,$$

$$[\mathbf{J}^{c_i \mu_i}]_{IJ} = \int_\Omega \nu_J\nu_I \, dx,$$

$$[\mathbf{J}^{\mu_i c_i}]_{IJ} = \int_\Omega \frac{1}{\Delta t}\nu_J\nu_I \, dx, \tag{23}$$

$$[\mathbf{J}^{\mu_i \mu_i}]_{IJ} = \int_\Omega \frac{M_0}{\Sigma_i}\nabla\nu_J \cdot \nabla\nu_I \, dx,$$

$$[\mathcal{R}^{c_i}]_I = \int_\Omega \mu_{ih}^{(k)}\nu_I \, dx - \int_\Omega D_i(\mathbf{c}_h^{(k)}, \mathbf{c}_h^n)\nu_I \, dx - \int_\Omega \frac{3}{4}\Sigma_i \varepsilon\left[\alpha\nabla c_{ih}^n + (1-\alpha)\nabla c_{ih}^{(k)}\right] \cdot \nabla\nu_I \, dx,$$

$$[\mathcal{R}^{\mu_i}]_I = \int_\Omega \frac{c_{ih}^{(k)} - c_{ih}^n}{\Delta t}\nu_I \, dx + \int_\Omega \frac{M_0}{\Sigma_i}\nabla\mu_{ih}^{(k)} \cdot \nabla\nu_I \, dx.$$

This linear system (22) is solved by the preconditionned GMRES iterative method. We use a diagonal bloc preconditioner:

$$\begin{pmatrix} \mathrm{P}^{c_1 c_1} & 0 & 0 & 0 \\ 0 & \mathrm{P}^{\mu_1 \mu_1} & 0 & 0 \\ 0 & 0 & \mathrm{P}^{c_2 c_2} & 0 \\ 0 & 0 & 0 & \mathrm{P}^{\mu_2 \mu_2} \end{pmatrix}$$

where $\mathrm{P}^{c_1 c_1}, \mathrm{P}^{\mu_1 \mu_1}, \mathrm{P}^{c_2 c_2}$ and $\mathrm{P}^{\mu_2 \mu_2}$ are preconditioners built as explain in Section 2.2 (either additive or multiplicative version) associated respectively to the matrices $A_J = \mathbf{J}^{c_1 c_1}, \mathbf{J}^{\mu_1 \mu_1}, \mathbf{J}^{c_2 c_2}$ and $\mathbf{J}^{\mu_2 \mu_2}$.

3.3.2. *Assembly avoiding field tranfers*

Among the integrals in (23), some involve discrete unknowns which belong to two distinct approximation spaces; for example the integral

$$\int_\Omega c_{ih}^n \nu_I \, dx$$

from $[\mathcal{R}^{\mu_i}]_I$ involves $c_{ih}^n \in \mathcal{V}_h^n$ and $\nu_I \in \mathcal{V}_h^{n+1}$. Such an integral can be exactly computed avoiding field tranfers thanks to the following notions. Note that until the end of this section, the time step $n + 1$ is fixed.

Definition 3.6 (Active DOF). We say that $k \in \{1, \ldots, N_{\text{dof}}^{[j]}\}$ is an active degree of freedom of level-$[j]$, if and only if $\varphi_k^{[j]} \in \mathcal{B}^n \cup \mathcal{B}^{n+1}$. We denote by $\mathcal{A}_{\text{dof}}^{[j]}$ the set of active degrees of freedom of level-$[j]$.

Definition 3.7. Let $j \in [\![0, J]\!]$. We say that a level-$[j]$ cell $K^{[j]}$ satisfies the $(\mathrm{A}_{K^{[j]}})$ property iff

$$\forall j' \in [\![j+1, J]\!], \ \forall k \in \mathcal{A}_{\mathrm{dof}}^{[j']}, \ \overset{\circ}{\overbrace{K^{[j]}}} \cap \overbrace{\mathrm{supp}[\varphi_k^{[j']}]} = \emptyset. \qquad (\mathrm{A}_{K^{[j]}})$$

Remark 3.8. This definition ensures that on an active cell, all basis functions involved in the discretization of the problem, has a "unique" expression (opposed to an elementwise definition). Thus, in the case of polynomial finite elements (e.g. $\mathbb{P}_k, \mathbb{Q}_k$), suitable quadrature rules can be used in order to exactly compute all the integrals.

Definition 3.9 (Active cells). A level-$[j]$ cell $K^{[j]}$ of $\mathcal{T}_j$ is called active iff:
- $K^{[j]}$ satisfies the property $(\mathrm{A}_{K^{[j]}})$, and
- in the case of $j > 0$, its parent cell $P(K^{[j]})$ (of level $[j-1]$) does not satisfy the property $\mathrm{A}_{P(K^{[j]})}$.

With this definition at hand, we can introduce the notion of a multilevel mesh.

Proposition 3.10 (Multilevel mesh). *Let $\widetilde{\mathcal{T}}_j$ be the set of active cells of $\mathcal{T}_j$. The set $\mathcal{T} = \bigcup_{j=0}^{J} \widetilde{\mathcal{T}}_j$ is a mesh of Ω called multilevel mesh.*

Proof. Let $K_e^{[j]}$ and $K_{e'}^{[j']}$ be two distinct active cells and let us show that $\overset{\circ}{\overbrace{K_e^{[j]}}} \cap \overset{\circ}{\overbrace{K_{e'}^{[j']}}} = \emptyset$.
- Case 1 : $j = j'$. Necessarily, we have $e \neq e'$.

  In this case, $K_e^{[j]}$ and $K_{e'}^{[j']}$ are two distinct cells of the mesh $\mathcal{T}_j$. Then, $\overset{\circ}{\overbrace{K_e^{[j]}}} \cap \overset{\circ}{\overbrace{K_{e'}^{[j']}}} = \emptyset$.
- Case 2 : $j > j'$.

  Arguing by contradiction, assume that $\overset{\circ}{\overbrace{K_e^{[j]}}} \cap \overset{\circ}{\overbrace{K_{e'}^{[j']}}} \neq \emptyset$. Since $j > j'$, we have $K_e^{[j]} \subset K_{e'}^{[j']}$. But, the cell $K_e^{[j]}$ is active and $\neg \left(\mathrm{A}_{P(K_e^{[j]})}\right)$ yields that there exits $j_0 \geqslant j$ and $k_0 \in \mathcal{A}_{\mathrm{dof}}^{[j_0]}$ such that $\overset{\circ}{\overbrace{P(K_e^{[j]})}} \cap \overbrace{\mathrm{supp}[\varphi_{k_0}^{[j_0]}]} \neq \emptyset$.

  Moreover, we have $j_0 > j'$ and $\left(\mathrm{A}_{K_{e'}^{[j']}}\right)$ yields $\overset{\circ}{\overbrace{K_{e'}^{[j']}}} \cap \overbrace{\mathrm{supp}[\varphi_{k_0}^{[j_0]}]} = \emptyset$. However, we have

$$(j > j' \text{ and } K_e^{[j]} \subset K_{e'}^{[j']}) \Longrightarrow \overset{\circ}{\overbrace{P(K_e^{[j]})}} \subset \overset{\circ}{\overbrace{K_{e'}^{[j']}}}.$$

  Hence, $\emptyset \neq \left(\overset{\circ}{\overbrace{P(K_e^{[j]})}} \cap \overbrace{\mathrm{supp}[\varphi_{k_0}^{[j_0]}]}\right) \subset \left(\overset{\circ}{\overbrace{K_{e'}^{[j']}}} \cap \overbrace{\mathrm{supp}[\varphi_{k_0}^{[j_0]}]}\right) = \emptyset$. This is a contradiction.

Hence, the interiors of two different active cells are disjoint.

Let $x \in \overline{\Omega}$. Since $\mathcal{T}_J$ is a mesh of Ω, there exists a level-$[J]$ cell $K_{e_J}^{[J]}$ which contains x. Then, for all $j \in [\![0, J-1]\!]$, we define $K_{e_j}^{[j]} = P(K_{e_{j+1}}^{[j+1]})$. Hence, for all $j \in [\![0, J]\!]$, x belongs to the cell $K_{e_j}^{[j]}$. Consider the set $E = \left\{ j \in [\![0, J]\!], \ \forall \ell \geqslant j, \ K_{e_\ell}^{[\ell]} \text{ satisfies } (\mathrm{A}_{K_{e_\ell}^{[\ell]}}) \right\}$. We have $J \in E$, so $E \neq \emptyset$. Let $j_m = \min_{j \in E} j$, then, by definition, $K_{e_{j_m}}^{[j_m]}$ is active and contains x. $\qquad\square$

Remark 3.11. Multilevel meshes are not geometrically conformal, but multilevel finite element spaces are $\mathrm{H}^1(\Omega)$-conformal by construction since span $\mathcal{B} \subset X_J \subset \mathrm{H}^1(\Omega)$. The non-conformity of multilevel meshes is not a problem since they are only used as elementary integration domains.

Since $\mathcal{T}$ is a partition of Ω, the integral can be decomposed into a sum over all cells of the multilevel mesh $\mathcal{T}$ on which a quadrature rule is then applied leading to exact computations (see Remark 3.8).

3.3.3. *Refinement criterion*

For the Cahn-Hilliard model, we want to refine in the interface zone (that is to say where the order parameters have important variations). For this reason, we need to introduce an indicator measuring whether or not a given cell (resp. basis function) lies in the interface at a given time step.

Definition 3.12 (Per-cell indicator). At a given time step n, for any active cell K, we define the following indicator:

$$\eta_K = \max\Big(\ \frac{1}{|K|} \int_K c_1^n \ , \ \ \frac{1}{|K|} \int_K c_2^n \ , \ \ \frac{1}{|K|} \int_K c_3^n \ \Big)$$

Remark 3.13. This per-cell indicator can be interpreted as follows:
- $\eta_K = 1$ means that the cell K is completely filled with one of the bulk phases.
- $\eta_K < 1$ means that the cell K contains an interface.

We can deduce a criterion for deciding whether or not a given basis function may be (un-)refined.

Definition 3.14 (Basis function indicator). At a given time step n, for a basis function φ, we define the following indicator:

$$\eta_\varphi = \frac{1}{|\text{supp}[\varphi]|} \sum_{\substack{K \in \mathcal{T}, \\ K \cap \text{supp}[\varphi] \neq \emptyset}} |K| \eta_K$$

Criterion 3.15 ((Un)Refinement criterion). *Given an anticipated cell size $h_{interface}$ for the interface neighborhood, the two following criterion let us decide if a basis function φ has to be refined or unrefined.*

- *Refinement criterion:*

$$\eta_\varphi < 0.90 \quad and \quad diam(K) > h_{interface} \quad for\ at\ least\ one\ cell\ \ K \subset supp[\varphi].$$

- *Unrefinement criterion:*

$$\eta_\varphi > 0.95.$$

3.4. Numerical experiments

Finally, we give an application of the local refinement procedure and the multigrid methods to the resolution of the above ternary Cahn-Hilliard model in the case of a liquid lens spreading between two stratified fluids. Figure 21 shows the initial position of the interface and the numbering of phases used to defined the surface tensions σ_{ij} between the phases i and j.
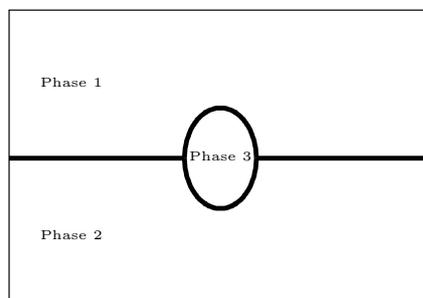

FIGURE 21. Initial position of interface. Numbering of phases.

Ω	ε	σ_{12}	σ_{13}	σ_{23}	M_0	Δt	t_f
$[0, 0.1] \times [0, 0.1]$	$2.8\,10^{-3}$	1	0.6	0.6	10^{-4}	10^{-4}	$2\,10^{-3}$

TABLE 6. Parameters.

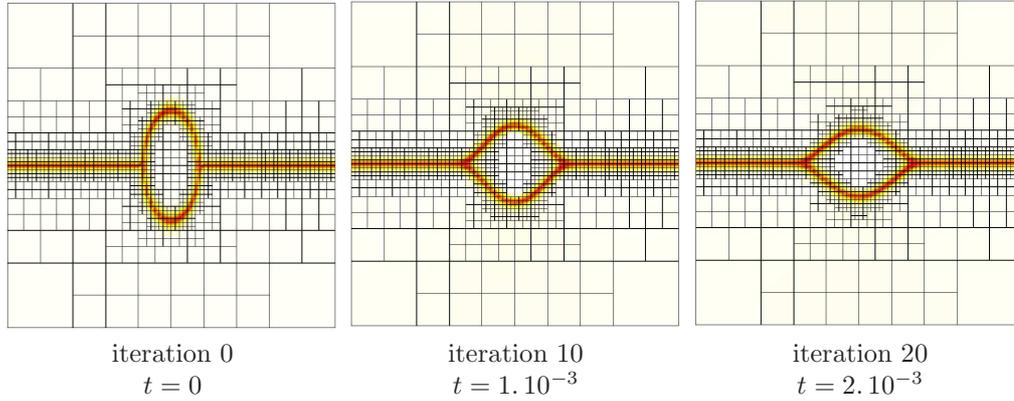| iteration 0 | iteration 10 | iteration 20 |
| $t = 0$ | $t = 1.\,10^{-3}$ | $t = 2.\,10^{-3}$ |

FIGURE 22. Square-$\mathbb{Q}_1$. Time marching. $h_{\text{interface}} = \varepsilon$

Table 6 gives the parameters we used for numerical simulations. The refinement criterion is given in Section 3.3.3; $h_{\text{interface}}$ is specified for each mesh.

Figures 22, 23, 24 shows three steps of the time marching. In each case, the value of the parameter $h_{\text{interface}}$ is voluntarily chosen large enough for the ease of visualisation.

As in Section 2.3.3, we represent in these figure the refined mesh and the function $(1 - c_1)(1 - c_2)(c_1 + c_2)$. This let us show the position of the interfaces.
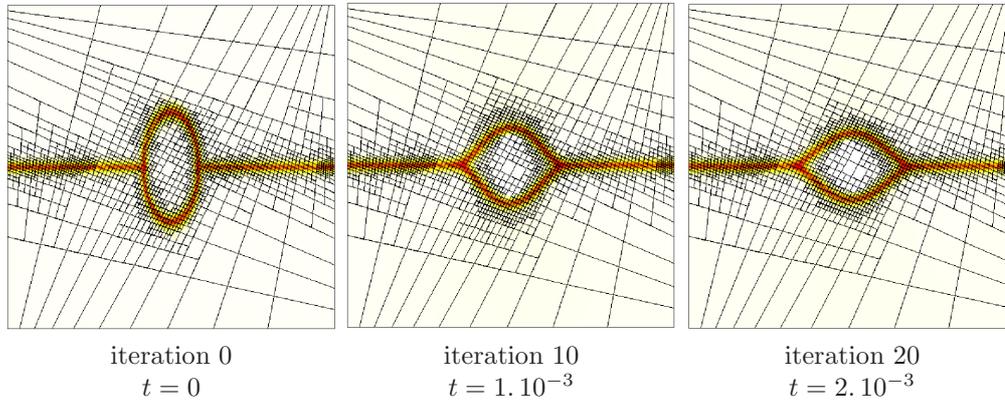

| iteration 0 | iteration 10 | iteration 20 |
| $t = 0$ | $t = 1.\,10^{-3}$ | $t = 2.\,10^{-3}$ |

FIGURE 23. Quadrangle-$\mathbb{Q}_1$. Time marching. $h_{\text{interface}} = 0.8\varepsilon$

Table 7 illustrates that the multigrid process can be used to develop convergent iterative algorithms for the solution of the Cahn-Hilliard equations on local refined meshes. The simulations have been done during twenty time steps. For each time step, we calculate an average of the number of iterations required to solve the linear system by the GMRES method, over all iterations of the Newton algorithm. In Table 7, the numbers which face the preconditioner name are the minimum, the maximum and the average of this numbers over all the twenty time steps. For these simulations, we used $h_{\text{interface}} = 0.2\varepsilon$. Note that in these simulations, we say that

iteration 0   iteration 10   iteration 20
$t = 0$ $t = 1. \, 10^{-3}$ $t = 2. \, 10^{-3}$
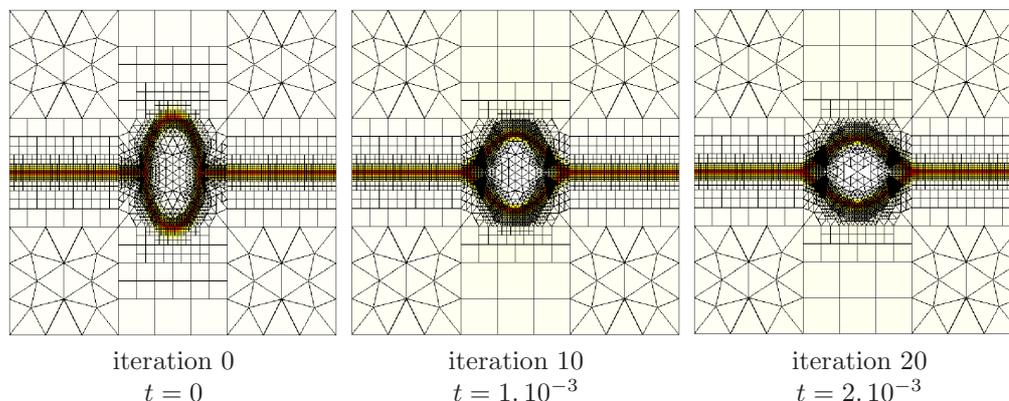
FIGURE 24. Tri/Quadr-angle-$\mathbb{P}_1/\mathbb{Q}_1$. Time marching. $h_{\text{interface}} = 0.4\varepsilon$

convergence in the GMRES method is achieved as soon as the relative L^2-norm of the residual (and not the preconditionned residual) is less than 10^{-10}. In particular, this criterion is independent of the preconditioners.

We observe that the number of necessary iterations is the almost the same from a time step to another and that the number of necessary iterations using (P_m) and (P_a) is significantly smaller than using (ILU0) preconditioner.

			Min	Max	Average
Square-$\mathbb{Q}_1$	Number of unknowns		29020	30844	30032
	Number of iterations	ILU0	188	232	213
		P_a	68	77	72
		P_m	18	19	18
Quadrangle-$\mathbb{Q}_1$	Number of unknowns		45000	47900	46428
	Number of iterations	ILU0	291	346	308
		P_a	72	80	74
		P_m	18	26	18
Tri/Quadr-angle-$\mathbb{P}_1/\mathbb{Q}_1$	Number of unknowns		28780	31204	30266
	Number of iterations	ILU0	265	289	278
		P_a	57	71	63
		P_m	16	20	17

TABLE 7. Number of iterations in the GMRES solver. $h_{\text{interface}} = 0.2\varepsilon$

4. CONCLUSION

This paper is devoted to the description and the resolution of some numerical issues linked to local adaptive refinement methods. We show how a hierarchy of nested conforming meshes can be built recursively applying a same refinement pattern to each cell of an initial mesh possibly unstructured. This hierarchy is then used in a slightly modified version of the CHARMS method [15] in order to perform local refinement. Furthermore, the multilevel structure of the approximation spaces built with this method is exploited to derive suitable multigrid preconditioners for solving the corresponding linear system. We illustrate possibilities of the whole method with simultations of a lens spreading between two stratified liquid phases thanks to a ternary Cahn-Hilliard model. We particularly point out the practical way to implement the method for a time dependent problem. More complex situation have been considered [6] with simulations of three phases flows by solving the coupling of

the Cahn-Hilliard system and the Navier-Stokes system. As an example, we give in Figure 25 the simulation of the rise of a gaz bubble across a light/heavy liquids interface. In future works, we will incorporate multigrid preconditioners in finite element penalty projection method for solving Navier-Stokes system [11,14] in order to build efficient solver for the complete model.
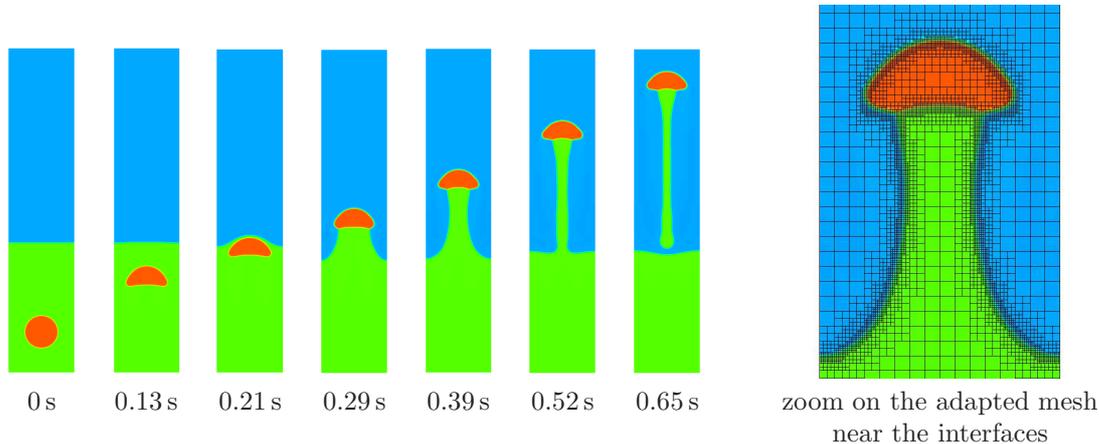

| 0 s | 0.13 s | 0.21 s | 0.29 s | 0.39 s | 0.52 s | 0.65 s | zoom on the adapted mesh near the interfaces |

FIGURE 25. Entrainment of an heavy liquid during a bubble rising.

REFERENCES

[1] R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numerische Mathematik*, 52:427–458, 1988.

[2] R. E. Bank, A. H. Sherman, and A. Weiser. Some refinement algorithms and data structures for regular local mesh refinement. In *Scientific Computing*, Applications of Mathematics and Computing to the Physical Sciences, 1983.

[3] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *Impact Comput. Sci. Engrg.*, 3(3):181–191, 1991.

[4] J. Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995.

[5] F. Boyer and C. Lapuerta. Study of a three component Cahn-Hilliard flow model. *M2AN*, 40(4):653–687, 2006.

[6] F. Boyer, C. Lapuerta, S. Minjeaud, B. Piar, and M. Quintard. Cahn-Hilliard / Navier-Stokes model for the simulation of three-phase flows. 2008. submitted.

[7] F. Boyer and S. Minjeaud. Numerical schemes for a three component Cahn-Hilliard model. 2008. in preparation.

[8] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, jul 1990.

[9] A. Ern and J.-L. Guermond. *Theory and Pratice of Finite Elements*, volume 159 of *Applied Mathmatical Sciences*. Springer, 2004.

[10] Eitan Grinspun, Petr Krysl, and Peter Schröder. CHARMS: A simple framework for adaptive simulation. In *29th International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*, 2002.

[11] J.-L. Guermond and L. Quartapelle. On the approximation of the unsteady Navier-Stokes equations by finite element projection methods. *Numer. Math.*, 80(2):207–238, 1998.

[12] W Hackbusch. *Multigrid Methods and Applications*, volume 4 of *Computational Mathematics*. Springer–Verlag, Berlin, 1985.

[13] P. Hammon and P. Krysl. Implementation of a general mesh refinement technique. Ctu reports, Czech Technical University, Prague, 2003.

[14] M. Jobelin, C. Lapuerta, J.-C. Latché, P. Angot, and B. Piar. A finite element penalty-projection method for incompressible flows. *J. Comput. Phys.*, 217(2):502–518, 2006.

[15] P. Krysl, E. Grinspun, and P. Schröder. Natural hierarchical refinement for finite element methods. *Internat. J. Numer. Methods Engrg.*, 56(8):1109–1124, 2003.

[16] P. Krysl, A. Trivedi, and B. Zhu. Object-oriented hierarchical mesh refinement with CHARMS. *International Journal for Numerical Methods in Engineering*, 60:1401–1424, 2004.

[17] J. M. Maubach. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM Journal on Scientific Computing*, 16(1):210–227, 1995.

[18] W. F. Mitchell. Adaptive refinement for arbitrary finite-element spaces with hierarchical bases. *J. Comput. Appl. Math.*, 36(1):65–78, 1991.

[19] PELICANS. Collaborative Development environment: `https://gforge.irsn.fr/gf/project/pelicans/`.

[20] P.A. Raviart and J.-M. Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles*. MASSON, 2e tirage edition, 1988.

[21] M.-C. Rivara. Mesh refinement processes based on the generalized bisection of simplices. *SIAM J. Numer. Anal.*, 21(3):604–613, 1984.

[22] M.-C. Rivara and G. Iribarren. The 4-triangles longest-side partition of triangles and linear refinement algorithms. *Math. Comp.*, 65(216):1485–1502, 1996.

[23] H. Yserentant. Hierarchical bases. In *ICIAM 91 (Washington, DC, 1991)*, pages 256–276. SIAM, Philadelphia, PA, 1992.

[24] S. Zhang. *Multi-level iteratives techniques*. Reseach report no. 88020, Penn State University, 1988.