THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Mathématiques et Modélisation

École doctorale Information, Structures, Systèmes

Unité de recherche Institut Montpelliérain Alexander Grothendieck

Méthodes Éléments Finis non-conformes adaptées à la conception en temps réel de jumeaux numériques d'organes

Présentée par Killian VUILLEMOT
en Décembre 2025

Sous la direction de Bijan MOHAMMADI, Vanessa LLERAS et Michel DUPREZ.

Devant le jury composé de

Bijan MOHAMMADI, Professeur des Universités, Université de Montpellier	Directeur de thèse
Vanessa LLERAS, Maître de conférence, Université de Montpellier	Co-encadrante
Michel DUPREZ, Chargé de recherche, Inria de l'Université de Lorraine	Co-encadrant
Michel FOURNIÉ, Professeur des Universités, Institut Supérieur de l'Aéronautique et de l'Espace	Rapporteur
Laurent MONASSE, Chargé de recherche, Centre Inria d'Université Côte d'Azur	Rapporteur
Stéphane BORDAS, Full Professor, University of Luxembourg	Examinateur
Emmanuel FRANCK, Chargé de recherche, Inria de l'Université de Lorraine	Examinateur
Lisl WEYNANS, Professeur des Universités, Université de Bordeaux	Présidente du jury

UNIVERSITÉ DE MONTPELLIER

Table des matières

Remerciements

Commençons ces remerciements par mes directeurs de thèse : Vanessa, Bijan et Michel. Merci à vous trois pour votre confiance. Merci de m'avoir permis de travailler sur ce sujet si intéressant et enrichissant. Bijan, merci d'avoir accepté de diriger cette thèse qui m'a tant apporté.

Vanessa, merci pour ton soutien, ton accompagnement et ton accueil toujours très chaleureux et attentionné à chacun de mes séjours Montpelliérains. Tu as su trouver les mots justes lorsque j'en ai eu besoin.

Michel, tu m'as accompagné depuis mon premier stage, il y a maintenant presque 5 ans. À tes côtés, j'ai découvert et appris énormément. Tu m'as notamment transmis ton intérêt pour les méthodes éléments finis (et tu m'as initié à la programmation avec FEniCS, ce qui, je crois, est positif...). Tu m'as aussi appris l'importance des pauses et des discussions entre collègues, en lien ou non avec la science. Merci profondément pour ces années partagées, presque toujours dans la bonne humeur.

Enfin, de manière plus générale, je vous remercie tous les trois de m'avoir intégré dans le projet φ-FEM.

Merci également à Stéphane BORDAS, Lisl WEYNANS, Emmanuel FRANCK et Simon MENDEZ qui ont accepté de faire partie de mon jury de thèse. Je remercie particulièrement Michel FOURNIÉ et Laurent MONASSE, rapporteurs de cette thèse. C'est pour moi un honneur d'avoir pu partager mon travail avec vous.

À toute l'équipe MIMESIS (et MLMS !) actuelle et passée, et en particulier à Stéphane, merci de m'avoir accueilli pendant ces quelques années. L'ambiance était extrêmement agréable et je suis très reconnaissant d'avoir pu faire partie de cette équipe. Merci pour les pauses Tarot, Coinche ou les sorties bar. Enfin, merci pour les discussions (parfois) scientifiques et plus particulièrement à Nikola pour nos échanges toujours très enrichissants.

Frédérique, j'ai vraiment apprécié partager ces années à tes côtés, sur notre « pôle φ-FEM ». Je te souhaite le meilleur pour la suite et te laisse seule entre les mains de Vanessa et de Michel.

Merci à toutes les personnes que j'ai pu croiser lors de mes séjours à l'IMAG, pour votre accueil à chaque fois très agréable. Merci à Tanguy qui a notamment supporté pendant nos années de Licence mes « Moi, je ne ferai jamais de thèse » ... Il faut croire que j'avais tort.

Je souhaite également remercier Alexeï Lozinski et Vincent Vigon, avec qui j'ai pu travailler lors de la préparation d'articles, qui m'ont beaucoup appris. Enfin, je me dois d'adresser un mot de remerciement à Franz Chouly, sans qui rien de cela ne serait arrivé et grâce à qui j'ai rencontré Michel.

A ma famille : merci pour tout. Ma soeur, Cassandre, merci pour ton soutien. A mes parents, qui m'ont toujours encouragé et accompagné : c'est grâce à vous que j'ai pu réaliser cette thèse. Enfin, Pauline, merci d'avoir partagé ces années à mes côtés et de m'avoir toujours poussé à continuer. Merci infiniment.

Table de notations

Nous introduisons dans la Table suivante un ensemble (non exhaustif) de notations qui seront utilisées à de multiples reprises dans ce manuscrit.

	Notation	Signification
Méthodes éléments finis	Ω	Domaine considéré
	$\Gamma = \partial\Omega$	Frontière du domaine Ω
	n	Vecteur normal unitaire extérieur au domaine Ω
	φ	Fonction level-set définissant Ω et Γ
	$\mathcal{O}$	Boîte $[0,1]^2$
	$\mathcal{T}_h$	Maillage de calcul φ-FEM
	h	Taille caractéristique des cellules d'un maillage
	$\mathcal{T}_h^\Gamma$	Cellules de $\mathcal{T}_h$ coupant la frontière Γ
	$\mathcal{F}_h^\Gamma$	Facettes internes de $\mathcal{T}_h^\Gamma$
	σ_D, σ_N, $\gamma_\cdot$	Paramètres de stabilisation et de pénalisation
Élasticité	$\boldsymbol{\sigma}$	Tenseur des contraintes (linéaire)
	$\boldsymbol{\epsilon}$	Tenseur de déformation
	$\boldsymbol{P}$	Premier tenseur de Piola-Kirchhoff
	μ, λ	Constantes de Lamé
FNO	$\mathcal{G}^\dagger$	Opérateur *ground truth*
	θ	Ensemble de paramètres à optimiser
	$\mathcal{G}_\theta$	Opérateur paramétrisé par θ
	$\mathcal{F}$, $\mathcal{F}^{-1}$	Transformation de Fourier discrète et son inverse
	σ	Fonction d'activation non linéaire
	$\mathcal{L}$	Fonctionnelle à minimiser (*loss function*)

Liste des contributions

Cette thèse est supportée par les contributions suivantes :

- **φ-FEM-FNO : a new approach to train a Neural Operator as a fast PDE solver for variable geometries.**
  Michel Duprez, Vanessa Lleras, Alexei Lozinski, Vincent Vigon, Killian Vuillemot. Communications in Nonlinear Science and Numerical Simulation, Volume 152, Part A, January 2026, 10913
  https://www.sciencedirect.com/science/article/pii/S1007570425005428?via%3Dihub
- **φ-FD : A well-conditioned finite difference method inspired by φ-FEM for general geometries on elliptic PDEs.**
  Michel Duprez, Vanessa Lleras, Alexei Lozinski, Vincent Vigon, Killian Vuillemot. Journal of Scientific Computing, 104(1) :1–27, 2025.
  https://link.springer.com/article/10.1007/s10915-025-02914-0
- **φ-FEM for the heat equation : optimal convergence on unfitted meshes in space.**
  Michel Duprez, Vanessa Lleras, Alexei Lozinski, Killian Vuillemot.
  Comptes Rendus. Mathématique, 2023, 361 (G11), pp.1699-1710.
  https://comptes-rendus.academie-sciences.fr/mathematique/item/10.5802/crmath.497.pdf
- **φ-FEM : an efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer.**
  Stéphane Cotin, Michel Duprez, Vanessa Lleras, Alexei Lozinski, Killian Vuillemot. Partition of Unity Methods, Stéphane Bordas ; Alexander Menk ; Sundararajan Natarajan. Wiley, pp.191-216, 2022, Wiley Series in Computational Mechanics, 978-0470667088.
  https://hal.science/hal-03372733

Les codes Python (FEniCS et FEniCSX) développés en tant que contributeur principal durant cette thèse sont disponibles sur les pages GitHub suivantes :

https://github.com/KVuillemot
https://github.com/PhiFEM

Plan du manuscrit

Le premier chapitre de ce manuscrit est consacré à une introduction générale. Nous y exposerons le contexte scientifique de cette thèse, accompagné d'une revue de méthodes existantes dans la littérature. Cela permettra de justifier le choix de la méthode φ-FEM comme méthode principale de ce travail. En fin de chapitre, nous présenterons en détail cette méthode appliquée à la résolution du problème de Poisson : d'abord avec conditions de Dirichlet, telle qu'introduite dans [28], puis avec conditions de Neumann, selon l'approche développée dans [23].

Le second chapitre propose une extension de la méthode φ-FEM à diverses équations aux dérivées partielles. Nous débuterons par une nouvelle formulation pour le problème de Poisson avec conditions de Dirichlet, avant d'introduire deux versions de φ-FEM capables de traiter efficacement des conditions mixtes Dirichlet-Neumann, y compris dans des contextes présentant des singularités. Nous poursuivrons par une étude théorique et numérique d'un schéma adapté à l'équation de la chaleur avec conditions de Dirichlet, développée dans le cadre de l'article « *φ-FEM for the heat equation : optimal convergence on unfitted meshes in space* », en collaboration avec Michel Duprez, Vanessa Lleras et Alexei Lozinski (cf. [27]). La suite du chapitre sera dédiée aux problèmes d'élasticité linéaire, notamment les cas d'interfaces et de fractures. Les trois premières parties de cette section s'appuient sur les travaux présentés dans l'article « *φ-FEM : an efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer* », réalisé en collaboration avec Stéphane Cotin, Michel Duprez, Vanessa Lleras et Alexei Lozinski (cf. [22]). Enfin, nous introduirons de nouveaux cas tests pour les conditions mixtes Dirichlet-Neumann, puis nous adapterons le schéma φ-FEM traitant le cas de conditions mixtes à des problèmes d'élasticité non-linéaire.

Le troisième chapitre de ce manuscrit sera consacré à une adaptation de l'idée utilisée pour φ-FEM au cas des différences finies. Ce chapitre sera issu de l'article « *φ-FD : A well-conditioned finite difference method inspired by φ-FEM for general geometries on elliptic PDEs* » publié en collaboration avec Michel Duprez, Vanessa Lleras, Alexei Lozinski et Vincent Vigon (cf. [25]). On proposera alors un nouveau schéma différences finies pour lequel une étude théorique sera proposée. Cette méthode sera ensuite comparée numériquement à la littérature différences finies ainsi qu'aux approches éléments finis.

Dans un quatrième chapitre, nous nous intéresserons à des combinaisons entre méthodes éléments finis et réseaux de neurones. Nous considérerons alors différentes équations et proposerons une nouvelle approche combinant les avantages de la méthode φ-FEM ainsi que la rapidité d'évaluation des réseaux de neurones. Ce chapitre sera majoritairement issu de l'article « *φ-FEM-FNO : a new approach to train a Neural Operator as a fast PDE solver for variable geometries* » (cf. [26]).

Dans le dernier chapitre, nous présenterons différents outils utilisés au cours de cette thèse, permettant d'utiliser des fonctions level-set pour construire des maillages « conformes » ainsi qu'une première méthode permettant de reconstruire des approximations de fonctions level-set à partir d'images binaires. Nous présenterons ensuite une nouvelle méthode type *multigrid* combinée à l'approche φ-FEM. Enfin, dans une dernière section, nous proposerons une nouvelle approche hybride réseaux de neurone et éléments finis en combinant la rapidité de φ-FEM-FNO et la précision de φ-FEM-Multigrid.

1 Introduction

Chapitre 1 – Introduction

1.1 Contexte et outils

Les équations aux dérivées partielles (EDP) jouent un rôle fondamental dans la modélisation d'une grande variété de phénomènes physiques, biologiques et mécaniques, en particulier dans le domaine de la biomécanique. Elles permettent de décrire des systèmes complexes pour lesquels les solutions analytiques sont généralement inaccessibles, notamment en présence de géométries irrégulières ou de conditions aux limites non triviales. La résolution numérique de ces équations revêt donc une importance majeure, avec un besoin croissant d'algorithmes rapides, voire capables de fonctionner en temps réel.

Parmi les approches les plus utilisées pour la résolution d'EDP, la méthode des éléments finis (MEF, ou FEM pour Finite Element Method) (voir par exemple [36, 32, 10] pour une présentation détaillée) occupe une place centrale. Néanmoins, cette méthode rencontre des limitations importantes lorsqu'il s'agit de traiter des géométries complexes, comme celles des organes, car elle repose sur la construction de maillages conformes. Cette étape de maillage, souvent délicate, constitue une difficulté majeure à l'application, notamment en temps réel, de cette méthode.

Pour contourner cette difficulté, des approches dites non conformes ont été développées. Ces méthodes, souvent regroupées sous les appellations de méthodes aux frontières immergées (Immersed Boundary Methods, IBM) [67] ou de domaines fictifs [39], permettent de s'affranchir de la nécessité de construire un maillage épousant précisément la frontière. Au fil des années, ces approches ont connu de nombreuses améliorations. Si les premières versions souffraient souvent d'un manque de précision, les développements plus récents ont permis d'atteindre des niveaux de performance bien supérieurs, parfois au prix d'une complexité d'implémentation accrue.

Dans ce contexte, il est pertinent de présenter certaines de ces méthodes non conformes. Cette analyse mènera alors naturellement à la présentation de la méthode qui sera au centre de ce manuscrit : la méthode φ-FEM.

Pour cela, nous considérons le problème de Poisson avec conditions de Dirichlet homogènes au bord, donné par

$$\begin{cases} -\Delta u &= f\,, \quad \text{dans } \Omega\,, \\ u &= 0\,, \quad \text{sur } \Gamma\,, \end{cases} \tag{1.1}$$

avec $\Omega \subset \mathbb{R}^d$ (ici $d = 2, 3$) un domaine de frontière Γ, $f \in L^2(\Omega)$ et la normale unitaire extérieure à Ω, n.

1.1.1 Méthode éléments finis standard

Une méthode classique pour résoudre (1.1) est la méthode des éléments finis (que l'on appellera par la suite « Standard-FEM ») utilisant des maillages conformes. Soit $v \in H_0^1(\Omega)$ une fonction test avec

$$H_0^1(\Omega) = \{u \in H^1(\Omega)\,|u = 0 \text{ sur } \Gamma\}\,,$$

la formulation faible de l'équation (1.1) est obtenue par multiplication par v et intégration par partie, ce qui donne le problème : trouver $u \in H_0^1(\Omega)$ vérifiant

$$\int_\Omega \nabla u \cdot \nabla v - \underbrace{\int_{\partial\Omega} \frac{\partial u}{\partial n} v}_{=0} = \int_\Omega f v\,, \forall\ v \in H_0^1(\Omega)\,.$$

Cela nous donne alors une formulation continue, que l'on discrétise afin de résoudre le problème numériquement. On considère un domaine polygonal $\Omega \subset \mathbb{R}^d$ dont la frontière Γ peut être exactement représentée par un maillage conforme $\mathcal{T}_h$, de taille h et constitué d'éléments finis simples (par exemple des triangles, tétraèdres), tel que :

$$\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K}.$$

On représente par exemple un tel maillage pour le cas d'un domaine circulaire à la Figure 1.1.

Remarque 1.1. Par la suite, on dira qu'un maillage $\mathcal{T}_h$ est de taille h lorsque $\mathrm{diam}(T) \leqslant h$ pour tout $T \in \mathcal{T}_h$. En pratique, on essaiera de construire des maillages aussi réguliers que possible, c'est-à-dire des maillages où la variation de $\mathrm{diam}(T)$ entre les cellules est minimale. On considèrera des maillages géométriquement qualitatifs au sens de Ciarlet [19].

Soit maintenant l'espace éléments finis de Lagrange de degré $k \in \mathbb{N}^*$, sur le maillage $\mathcal{T}_h$ défini par

$$V_h = \{v_h \in \mathcal{C}^0(\overline{\Omega})\ |v_{h|T} \in \mathbb{P}_k(T)\,, \forall\ T \in \mathcal{T}_h\}\,, \tag{1.2}$$

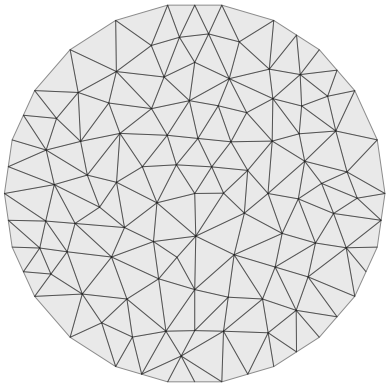

FIGURE 1.1 – Maillage conforme pour une méthode éléments finis standard.

où $\mathbb{P}_k(T)$ est l'espace des polynômes de degré inférieur ou égal à k. On notera V_h^0 l'espace homogène associé, incluant la contrainte $u_h = 0$ sur le bord de Ω. Finalement, on peut introduire la version discrétisée de la formulation faible : trouver $u_h \in V_h$, telle que

$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f_h v_h \,, \forall\ v_h \in V_h^0 \,. \tag{1.3}$$

La discrétisation du problème conduit à un système linéaire, qui peut être résolu efficacement par des méthodes numériques standards.

1.1.2 Méthodes non conformes

Intéressons-nous maintenant aux techniques basées sur les éléments finis non conformes. Les approches initiales telles que [67, 39, 38] manquent de précision en raison de leur traitement simplifié des conditions aux limites et produisent également des matrices mal conditionnées. Au cours des deux dernières décennies, des méthodes plus précises ont été développées, notamment la méthode des éléments finis étendus XFEM [69, 46] initialement introduite pour des problèmes d'interfaces ou de fractures, ou encore les méthodes CutFEM [15, 16, 13] et « Shifted Boundary Method » (SBM) [63]. Ces méthodes présentent généralement une convergence optimale et les matrices associées sont bien conditionnées, mais elles nécessitent des règles de quadrature non standard ou des extrapolations pour assembler les matrices, pouvant rendre les implémentations numériques complexes. Plus récemment, pour éviter ces contraintes, les auteurs de [28] ont développé une méthode non conforme appelée φ-FEM, qui utilise une fonction level-set pour décrire le domaine.

Afin de comprendre l'intérêt de l'approche φ-FEM par rapport à d'autres méthodes non conformes, il est intéressant de présenter brièvement certaines de ces méthodes.

L'une des premières méthodes, notamment introduite par [67, 39, 38], dont l'idée est d'étendre la solution u du problème considéré à un maillage cartésien contenant le domaine Ω, a notamment l'inconvénient d'être relativement lourde numériquement.

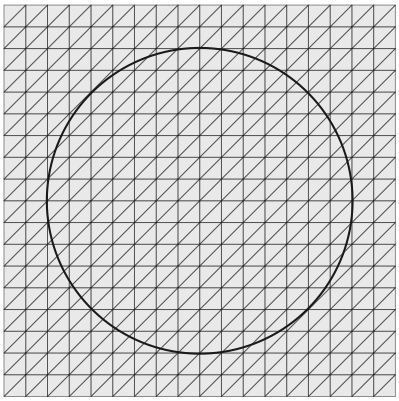

FIGURE 1.2 – Exemple de grille cartésienne $\mathcal{T}_h$ contenant une géométrie complexe.

Cela est dû à la résolution qui s'effectue sur l'ensemble de la grille cartésienne puisque le schéma éléments finis correspondant est donné par : trouver $u_h \in V_h$ et $\lambda_h \in M_h$ tels que

$$\begin{aligned} a(u_h, v_h) + b(v_h, \lambda_h) &= l(v_h)\,, \quad \forall\ v_h \in V_h\,, \\ b(u_h, \mu_h) &= 0\,, \qquad \forall\ \mu_h \in M_h\,, \end{aligned}$$

avec

$$a(u,v) = \int_{\Omega_h^{\mathcal{O}}} \nabla u \cdot \nabla v\,, \quad b(v,\lambda) = \langle v, \lambda \rangle\,, \quad \text{et } l(v) = \int_{\Omega_h^{\mathcal{O}}} fv\,,$$

où $\Omega_h^{\mathcal{O}}$ est le domaine couvrant un maillage cartésien $\mathcal{T}_h$, comme représenté à la Figure 1.2, V_h défini par (1.2) et $M_h = \{\mu_h : \mu_{h|S} \in \mathbb{P}_0(S)\,, \forall S \in S_\Gamma\}$ avec S_Γ une subdivision de la frontière Γ.

Une autre méthode plus complexe mais, offrant de très bons résultats (tant théoriques que numériques) a été introduite plus récemment. Cette méthode, nommée CutFEM [13, 14, 44, 16] utilise également l'idée d'immerger le domaine considéré dans un maillage cartésien. Cependant, on ne considère ici qu'une partie des cellules de la grille, celles en intersection avec le domaine physique Ω afin de construire le maillage $\mathcal{T}_h$, comme représenté à la Figure 1.3. On obtient alors des cellules coupées par la frontière, i.e. des cellules contenant une partie à l'intérieur du domaine et une partie à l'extérieur du domaine. Pour prendre en compte ces cellules dans le schéma éléments finis, des termes de stabilisation ont été introduits.

Le schéma est une nouvelle fois construit à partir d'une intégration par parties de (1.1), que l'on peut, puisque $u = 0$ sur $\partial\Omega$, combiner avec les expressions suivantes :

$$\int_{\partial\Omega} u\partial_n v = 0\,, \quad \int_{\partial\Omega} uv = 0$$

et

$$G(u,v) = \sum_{E \in \mathcal{F}_h^\Gamma} \int_E [\partial_n u]\,[\partial_n v]\ ,$$

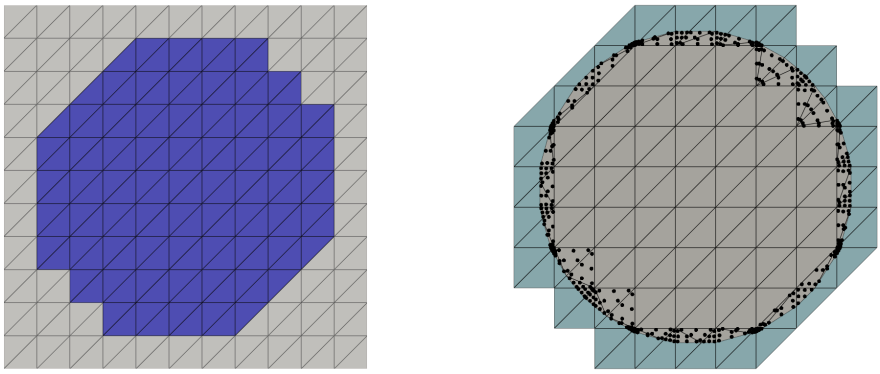

FIGURE 1.3 – Exemple de maillage considéré pour l'approche CutFEM. Gauche : grille et cellules sélectionnées. Droite : domaine coupé considéré lors de l'intégration par parties (gris), où les points noirs sont les points de quadrature utilisés sur les cellules coupées.

où $\mathcal{F}_h^\Gamma$ est l'ensemble des faces internes du maillage $\mathcal{T}_h$, appartenant à des cellules coupées par Γ. Le schéma CutFEM pour résoudre (1.1) est finalement donné par : trouver $u_h \in V_h$ tel que

$$\int_\Omega \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega} \partial_n u_h v_h - \int_{\partial\Omega} u_h \partial_n v_h + \frac{\gamma}{h} \int_{\partial\Omega} u_h v_h + G(u_h, v_h) = \int_\Omega f v_h \,, \forall v_h \in V_h \,.$$

On remarque alors en particulier dans cette formulation que, contrairement à la méthode précédente, l'intégration par parties est réalisée sur le domaine physique. Ainsi, cela génère une complexité d'implémentation plus élevée. On trouve notamment un package spécialement développé pour cela, le package CutFEMx [1], dépendant de DolfinX, qui permet d'implémenter la méthode.

1.2 La méthode φ-FEM

Une autre méthode, introduite plus récemment dans [28] sous le nom de φ-FEM, propose de résoudre (1.1) en imposant les conditions de bord à l'aide d'une fonction level-set caractérisant la géométrie et sa frontière. Cette approche a par la suite été étendue aux conditions de bord de Neumann dans [23]. Plusieurs autres variantes de la méthode ont ensuite été proposées par exemple pour résoudre le problème de Stokes dans [24] ou des problèmes d'élasticité linéaire dans [22]. Nous allons ici rappeler la méthode φ-FEM pour résoudre le problème de Poisson, dans un premier temps avec conditions de Dirichlet, puis conditions de Neumann au bord. Nous ne présenterons pas les aspects théoriques qui ont été proposés dans [28, 23], mais seulement les méthodes, afin d'assurer une bonne compréhension de la suite du manuscrit.

1. `https://github.com/sclaus2/CutFEMx`

1.2.1 Conditions de Dirichlet

On considère un domaine $\Omega \subset \mathcal{O} = [0,1]^2 \subset \mathbb{R}^2$, défini par une fonction level-set φ telle que

$$\Omega := \{\varphi < 0\} \qquad \text{et} \qquad \Gamma := \{\varphi = 0\}\,, \tag{1.4}$$

avec Γ la frontière de Ω. L'idée principale de la méthode φ-FEM repose sur cette représentation du domaine permettant de considérer

$$u = \varphi w\,, \tag{1.5}$$

et ainsi de chercher une solution w telle que φw vérifie l'équation (1.1). Par construction, $\varphi w = 0$ sur Γ et donc u satisfait automatiquement les conditions de Dirichlet.

Soit $\mathcal{T}_h^{\mathcal{O}}$ un maillage triangulaire cartésien de $\mathcal{O}$, dont la taille de cellule est h. Soit également $\varphi_h = I_{h,\mathcal{O}}^{(l)}\varphi$ l'interpolation continue de Lagrange de φ (de degré $l > 0$) sur $\mathcal{T}_h^{\mathcal{O}}$, avec $I_{h,\mathcal{O}}^{(l)}$ l'opérateur d'interpolation de Lagrange sur l'espace éléments finis de degré l sur $\mathcal{T}_h^{\mathcal{O}}$.

On construit alors à l'aide de φ_h un sous-maillage $\mathcal{T}_h$ de $\mathcal{T}_h^{\mathcal{O}}$ contenant toutes les cellules intersectant le domaine $\{\varphi_h < 0\}$, i.e.

$$\mathcal{T}_h := \left\{ T \in \mathcal{T}_h^{\mathcal{O}} :\ T \cap \{\varphi_h < 0\} \neq \emptyset \right\}. \tag{1.6}$$

Introduisons également un second sous-maillage, cette fois de $\mathcal{T}_h$, contenant les cellules coupées par la frontière, i.e. $\{\varphi_h = 0\}$, donné par

$$\mathcal{T}_h^{\Gamma} := \{T \in \mathcal{T}_h \ \ T \cap \{\varphi_h = 0\} \neq \emptyset\}\,. \tag{1.7}$$

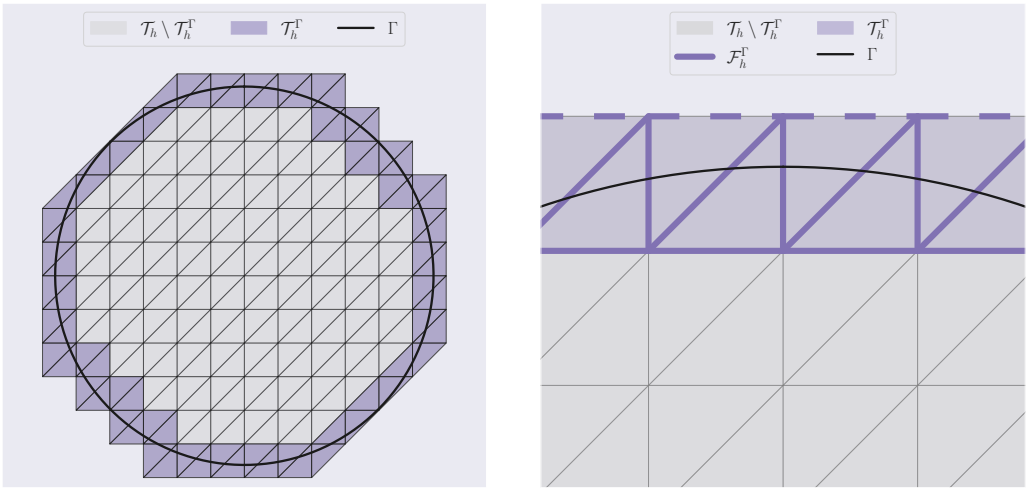

FIGURE 1.4 – Gauche : représentation des ensembles $\mathcal{T}_h$ et $\mathcal{T}_h^{\Gamma}$. Droite : représentation de $\mathcal{F}_h^{\Gamma}$ sur le même exemple.

On notera par la suite Ω_h et Ω_h^Γ les domaines occupés par les maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ respectivement ainsi que $\partial\Omega_h$ la frontière de Ω_h (qui est différente de $\partial\Omega = \Gamma$ et de $\Gamma_h = \{\varphi_h = 0\}$). Un exemple de représentation des maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ est donné à la Figure 1.4 (gauche).

Il est également nécessaire de construire un ensemble $\mathcal{F}_h^\Gamma$, contenant les facettes internes du maillage $\mathcal{T}_h^\Gamma$, défini par

$$\mathcal{F}_h^\Gamma := \{F \in \mathcal{T}_h^\Gamma \setminus \partial\Omega_h\}\,. \tag{1.8}$$

Sur la Figure. 1.4 (droite), ces facettes sont représentées en violet (trait plein) et les facettes de $\partial\Omega_h$ en traits discontinus.

Soit $k \geqslant 1$, un entier. L'espace éléments finis de degré k sur $\mathcal{T}_h$ est défini par

$$V_h^{(k)} := \{v_h \in H^1(\Omega_h)\ \ v_h|_T \in \mathbb{P}_k(T)\ \forall\ T \in \mathcal{T}_h\}\,. \tag{1.9}$$

Le schéma φ-FEM pour résoudre (1.1) est finalement donné par : trouver $w_h \in V_h^{(k)}$ telle que, pour tout $s_h \in V_h^{(k)}$ avec $u_h = \varphi_h w_h$ et $v_h = \varphi_h s_h$,

$$\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h} \frac{\partial u_h}{\partial n} v_h + G_h(u_h, v_h) = \int_{\Omega_h} f_h v_h + G_h^{rhs}(v_h)\,, \tag{1.10}$$

où f_h est l'interpolation de Lagrange de f sur $V_h^{(k)}$,

$$G_h(u,v) = \sigma_D h \sum_{E\in\mathcal{F}_h^\Gamma} \int_E \left[\frac{\partial u}{\partial n}\right]\left[\frac{\partial u}{\partial n}\right] + \sigma_D h^2 \sum_{T\in\mathcal{T}_h^\Gamma} \int_T \Delta u \Delta v\,, \tag{1.11}$$

et

$$G_h^{rhs}(v) = -\sigma_D h^2 \sum_{T\in\mathcal{T}_h^\Gamma} \int_T f_h \Delta v\,. \tag{1.12}$$

Les crochets $[\cdot]$ dans l'expression de G_h correspondent aux sauts sur les facettes de $\mathcal{F}_h^\Gamma$, i.e.

$$\left[\frac{\partial u}{\partial n}\right] = (\nabla u^+ - \nabla u^-)\cdot n\,,$$

où $\frac{\partial u}{\partial n} = \nabla u \cdot n$ est la dérivée normale de u et $\sigma_D > 0$ est un paramètre de stabilisation indépendant de h. Le premier terme de (1.11) a été introduit dans [12], sous le nom de « Ghost penalty » et a été notamment utilisé dans l'approche CutFEM [13].

Remarque 1.2 (Conditions non homogènes). Dans le cas de conditions de bord non homogènes ($u = u_D$ sur Γ avec u_D non nulle), u_h deviendra $u_h = \varphi_h w_h + u_D$.

Remarque 1.3. Par la suite, nous ferons régulièrement référence à ce schéma sous le nom de schéma direct, par opposition à sa variante duale qui sera présentée en Section 2.1.

Dans [28], le théorème de convergence suivant a été prouvé :

Théorème 1.1 (c.f. [28, Théorème 2.1])**.** *Sous les hypothèses [28, Hypothèse 1] et [28, Hypothèse 2], pour $l \geqslant k$, un maillage $\mathcal{T}_h^\Gamma$ quasi-uniforme, une fonction $f \in H^k(\Omega_h \cup \Omega)$. Soient $u \in H^{k+2}(\Omega)$ solution exacte du problème* (1.1) *et w_h solution approchée du*

problème (1.10). *Soit* $u_h = \varphi_h w_h$ *(la solution approchée de* (1.1)*). Alors, il existe une constante positive* C *telle que :*

$$|u - u_h|_{1,\Omega_h \cap \Omega} \leqslant C h^k \|f\|_{k,\Omega \cup \Omega_h} ,$$

et si $\Omega \subset \Omega_h$*, alors :*

$$\|u - u_h\|_{0,\Omega} \leqslant C h^{k+\frac{1}{2}} \|f\|_{k,\Omega_h} ,$$

où $\|\cdot\|_{0,\mathcal{O}}$ *désigne la norme* L^2 *sur* $\mathcal{O}$*,* $|\cdot|_{1,\mathcal{O}}$ *la semi-norme* H^1 *et* $\|\cdot\|_{k,\mathcal{O}}$ *la norme* k*.*

Ainsi, la méthode φ-FEM offre une convergence sous-optimale pour la norme L^2 et optimale pour la norme H^1 (i.e. l'erreur converge selon le même ordre que l'erreur d'interpolation), sous certaines hypothèses sur la régularité de la frontière Γ et des maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$. De plus, les résultats numériques ont montré un ordre de convergence optimal également pour l'erreur en norme L^2. Enfin, le bon conditionnement de la matrice éléments finis associée au schéma a été démontré et illustré numériquement.

1.2.2 Conditions de Neumann

Par la suite, nous aurons également besoin de traiter des conditions de Neumann. Dans [23], un schéma φ-FEM permettant de résoudre l'équation

$$\begin{cases} -\Delta u + u &= f\,, \quad \text{dans } \Omega\,, \\ \nabla u \cdot n &= 0\,, \quad \text{sur } \Gamma_N = \Gamma\,, \end{cases} \tag{1.13}$$

a été introduit. Pour traiter ce cas, il est nécessaire d'introduire différentes variables auxiliaires permettant d'utiliser la fonction level-set cette fois en la reliant à ∇u. On considère une fois de plus les domaines Ω_h et Ω_h^Γ ainsi que les maillages correspondant $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$. Soit également $\mathcal{F}_h^{N_s} = \partial(\mathcal{T}_h \setminus \mathcal{T}_h^\Gamma)$, l'ensemble des faces entre $\mathcal{T}_h \setminus \mathcal{T}_h^\Gamma$ et $\mathcal{T}_h^\Gamma$. Les différents maillages et ensembles de faces sont représentés à la Figure 1.5.

Soient les espaces éléments finis

$$Z_h^{(k)}(O) := \left\{ z_h : O \to \mathbb{R}^d \ : z_{h|T} \in \mathbb{P}^k(T)^d \ \forall T \in \mathcal{T}_h^O,\ z_h \text{ continue sur } O \right\} \tag{1.14}$$

et

$$Q_h^{(l)}(O) := \left\{ q_h : O \to \mathbb{R} : q_{h|T} \in \mathbb{P}^l(T) \ \ \forall T \in \mathcal{T}_h^O,\ q_h \text{ continue sur } O \text{ si } l \geq 0 \right\} , \tag{1.15}$$

avec $O \subset \Omega_h$ et $\mathcal{T}_h^O$ le maillage couvrant le domaine O. Les conditions de bord seront alors imposées via les variables y et p introduites de sorte que

$$y = -\nabla u, \quad \text{sur } \Omega_h^\Gamma\,, \tag{1.16}$$

$$\operatorname{div} y + u = f\,, \quad \text{sur } \Omega_h^\Gamma\,, \tag{1.17}$$

$$y \cdot \nabla \varphi + p\varphi = 0, \quad \text{sur } \Omega_h^\Gamma\,, \tag{1.18}$$

où l'on a utilisé le fait que $n = \nabla\varphi/|\nabla\varphi|$ sur Γ pour établir (1.18).

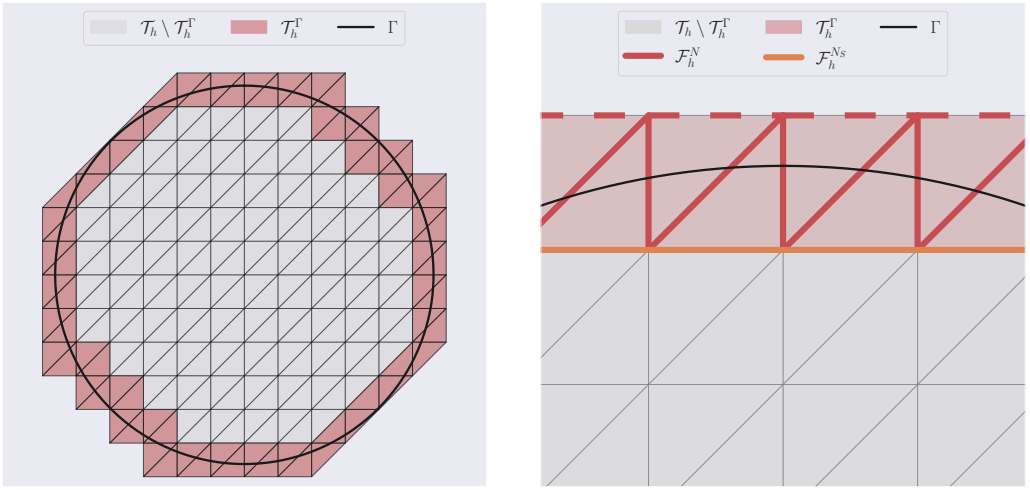

FIGURE 1.5 – Exemple de représentation des ensembles $\mathcal{T}_h$, $\mathcal{T}_h^\Gamma$, $\mathcal{F}_h^\Gamma$ et $\mathcal{F}_h^{\Gamma_s}$.

Le problème est finalement de trouver $(u_h, y_h, p_h) \in V_h^{(k)} \times Z_h^{(k)}(\Omega_h^\Gamma) \times Q_h^{(k-1)}(\Omega_h^\Gamma)$ tel que

$$\begin{aligned}
&\int_{\Omega_h} \nabla u_h \cdot \nabla v_h + \int_{\partial\Omega_h} y_h \cdot n\, v_h + \int_{\Omega_h} u_h\, v_h \\
&\quad + \gamma_{div} \int_{\Omega_h^\Gamma} (\operatorname{div} y_h + u_h) \cdot (\operatorname{div} z_h + v_h) + \gamma_u \int_{\Omega_h^\Gamma} (y_h + \nabla u_h) \cdot (z_h + \nabla v_h) \\
&\quad + \frac{\gamma_p}{h^2} \int_{\Omega_h^\Gamma} \left(y_h \cdot \nabla \varphi_h + \frac{1}{h} p_h \varphi_h \right) \cdot \left(z_h \cdot \nabla \varphi_h + \frac{1}{h} q_h \varphi_h \right) \\
&\quad + \sigma_N h \sum_{E \in \mathcal{F}_h^{N_s}} \int_E \left[\frac{\partial u_h}{\partial n} \right] \left[\frac{\partial v_h}{\partial n} \right] \\
&\quad = \int_{\Omega_h} f_h v_h + \gamma_{div} \int_{\Omega_h^\Gamma} f_h \cdot (\operatorname{div} z_h + v_h)\,, \\
&\qquad \forall (v_h, z_h, q_h) \in V_h^{(k)} \times Z_h^{(k)}(\Omega_h^\Gamma) \times Q_h^{(k-1)}(\Omega_h^\Gamma)\,. \quad (1.19)
\end{aligned}$$

Le schéma (1.19) est obtenu après intégration par parties et ajout des équations (1.16)-(1.18) sous la forme des moindres carrés. On retrouve également la Ghost penalty appliquée sur les faces de $\mathcal{F}_h^{N_S}$. Enfin, les termes multipliés par γ_{div} sont les termes de stabilisation d'ordre 2.

Remarque 1.4 (Conditions de Neumann non homogènes). Dans le cas de conditions de Neumann non homogènes, $\frac{\partial u}{\partial n} = g$ sur Γ, l'équation (1.18) sera modifiée par

$$y \cdot \nabla \varphi + p\varphi = \tilde{g} |\nabla \varphi|\,,$$

avec $\tilde{g}$ un prolongement de g de Γ au voisinage de Γ, et le schéma sera adapté en conséquence.

Remarque 1.5 (Aspects software et hardware). Dans la suite de ce manuscrit, sauf mention explicite du contraire, toutes les simulations ont été exécutées avec un processeur `Intel Core i7-12700H`, avec 32Gb de mémoire RAM ainsi qu'un GPU `NVIDIA RTX A2000` avec 8Gb de mémoire. De plus, toutes les simulations éléments finis ont été réalisées avec des implémentations Python à l'aide de la librairie FEniCS [2] (version 2019.1.0) et de son évolution, la librairie FEniCSx [5, 3, 80, 79] (version 0.8.0).

2 Les nouveaux schémas φ-FEM

Résumé

Ce chapitre est consacré à la résolution de plusieurs EDP avec la méthode φ-FEM. Dans un premier temps, nous introduirons un nouveau schéma φ-FEM traitant l'équation de Poisson avec conditions de Dirichlet. Nous présenterons ensuite une combinaison de ce schéma avec deux variantes d'imposition de conditions de Neumann afin de traiter des problèmes de conditions mixtes. Nous proposerons ensuite une méthode φ-FEM permettant de résoudre l'équation de la chaleur avec conditions de Dirichlet homogènes au bord, introduite dans [22, 27]. Enfin, nous proposerons plusieurs schémas permettant de résoudre divers problèmes d'élasticité linéaire (présentés dans [22]) et non-linéaire.

Chapitre 2 – Les nouveaux schémas φ-FEM

2.1 Le schéma φ-FEM « dual »

Intéressons-nous à présent à un second schéma φ-FEM permettant de résoudre l'équation de Poisson avec conditions de Dirichlet homogènes (1.1). Contrairement au premier schéma présenté en (1.10), celui-ci s'inspire du schéma Neumann (1.19), en introduisant une variable auxiliaire localisée sur la bande Ω_h^Γ.

On considère donc l'équation (1.1) définie sur un domaine Ω inclus dans une boîte $\mathcal{O}$, sur laquelle est construit un maillage cartésien $\mathcal{T}_h^{\mathcal{O}}$. Le domaine Ω et sa frontière Γ sont décrits à l'aide d'une fonction level-set φ (voir (1.4)), permettant de construire les maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ (cf. (1.6) et (1.7)), ainsi que les sous-domaines associés Ω_h et Ω_h^Γ.

Dans ce nouveau schéma, les conditions de Dirichlet sont imposées par pénalisation à l'aide d'une variable auxiliaire p définie sur Ω_h^Γ, via l'équation

$$u = \varphi p\,, \quad \text{sur } \Omega_h^\Gamma\,. \tag{2.1}$$

Ainsi, soit $k \geqslant 0$. La variable principale du problème, u sera discrétisée par $u_h \in V_h^{(k)}$ (cf. (1.9)) et la variable auxiliaire p par $p_h \in Q_h^{(k)}(\Omega_h^\Gamma)$ (cf. (1.15)). Le schéma φ-FEM dual pour (1.1) est alors donné par : trouver $u_h \in V_h^{(k)}$ et $p_h \in Q_h^{(k)}(\Omega_h^\Gamma)$ tels que

$$\begin{aligned}\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h} \frac{\partial u_h}{\partial n} v_h + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma} (u_h - \frac{1}{h}\varphi_h p_h)(v_h - \frac{1}{h}\varphi_h q_h) \\ + G_h^{lhs}(u_h, v_h) = \int_{\Omega_h} f v_h + G_h^{rhs}(v_h)\,, \quad \forall v_h \in V_h^{(k)}, q_h \in Q_h^{(k)}(\Omega_h^{\Gamma_D})\,,\end{aligned} \tag{2.2}$$

où G_h^{lhs} et G_h^{rhs} sont les termes de stabilisation introduits dans (1.11) et (1.12) respectivement.

Remarque 2.1 (Conditions de Dirichlet non homogènes). On reconnait la formulation de départ utilisée pour construire le schéma direct (1.10) ($u = \varphi w$ sur Ω), imposée localement via l'équation (2.1). Ainsi, pour le cas de conditions non homogènes, i.e. $u = u_D$ sur Γ, il suffit d'appliquer le même principe et d'imposer $u = \varphi p + u_D$ dans le schéma ce qui modifie uniquement l'intégrale sur Ω_h^Γ, qui devient

$$\frac{\gamma}{h^2}\int_{\Omega_h^\Gamma} (u_h - \frac{1}{h}\varphi_h p_h - u_D)(v_h - \frac{1}{h}\varphi_h q_h)\,.$$

Remarque 2.2 (Schéma direct et schéma dual). Les relations (1.5) et (2.1) semblent analogues, mais leur rôle diffère sensiblement. Dans le schéma direct, la variable w remplace entièrement u, tandis que dans la version duale, la variable auxiliaire p vient en

complément de u, ce qui augmente légèrement le coût du calcul. Ce coût supplémentaire reste limité puisque p est restreinte à la bande Ω_h^Γ, de taille h.

Cependant, la version duale présente deux avantages notables. D'une part, elle utilise dans la formulation la fonction φ localement autour de la frontière, et non sur l'ensemble du domaine. D'autre part, elle est naturellement compatible avec le schéma Neumann, ce qui en fait un outil particulièrement adapté pour le traitement de conditions mixtes Dirichlet/Neumann, que nous aborderons par la suite. À l'inverse, la version directe du schéma, a l'avantage d'offrir une sorte de correction de la solution lors de la multiplication de w par φ, ce qui offre généralement de meilleurs résultats numériquement.

2.1.1 Analyse théorique : résultats principaux et lemmes importants

Dans un premier temps, rappelons les hypothèses sur le domaine et le maillage, issues de [28], nécessaires à l'étude de convergence du schéma (2.2).

Hypothèse 2.1.1. La frontière Γ peut être recouverte par des ouverts $\mathcal{O}_i$, $i = 1, \dots, I$ sur lesquels on peut introduire des coordonnées locales $\xi_1, \dots, \xi_d$ avec $\xi_d = \varphi$ telles que, jusqu'à l'ordre $k+1$, toutes les dérivées partielles $\partial^\alpha \xi_i / \partial x^\alpha$ et $\partial x^\alpha / \partial^\alpha \xi_i$ sont bornées par une constante $C_0 > 0$. Ainsi, sur $\mathcal{O}$, φ est de classe C^{k+1} et il existe $m > 0$ tel que $|\varphi| \geqslant m$ sur $\mathcal{O} \setminus \cup_{i=1,\dots,I} \mathcal{O}_i$.

Hypothèse 2.1.2. La frontière approchée, définie par $\Gamma_h = \{\varphi_h = 0\}$ peut être recouverte par des patchs d'éléments $\{\Pi_r\}_{r=1,\dots,N_\Pi}$ tels que :

- Chaque patch Π_r peut s'écrire $\Pi_r = \Pi_r^\Gamma \cup T_r$ où $\Pi_r^\Gamma \subset \mathcal{T}_h^\Gamma$ et $T_r \in \mathcal{T}_h \setminus \mathcal{T}_h^\Gamma$. De plus Π_r, comporte au plus M éléments qui sont connectés avec M indépendant de h ;
- Le maillage $\mathcal{T}_h^\Gamma$ vérifie $\mathcal{T}_h^\Gamma = \cup_{r=1,\dots,N_\Pi} \Pi_r^\Gamma$;
- Deux patchs Π_r et Π_s sont disjoints si $r \neq s$.

Ces hypothèses sont satisfaites lorsque la frontière Γ est suffisamment régulière, et le maillage $\mathcal{T}_h$ suffisamment fin.

Nous allons maintenant énoncer le théorème de convergence du schéma (2.2)

Théorème 2.1. *On suppose que les hypothèses 2.1.1 et 2.1.2 sont satisfaites, $k > 0$ et $f \in H^{k-1}(\Omega_h)$. Enfin, on suppose $\Omega \subset \Omega_h$. Soit $u \in H^{k+1}(\Omega)$ la solution de* (1.1)*. La solution de* (2.2) *$u_h \in V_h^{(k)}$ satisfait*

$$|u - u_h|_{1,\Omega} \leqslant C h^k \|f\|_{k-1,\Omega_h} \quad \text{et} \quad \|u - u_h\|_{0,\Omega} \leqslant C h^{k+1/2} \|f\|_{k-1,\Omega_h},$$

où $C > 0$ est une constante.

Dans un premier temps, nous rappelons plusieurs lemmes de [28] et [23], qui seront nécessaires dans les preuves suivantes.

Lemme 2.1 (cf. [28, Lemme 3.3])**.** *Sous l'hypothèse [28, Assumption 2], pour tout $\beta > 0$ et $s \in \mathbb{N}^*$, il est possible de choisir $a \in]0,1[$ dépendant uniquement de la régularité du maillage et de s, tel que pour tout $v_h \in V_h^{(s)}$*

$$|v_h|^2_{1,\Omega_h^\Gamma} \leqslant \alpha |v_h|^2_{1,\Omega_h} + \beta h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial v_h}{\partial n} \right] \right\|^2_{0,F} + \beta h^2 \|\Delta v_h\|^2_{0,\Omega_h^\Gamma}.$$

Lemme 2.2 (cf. [23, Lemme 3.4])**.** *Sous l'hypothèse 2.1.1, toute fonction $v \in H^s(\Omega_h)$ s'annulant sur Ω, avec $1 \leqslant s \leqslant k+1$, vérifie*

$$\|v\|_{0,\Omega_h\setminus\Omega} \leqslant Ch^s \|v\|_{s,\Omega_h\setminus\Omega}.$$

Nous rappelons également un Lemme démontré dans [31, Lemme 4.10] :

Lemme 2.3. *Pour toute fonction $u \in H^1(\Omega_h)$,*

$$\|u\|_{0,\Omega_h^\Gamma} \leqslant C\sqrt{h}\|u\|_{1,\Omega_h}.$$

Enfin, nous introduisons un nouveau résultat :

Lemme 2.4. *Pour tous $u \in V_h^{(k)}$ et $p \in Q_h^{(k)}(\Omega_h^\Gamma)$, il existe $C > 0$ tel que*

$$\|\varphi_h p\|_{0,\Omega_h^\Gamma} \leqslant C(h\|\nabla u\|_{0,\Omega_h^\Gamma} + \|u - \varphi_h p\|_{0,\Omega_h^\Gamma}).$$

Preuve. En utilisant l'inégalité de Poincaré, l'inégalité triangulaire et une inégalité inverse, on obtient :

$$\begin{aligned}\frac{C}{h}\|\varphi_h p\|_{0,\Omega_h^\Gamma} \leqslant \|\nabla\varphi_h p\|_{0,\Omega_h^\Gamma} &\leqslant \|\nabla u\|_{0,\Omega_h^\Gamma} + \|\nabla(u-\varphi_h p)\|_{0,\Omega_h^\Gamma} \\ &\leqslant \|\nabla u\|_{0,\Omega_h^\Gamma} + \frac{C}{h}\|u-\varphi_h p\|_{0,\Omega_h^\Gamma},\end{aligned}$$

ce qui donne le résultat. □

2.1.2 Coercivité de la forme bilinéaire

Lemme 2.5. *Pour γ et σ_D suffisamment grands, la forme bilinéaire donnée par*

$$\begin{aligned}a_h(u,p;v,q) = \int_{\Omega_h}\nabla u\cdot\nabla v - \int_{\partial\Omega_h}\frac{\partial u}{\partial n}v + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(u-\frac{1}{h}\varphi_h p)(v-\frac{1}{h}\varphi_h q) \\ + \sigma_D h\sum_{F\in\mathcal{F}_h^\Gamma}\int_F\left[\frac{\partial u}{\partial n}\right]\left[\frac{\partial v}{\partial n}\right] + \sigma_D h^2\int_{\Omega_h^\Gamma}\Delta u\Delta v\,, \quad (2.3)\end{aligned}$$

est coercive sur $V_h^{(k)}$, selon la norme

$$|||(u,p)|||_h^2 = |u|_{1,\Omega_h}^2 + \frac{1}{h^2}\left\|u-\frac{1}{h}\varphi_h p\right\|_{0,\Omega_h^\Gamma}^2 + h\sum_{F\in\mathcal{F}_h^\Gamma}\left\|\left[\frac{\partial u}{\partial n}\right]\right\|_{0,F}^2 + h^2\|\Delta u\|_{0,\Omega_h^\Gamma}^2. \quad (2.4)$$

Preuve. Soit B_h la bande entre $\partial\Omega_h$ et Γ_h, définie par $B_h = \{\varphi_h > 0\}\cap\Omega_h$. Puisque $\varphi_h = 0$ sur Γ_h, le terme de bord de (2.3) peut être réécrit pour tout u sous la forme :

$$\begin{aligned}\int_{\partial\Omega_h}\frac{\partial u}{\partial n}u = \overbrace{\int_{B_h}|\nabla u|^2}^{I} + \overbrace{\int_{\Gamma_h}\frac{\partial u}{\partial n}(u-\frac{1}{h}\varphi_h p)}^{II} \\ \underbrace{-\sum_{F\in\mathcal{F}_h^\Gamma}\int_{F\cap B_h}\left[\frac{\partial u}{\partial n}\right]u}_{III} + \underbrace{\int_{B_h}\Delta(u)u}_{IV}\,. \quad (2.5)\end{aligned}$$

Avec le Lemme 2.1,

$$I \leqslant \alpha |u|^2_{1,\Omega_h} + \beta h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial u}{\partial n} \right] \right\|^2_{0,F} + \beta h^2 \|\Delta u\|^2_{0,\Omega_h^\Gamma},$$

pour tout $\beta > 0$. De plus, en utilisant l'inégalité de trace suivie d'une inégalité inverse, pour tout $\varepsilon > 0$,

$$\begin{aligned} II &\leqslant C \left(\frac{1}{\sqrt{h}} \|\nabla u\|_{0,\Omega_h^\Gamma} + \sqrt{h} |\nabla u|_{1,\Omega_h^\Gamma} \right) \left(\frac{1}{\sqrt{h}} \left\| u - \frac{1}{h} \varphi_h p \right\|_{0,\Omega_h^\Gamma} + \sqrt{h} \left| u - \frac{1}{h} \varphi_h p \right|_{1,\Omega_h^\Gamma} \right) \\ &\leqslant \frac{C}{h} |u|_{1,\Omega_h^\Gamma} \left\| u - \frac{1}{h} \varphi_h p \right\|_{0,\Omega_h^\Gamma} \\ &\leqslant C\varepsilon |u|^2_{1,\Omega_h^\Gamma} + \frac{C}{\varepsilon h^2} \left\| u - \frac{1}{h} \varphi_h p \right\|^2_{0,\Omega_h^\Gamma}. \end{aligned}$$

Pour le terme III, en utilisant les inégalités de Cauchy-Schwarz, de Young combinée à l'inégalité de trace puis le Lemme 2.4, on a

$$\begin{aligned} III &\leqslant \left(h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial u}{\partial n} \right] \right\|^2_{0,F} \right)^{1/2} \left(\frac{1}{h} \sum_{F \in \mathcal{F}_h^\Gamma} \|u\|^2_{0,F} \right)^{1/2} \\ &\leqslant \frac{C}{\varepsilon} h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial u}{\partial n} \right] \right\|^2_{0,F} + \frac{C\varepsilon}{h} \left(\frac{1}{h} \|u\|^2_{0,\Omega_h^\Gamma} + h |u|^2_{1,\Omega_h^\Gamma} \right) \\ &\leqslant \frac{C}{\varepsilon} h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial u}{\partial n} \right] \right\|^2_{0,F} + C\varepsilon |u|^2_{1,\Omega_h^\Gamma} + \frac{C\varepsilon}{h^2} \left\| u - \frac{1}{h} \varphi_h p \right\|^2_{0,\Omega_h^\Gamma}. \end{aligned}$$

Enfin, pour le terme IV,

$$\begin{aligned} IV &\leqslant \frac{Ch^2}{\varepsilon} \|\Delta u\|^2_{0,\Omega_h^\Gamma} + \frac{C\varepsilon}{h^2} \|u\|^2_{0,\Omega_h^\Gamma} \\ &\leqslant \frac{Ch^2}{\varepsilon} \|\Delta u\|^2_{0,\Omega_h^\Gamma} + C\varepsilon |u|^2_{1,\Omega_h^\Gamma} + \frac{C\varepsilon}{h^2} \left\| u - \frac{1}{h} \varphi_h p \right\|^2_{0,\Omega_h^\Gamma}. \end{aligned}$$

Ainsi,

$$\begin{aligned} \int_{\partial\Omega_h} \frac{\partial u}{\partial n} u \leqslant (\alpha + C\varepsilon) |u|^2_{1,\Omega_h} &+ \left(\frac{C}{\varepsilon} + \beta \right) \left(h \sum_{F \in \mathcal{F}_h^\Gamma} \left\| \left[\frac{\partial u}{\partial n} \right] \right\|^2_{0,F} + h^2 \|\Delta u\|^2_{0,\Omega_h^\Gamma} \right) \\ &+ \left(C\varepsilon + \frac{C}{\varepsilon} \right) \frac{1}{h^2} \left\| u - \frac{1}{h} \varphi_h p \right\|^2_{0,\Omega_h^\Gamma}. \end{aligned}$$

Alors, en utilisant l'expression (2.3),

$$a_h(u,p;u,p) \geqslant (1-\alpha-C\varepsilon)|u|^2_{1,\Omega_h} + \left(\sigma_D - \frac{C}{\varepsilon} - \beta\right)\left(h\sum_{F\in\mathcal{F}_h^\Gamma}\left\|\left[\frac{\partial u}{\partial n}\right]\right\|^2_{0,F} + h^2\|\Delta u\|^2_{0,\Omega_h^\Gamma}\right) + \left(\gamma - C\varepsilon - \frac{C}{\varepsilon}\right)\frac{1}{h^2}\left\|u - \frac{1}{h}\varphi_h p\right\|^2_{0,\Omega_h^\Gamma}.$$

Finalement, en prenant ε suffisamment petit, σ_D et γ assez grands, on obtient

$$a_h(u,p;u,p) \geqslant C|||(u,p)|||_h^2.$$

□

2.1.3 Preuve de l'estimation H^1.

Preuve du Théorème 2.1, estimation H^1. Soit $\tilde{u} \in H^{k+1}(\Omega_h)$ une extension de la solution u de Ω à Ω_h, telle que $\tilde{u} = u$ sur Ω et

$$\|\tilde{u}\|_{k+1,\Omega_h} \leqslant C\|u\|_{k+1,\Omega} \leqslant C\|f\|_{k-1,\Omega}.$$

On considère $\tilde{f} := -\Delta\tilde{u}$ et $p = \frac{h}{\varphi}\tilde{u}$. Alors,

$$a_h(\tilde{u},p;v_h,q_h) = \int_{\Omega_h}\tilde{f}v_h - \sigma_D h^2\int_{\Omega_h^\Gamma}\tilde{f}\Delta v_h + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(\tilde{u} - \frac{1}{h}\varphi_h p)(v_h - \frac{1}{h}\varphi_h q_h)\,,\ \forall (v_h,q_h)\,.$$

Ainsi, on obtient l'orthogonalité de Galerkin suivante

$$a_h(\tilde{u}-u_h,p-p_h;v_h,q_h) = \int_{\Omega_h}(\tilde{f}-f)v_h - \sigma_D h^2\int_{\Omega_h^\Gamma}(\tilde{f}-f)\Delta v_h + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(\tilde{u} - \frac{1}{h}\varphi_h p)(v_h - \frac{1}{h}\varphi_h q_h)\,,\ \forall (v_h,q_h)\,. \quad (2.6)$$

Alors, par coercivité (c.f. Lemme 2.5),

$$\begin{aligned} c|||(u_h - I_h\tilde{u}, p_h - I_h p)|||_h &\leqslant \sup_{(v_h,q_h)}\frac{a_h(u_h - I_h\tilde{u}, p_h - I_h p; v_h, q_h)}{|||(v_h,q_h)|||_h} \\ &\leqslant \sup_{(v_h,q_h)}\frac{I - II - III}{|||(v_h,q_h)|||_h}, \end{aligned}$$

avec

$$\begin{aligned} I &= a_h(e_u, e_p; v_h, q_h)\,, \\ II &= \int_{\Omega_h}(\tilde{f}-f)v_h - \sigma_D h^2\int_{\Omega_h^\Gamma}(\tilde{f}-f)\Delta v_h\,, \\ III &= \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(\tilde{u} - \frac{1}{h}\varphi_h p)(v_h - \frac{1}{h}\varphi_h q_h)\,, \end{aligned}$$

où $e_u = \tilde{u} - I_h\tilde{u}$ et $e_p = p - I_h p$.

Estimons chacun des termes. À l'aide de l'expression (2.5) :

$$\begin{aligned}
I &= \int_{\Omega_h} \nabla e_u \cdot \nabla v_h - \int_{\partial\Omega_h} \frac{\partial e_u}{\partial n} v_h \quad + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma} (e_u - \frac{1}{h}\varphi_h e_p)(v_h - \frac{1}{h}\varphi_h q_h) \\
&+ \sigma_D h \sum_{F\in\mathcal{F}_h^\Gamma} \int_F \left[\frac{\partial e_u}{\partial n}\right]\left[\frac{\partial v_h}{\partial n}\right] + \sigma_D h^2 \sum_{T\in\mathcal{T}_h^\Gamma}\int_T \Delta e_u \Delta v_h \\
&\leqslant |e_u|_{1,\Omega_h}|v_h|_{1,\Omega_h} - \int_{B_h} \nabla e_u \cdot \nabla v_h - \int_{B_h} \Delta e_u v_h + \sum_{F\in\mathcal{F}_h^\Gamma\cap B_h}\int_F e_u \left[\frac{\partial v_h}{\partial n}\right] \\
&- \int_{\Gamma_h} \frac{\partial e_u}{\partial n}(v_h - \frac{1}{h}\varphi_h q_h) + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma} (e_u - \frac{1}{h}\varphi_h e_p)(v_h - \frac{1}{h}\varphi_h q_h) \\
&+ \sigma_D h \sum_{F\in\mathcal{F}_h^\Gamma} \int_F \left[\frac{\partial e_u}{\partial n}\right]\left[\frac{\partial v_h}{\partial n}\right] + \sigma_D h^2 \sum_{T\in\mathcal{T}_h^\Gamma}\int_T \Delta e_u \Delta v_h \,.
\end{aligned}$$

En utilisant l'inégalité de trace [23, Lemme 3.5], les inégalités d'interpolation et l'expression (2.4),

$$I \leqslant Ch^k \|\tilde{u}\|_{k+1,\Omega_h} |||(v_h,q_h)|||_h \leqslant Ch^k \|f\|_{k-1,\Omega} |||(v_h,q_h)|||_h \,.$$

Pour le terme II, puisque $\tilde{f} = f$ sur Ω, et en rappelant que l'on a supposé $\Omega \subset \Omega_h$, on a

$$\begin{aligned}
II &\leqslant C\|\tilde{f} - f\|_{0,\Omega_h\setminus\Omega} \left(\|v_h\|_{0,\Omega_h\setminus\Omega} + \sigma h^2 \|\Delta v_h\|_{0,\Omega_h\setminus\Omega}\right) \\
&\leqslant Ch^{k-1}\|\tilde{f} - f\|_{k-1,\Omega_h\setminus\Omega} \left(h|v_h|_{1,\Omega_h} + \left\|v - \frac{1}{h}\varphi_h q\right\|_{0,\Omega_h^\Gamma} + \sigma h^2 \|\Delta v_h\|_{0,\Omega_h^\Gamma}\right) \\
&\leqslant Ch^k \|f\|_{k-1,\Omega_h} |||(v_h,q_h)|||_h \,.
\end{aligned}$$

Il ne reste finalement plus qu'à estimer le terme III. Alors,

$$III \leqslant \frac{C}{h}\|\tilde{u} - \frac{1}{h}\varphi_h p\|_{0,\Omega_h^\Gamma} |||(v_h,q_h)|||_h \,.$$

Or, puisque $\tilde{u} = \frac{1}{h}p\varphi$ sur Ω_h^Γ,

$$III \leqslant \frac{C}{h}\|\varphi - \varphi_h\|_\infty \left\|\frac{p}{h}\right\|_{0,\Omega_h^\Gamma} |||(v_h,q_h)|||_h \,.$$

En utilisant les inégalités d'interpolation et l'inégalité de Hardy (cf. [28, Lemme 3.1]),

$$\begin{aligned}
III &\leqslant Ch^k \|\varphi\|_{W^\infty_{k+1}} \|\tilde{u}\|_{1,\Omega_h^\Gamma} |||(v_h,q_h)|||_h \\
&\leqslant Ch^k \|f\|_{k-1,\Omega} |||(v_h,q_h)|||_h \,.
\end{aligned}$$

À l'aide des estimations de $(I) - (III)$, par définition de $|||\cdot|||_h$, on obtient ainsi

$$|u_h - I_h\tilde{u}|_{1,\Omega_h} \leqslant |||(u_h - I_h\tilde{u}, p_h - I_h p)|||_h \leqslant Ch^k \|f\|_{k-1,\Omega} \,.$$

Finalement, par inégalité triangulaire et les inégalités d'interpolation,

$$\begin{aligned}|u_h - u|_{1,\Omega} &\leqslant |u_h - I_h\tilde{u}|_{1,\Omega_h} + |I_h\tilde{u} - \tilde{u}|_{1,\Omega_h} \\ &\leqslant Ch^k\|f\|_{k-1,\Omega} + Ch^k\|\tilde{u}\|_{k+1,\Omega_h} \\ &\leqslant Ch^k\|f\|_{k-1,\Omega}\,.\end{aligned}$$

□

2.1.4 Preuve de l'estimation L^2.

Preuve du Théorème 2.1, estimation L^2. Soit $w : \Omega \to \mathbb{R}$, telle que

$$\begin{cases} -\Delta w = u - u_h, & \text{dans } \Omega, \\ w = 0, & \text{sur } \Gamma\,. \end{cases}$$

Par régularité elliptique,

$$\|w\|_{2,\Omega} \leqslant C\|u - u_h\|_{0,\Omega}.$$

Soient $\tilde{w}$ une extension H^2 de w de Ω à Ω_h telle que

$$\|\tilde{w}\|_{2,\Omega} \leqslant C\|w\|_{2,\Omega}$$

et $w_h := I_h\tilde{w}$.

À l'aide d'une intégration par partie, on remarque que

$$\begin{aligned}\|u - u_h\|^2_{0,\Omega} &= \int_\Omega (u - u_h)(-\Delta w) = \int_\Omega \nabla(u - u_h)\cdot\nabla w - \int_\Gamma \frac{\partial w}{\partial n}(u - u_h) \\ &= \int_\Omega \nabla(u - u_h)\cdot\nabla(w - w_h) + \int_\Omega \nabla(u - u_h)\cdot\nabla w_h - \int_\Gamma \frac{\partial w}{\partial n}(u - u_h) \\ &\leqslant Ch^{k+1}\|f\|_{k-1,\Omega_h}|\tilde{w}|_{2,\Omega_h} + \left|\int_\Omega \nabla(u - u_h)\cdot\nabla w_h\right| - \int_\Gamma \frac{\partial w}{\partial n}(u - u_h)\,.\end{aligned}$$

Pour traiter le dernier terme, nous remarquons que

$$-\int_\Gamma \frac{\partial w}{\partial n}(u - u_h) \leq \left\|\frac{\partial w}{\partial n}\right\|_{0,\Gamma}\|u - u_h\|_{0,\Gamma} \leq C\|u - u_h\|_{0,\Gamma}\|u - u_h\|_{0,\Omega}.$$

De plus, comme la distance entre Γ et Γ_h est d'ordre h^{k+1}, on a

$$\begin{aligned}\|u - u_h\|_{0,\Gamma} &\leq C(\|\tilde{u} - u_h\|_{0,\Gamma_h} + h^{(k+1)/2}|\tilde{u} - u_h|_{1,\Omega_h}) \\ &= C(\|\frac{1}{h}(\varphi - \varphi_h)p\|_{0,\Gamma_h} + h^{(k+1)/2}|\tilde{u} - u_h|_{1,\Omega_h}) \\ &= C(h^{k+1}\|\varphi\|_{W^{k+1}_\infty(\Omega^\Gamma_h)}\|\frac{1}{h}p\|_{0,\Gamma_h} + h^{(k+1)/2}|\tilde{u} - u_h|_{1,\Omega_h}) \\ &\leq C(h^{k+1}\|\tilde{u}\|_{2,\Omega_h} + h^{(k+1)/2+k}\|f\|_{k-1,\Omega_h}).\end{aligned}$$

D'où,

$$-\int_\Gamma \frac{\partial w}{\partial n}(u - u_h) \leq Ch^{k+1}\|f\|_{k-1,\Omega_h}\|u - u_h\|_{0,\Omega}.$$

En utilisant les expressions (2.3) et (2.6), avec $v_h = w_h$ et $q_h = 0$, on obtient

$$\begin{aligned}\int_{\Omega_h} \nabla(\tilde{u}-u_h)\cdot\nabla w_h - \int_{\partial\Omega_h} \frac{\partial(\tilde{u}-u_h)}{\partial n} w_h \\ + \frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(u_h - \frac{1}{h}\varphi_h p_h)w_h + \sigma_D h \sum_{F\in\mathcal{F}_h^\Gamma}\int_F \left[\frac{\partial(\tilde{u}-u_h)}{\partial n}\right]\left[\frac{\partial w_h}{\partial n}\right] \\ + \sigma_D h^2 \int_{\Omega_h^\Gamma}\Delta(\tilde{u}-u_h)\Delta w_h = \int_{\Omega_h}(\tilde{f}-f)w_h \\ - \sigma_D h^2 \int_{\Omega_h^\Gamma}(\tilde{f}-f)\Delta w_h\,.\end{aligned}$$

On rappelle que $\tilde{u} = u$ sur Ω, ce qui entraîne

$$\begin{aligned}\|u-u_h\|_{0,\Omega}^2 \leqslant Ch^{k+1}\|f\|_{k-1,\Omega_h}\|u-u_h\|_{0,\Omega} + \overbrace{\left|\int_{\Omega_h\setminus\Omega}\nabla(\tilde{u}-u_h)\cdot\nabla w_h\right|}^{I} \\ + \overbrace{\left|\int_{\partial\Omega_h}\frac{\partial(\tilde{u}-u_h)}{\partial n}w_h\right|}^{II} + \overbrace{\left|\frac{\gamma}{h^2}\int_{\Omega_h^\Gamma}(u_h-\frac{1}{h}\varphi_h p_h)w_h\right|}^{III} \\ + \overbrace{\left|\sigma_D h\sum_{F\in\mathcal{F}_h^\Gamma}\int_F\left[\frac{\partial(\tilde{u}-u_h)}{\partial n}\right]\left[\frac{\partial w_h}{\partial n}\right]\right|}^{IV} + \underbrace{\left|\sigma_D h^2\int_{\Omega_h^\Gamma}\Delta(\tilde{u}-u_h)\Delta w_h\right|}_{V} \\ + \underbrace{\left|\int_{\Omega_h}(\tilde{f}-f)w_h\right|}_{VI} + \underbrace{\left|\sigma_D h^2\int_{\Omega_h^\Gamma}(\tilde{f}-f)\Delta w_h\right|}_{VII}.\end{aligned}$$

Pour le terme I, on utilise l'estimation H^1, une inégalité inverse et le Lemme 2.3 :

$$I \leqslant C|\tilde{u}-u_h|_{1,\Omega_h}|w_h|_{1,\Omega_h\setminus\Omega} \leqslant Ch^k\|u\|_{k+1,\Omega_h}h^{1/2}\|w\|_{2,\Omega}\,.$$

Pour le terme II, en utilisant l'inégalité de trace [28, Lemme 3.5],

$$II \leqslant C\left(\sqrt{h}|\nabla(\tilde{u}-u_h)|_{1,\Omega_h^\Gamma} + \frac{1}{\sqrt{h}}\|\nabla(\tilde{u}-u_h)\|_{0,\Omega_h^\Gamma}\right)\left(\sqrt{h}|w_h|_{1,\Omega_h^\Gamma} + \frac{1}{\sqrt{h}}\|w_h\|_{0,\Omega_h^\Gamma}\right).$$

Or, par inégalité triangulaire, inégalité inverse et inégalité d'interpolation,

$$\begin{aligned}\sqrt{h}|\nabla(\tilde{u}-u_h)|_{1,\Omega_h^\Gamma} &\leqslant \sqrt{h}\left(|\nabla(\tilde{u}-I_h\tilde{u})|_{1,\Omega_h^\Gamma} + |\nabla(I_h\tilde{u}-u_h)|_{1,\Omega_h^\Gamma}\right) \\ &\leqslant \sqrt{h}\left(h^{k-1}|\tilde{u}|_{k+1,\Omega_h} + \frac{1}{h}|I_h\tilde{u}-u_h|_{1,\Omega_h^\Gamma}\right) \\ &\leqslant \sqrt{h}\left(h^{k-1}|\tilde{u}|_{k+1,\Omega_h} + \frac{1}{h}|I_h\tilde{u}-\tilde{u}|_{1,\Omega_h^\Gamma} + \frac{1}{h}|\tilde{u}-u_h|_{1,\Omega_h^\Gamma}\right) \\ &\leqslant h^{k-1/2}|\tilde{u}|_{k+1,\Omega_h}\,.\end{aligned}$$

De plus, par inégalité de Poincaré et le Lemme 2.3

$$\sqrt{h}|w_h|_{1,\Omega_h^\Gamma} + \frac{1}{\sqrt{h}}\|w_h\|_{0,\Omega_h^\Gamma} \leqslant C\sqrt{h}|w_h|_{1,\Omega_h^\Gamma} \leqslant Ch\|w\|_{2,\Omega}\,.$$

Finalement, on obtient ainsi

$$II \leqslant Ch^{k-1/2}\|f\|_{k-1,\Omega}h\|w\|_{2,\Omega} \leqslant Ch^{k+1/2}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega}\,.$$

Pour le terme III, on a

$$III \leqslant \frac{\gamma}{h^2}\left\|u_h - \frac{1}{h}\varphi_h p_h\right\|_{0,\Omega_h^\Gamma}\|w_h\|_{0,\Omega_h^\Gamma}\,.$$

Estimons les deux termes

$$\begin{aligned}
\left\|u_h - \frac{1}{h}\varphi_h p_h\right\|_{0,\Omega_h^\Gamma} &\leqslant \left\|u_h - I_h\tilde{u} - \frac{1}{h}\varphi_h(p_h - I_h p)\right\|_{0,\Omega_h^\Gamma} + \left\|I_h\tilde{u} - \frac{1}{h}\varphi_h I_h p\right\|_{0,\Omega_h^\Gamma} \\
&\leqslant h|||(u_h - I_h\tilde{u}, p_h - I_h p)|||_h + \left\|I_h\tilde{u} - \frac{1}{h}\varphi_h I_h p\right\|_{0,\Omega_h^\Gamma} \\
&\leqslant Ch^{k+1}\|f\|_{k-1,\Omega} + \left\|I_h\tilde{u} - \frac{1}{h}\varphi_h I_h p\right\|_{0,\Omega_h^\Gamma}\,.
\end{aligned}$$

Or, par inégalité triangulaire et de Hardy

$$\begin{aligned}
\left\|I_h\tilde{u} - \frac{1}{h}\varphi_h I_h p\right\|_{0,\Omega_h^\Gamma} &\leqslant \|I_h\tilde{u} - \tilde{u}\|_{0,\Omega_h^\Gamma} + \left\|\frac{1}{h}\varphi p - \frac{1}{h}\varphi_h p\right\|_{0,\Omega_h^\Gamma} + \left\|\frac{1}{h}\varphi_h p - \frac{1}{h}\varphi_h I_h p\right\|_{0,\Omega_h^\Gamma} \\
&\leqslant Ch^{k+1}\|\tilde{u}\|_{k+1,\Omega_h} + Ch^{k+1}\|\varphi\|_{W_\infty^{k+1}(\Omega_h^\Gamma)}\left\|\frac{p}{h}\right\|_{0,\Omega_h^\Gamma} + C\|p - I_h p\|_{0,\Omega_h^\Gamma} \\
&\leqslant Ch^{k+1}\|\tilde{u}\|_{k+1,\Omega_h} + Ch^{k+1}\left\|\frac{p}{h}\right\|_{k,\Omega_h^\Gamma} \\
&\leqslant Ch^{k+1}\|\tilde{u}\|_{k+1,\Omega_h}\,.
\end{aligned}$$

De plus, d'après l'inégalité de Poincaré et le Lemme 2.3,

$$\|w_h\|_{0,\Omega_h^\Gamma} \leqslant Ch\,|w_h|_{1,\Omega_h^\Gamma} \leqslant Ch^{3/2}\,\|w\|_{2,\Omega}\,.$$

D'où,

$$III \leqslant Ch^{k-1}\|\tilde{u}\|_{k+1,\Omega_h}h^{3/2}\|w\|_{2,\Omega} \leqslant Ch^{k+1/2}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega}\,.$$

Pour le terme IV, en utilisant le raisonnement appliqué au terme II, on obtient

$$IV \leqslant Chh^{k-1/2}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega} \leqslant Ch^{k+1/2}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega}\,.$$

Pour le terme V,

$$\begin{aligned}
V &\leqslant \sigma_D h^2\left(|\tilde{u} - I_h\tilde{u}|_{2,\Omega_h^\Gamma} + |I_h\tilde{u} - u_h|_{2,\Omega_h^\Gamma}\right)\|w\|_{2,\Omega} \\
&\leqslant \sigma_D h^2\left(Ch^{k-1}\|\tilde{u}\|_{k+1,\Omega_h^\Gamma} + \frac{C}{h}|I_h\tilde{u} - u_h|_{1,\Omega_h^\Gamma}\right)\|w\|_{2,\Omega} \\
&\leqslant \sigma_D h^2\left(Ch^{k-1}\|\tilde{u}\|_{k+1,\Omega_h^\Gamma} + \frac{C}{h}(|I_h\tilde{u} - \tilde{u}|_{1,\Omega_h^\Gamma} + |\tilde{u} - u_h|_{1,\Omega_h^\Gamma})\right)\|w\|_{2,\Omega} \\
&\leqslant \sigma_D h^{k+1}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega}\,.
\end{aligned}$$

Pour les termes VI et VII, on a de manière similaire,

$$\begin{aligned} VI + VII &\leqslant Ch^{k-1}\|f\|_{k-1,\Omega}h^{3/2}\|w\|_{2,\Omega} + Ch^{k+1}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega} \\ &\leqslant Ch^{k+1/2}\|f\|_{k-1,\Omega}\|w\|_{2,\Omega}\,. \end{aligned}$$

Finalement, en combinant toutes les estimations précédentes, on obtient

$$\|u - u_h\|_{0,\Omega}^2 \leqslant Ch^{k+1/2}\|f\|_{k-1,\Omega_h}\|u - u_h\|_{0,\Omega}\,,$$

ce qui mène à la conclusion. □

2.1.5 Conditionnement

Théorème 2.2. *On suppose que les Hypothèses 2.1.1 et 2.1.2 sont satisfaites et on rappelle que l'on considère un maillage $\mathcal{T}_h$ quasi-uniforme. Alors, le conditionnement de la matrice éléments finis $\mathbf{A}$ associée à la forme bilinéaire a_h vérifie $\kappa(\mathbf{A}) \leqslant Ch^{-2}$ où $\kappa(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2$.*

Démonstration. On rappelle (cf. [23, Equation (17)]) que, pour tout $q_h \in Q_h^{(k)}(\Omega_h^\Gamma)$,

$$\|q_h\|_{0,\Omega_h^\Gamma} \leqslant Ch^{-1}\|\varphi_h q_h\|_{0,\Omega_h^\Gamma}.$$

On suppose, sans perte de généralité, que $h < 1$. On cherche dans un premier temps à démontrer que

$$a_h(v_h, q_h; v_h, q_h) \geqslant C\left(\|v_h\|_{0,\Omega_h}^2 + \|q_h\|_{0,\Omega_h^\Gamma}^2\right)\,. \tag{2.7}$$

Or, en utilisant le Lemme 2.4 et la coercivité de a_h,

$$\begin{aligned} \|q_h\|_{0,\Omega_h^\Gamma} &\leqslant \left\|\frac{1}{h}\varphi_h q_h\right\|_{0,\Omega_h} \leqslant C\left(h\|\nabla v_h\|_{0,\Omega_h^\Gamma} + \left\|v_h - \frac{1}{h}\varphi_h q_h\right\|_{0,\Omega_h^\Gamma}\right) \\ &\leqslant Ch|||(v_h, q_h)|||_h\,. \end{aligned}$$

De plus, en utilisant l'inégalité de Poincaré [11, Equation (1.1)] combinée à l'inégalité de trace

$$\begin{aligned} \|v_h\|_{0,\Omega_h} &\leqslant C\left(|v_h|_{1,\Omega_h} + \|v_h\|_{0,\partial\Omega_h}\right) \\ &\leqslant C\left(|v_h|_{1,\Omega_h} + \frac{1}{\sqrt{h}}\|v_h\|_{0,\Omega_h^\Gamma}\right) \\ &\leqslant C\left(|v_h|_{1,\Omega_h} + \frac{1}{\sqrt{h}}\left\|v_h - \frac{1}{h}\varphi_h q_h\right\|_{0,\Omega_h^\Gamma} + \frac{1}{\sqrt{h}}\left\|\frac{1}{h}\varphi_h q_h\right\|_{0,\Omega_h^\Gamma}\right) \\ &\leqslant C|||(v_h, q_h)|||_h\,. \end{aligned}$$

Finalement, en utilisant la coercivité de a_h,

$$a_h(v_h, q_h; v_h, q_h) \geqslant C|||(v_h, q_h)|||_h^2 \geqslant C\left(\|v_h\|_{0,\Omega_h}^2 + \|q_h\|_{0,\Omega_h^\Gamma}^2\right)\,.$$

Dans un second temps, montrons que

$$a_h(v_h, q_h; v_h, q_h) \leqslant \frac{C}{h^2}\left(\|v_h\|^2_{0,\Omega_h} + \|q_h\|^2_{0,\Omega_h^\Gamma}\right). \tag{2.8}$$

Par définition de a_h (cf. (2.3)) et en utilisant l'inégalité de Cauchy-Schwarz,

$$\begin{aligned} a_h(v_h, q_h; v_h, q_h) \leqslant C\Bigg(|v_h|^2_{1,\Omega_h} + \left\|\frac{\partial v_h}{\partial n}\right\|_{0,\partial\Omega_h}\|v_h\|_{0,\partial\Omega_h} + \frac{1}{h^2}\|v_h\|^2_{0,\Omega_h^\Gamma} \\ + \frac{1}{h^2}\left\|\frac{1}{h}\varphi_h q_h\right\|^2_{0,\Omega_h^\Gamma} + h\left\|\frac{\partial v_h}{\partial n}\right\|^2_{0,\partial\Omega_h} + h^2|v_h|^2_{2,\Omega_h^\Gamma}\Bigg). \end{aligned}$$

Alors, en utilisant l'inégalité de trace, et l'inégalité inverse, on obtient,

$$\begin{aligned} a_h(v_h, q_h; v_h, q_h) \leqslant C\Bigg(\frac{1}{h^2}\|v_h\|^2_{0,\Omega_h} + \frac{1}{h^{3/2}}\frac{1}{\sqrt{h}}\|v_h\|^2_{0,\Omega_h^\Gamma} \\ + \frac{1}{h^2}\|q_h\|^2_{0,\Omega_h^\Gamma} + \frac{1}{h^2}\|v_h\|^2_{0,\Omega_h^\Gamma}\Bigg), \end{aligned}$$

ce qui donne le résultat désiré.

Appelons N la dimension de $V_h^{(k)} \times Q_h^{(k)}(\Omega_h^\Gamma)$ et associons à tout $(v_h, q_h) \in V_h^{(k)} \times Q_h^{(k)}(\Omega_h^\Gamma)$ le vecteur $\mathbf{v} \in \mathbb{R}^N$ des coefficients de (v_h, q_h) dans la base éléments finis standard. Rappelons que le maillage est quasi-uniforme et en utilisant l'équivalence de normes sur un élément de référence, on a

$$C_1 h^{d/2}|\mathbf{v}|_2 \leq \|v_h\|_{0,\Omega_h} + \|q_h\|_{0,\Omega_h^\Gamma} \leq C_2 h^{d/2}|\mathbf{v}|_2. \tag{2.9}$$

Les bornes (2.9) et (2.8) donnent

$$\begin{aligned} \|\mathbf{A}\|_2 = \sup_{\mathbf{v}\in\mathbb{R}^N}\frac{(\mathbf{Av},\mathbf{v})}{|\mathbf{v}|_2^2} &= \sup_{\mathbf{v}\in\mathbb{R}^N}\frac{a_h(v_h, q_h; v_h, q_h)}{|\mathbf{v}|_2^2} \\ &\leq Ch^d \sup_{(v_h,q_h)\in V_h^{(k)}\times Q_h^{(k)}(\Omega_h^\Gamma)}\frac{a_h(v_h, q_h; v_h, q_h)}{\|v_h\|^2_{0,\Omega_h} + \|q_h\|^2_{0,\Omega_h^\Gamma}} \leq Ch^{d-2}. \end{aligned}$$

De la même manière, (2.9) et (2.7) impliquent

$$\begin{aligned} \|\mathbf{A}^{-1}\|_2 = \sup_{\mathbf{v}\in\mathbb{R}^N}\frac{|\mathbf{v}|_2^2}{(\mathbf{Av},\mathbf{v})} &= \sup_{\mathbf{v}\in\mathbb{R}^N}\frac{|\mathbf{v}|_2^2}{a_h(v_h, q_h; v_h, q_h)} \\ &\leq \frac{C}{h^d}\sup_{(v_h,q_h)\in V_h^{(k)}\times Q_h^{(k)}(\Omega_h^\Gamma)}\frac{\|v_h\|^2_{0,\Omega_h} + \|q_h\|^2_{0,\Omega_h^\Gamma}}{a_h(v_h, q_h; v_h, q_h)} \leq \frac{C}{h^d}, \end{aligned}$$

ce qui mène au résultat.

□

2.1.6 Résultats numériques

Nous allons maintenant illustrer sur plusieurs cas test la convergence numérique de cette méthode, comparée au premier schéma φ-FEM introduit dans [28] et rappelé à la Section 1.2, qui sera appelé schéma direct. Nous comparerons également les deux approches à une méthode standard conforme (cf. (1.3)).

Remarque 2.3 (Normes considérées). Pour illustrer la convergence des méthodes, nous considérerons les normes suivantes :

$$\frac{\|u_h - u_{\text{ref}}\|^2_{L^2(\Omega_{\text{ref}})}}{\|u_{\text{ref}}\|^2_{L^2(\Omega_{\text{ref}})}} \approx \frac{\int_{\Omega_{\text{ref}}} |u_h - u_{\text{ref}}|^2 \mathrm{d}x}{\int_{\Omega_{\text{ref}}} |u_{\text{ref}}|^2 \mathrm{d}x}\,, \tag{2.10}$$

et

$$\frac{|u_h - u_{\text{ref}}|^2_{H^1(\Omega_{\text{ref}})}}{|u_{\text{ref}}|^2_{H^1(\Omega_{\text{ref}})}} \approx \frac{\int_{\Omega_{\text{ref}}} |\nabla u_h - \nabla u_{\text{ref}}|^2 \mathrm{d}x}{\int_{\Omega_{\text{ref}}} |\nabla u_{\text{ref}}|^2 \mathrm{d}x}\,, \tag{2.11}$$

où l'on note u_h une approximation de la projection orthogonale L^2 de la solution calculée, sur un maillage de référence $\mathcal{T}_{\text{ref}}$ du domaine Ω_{ref}, approximation de Ω et u_{ref} une solution de référence (manufacturée ou solution fine éléments finis).

Cas test 1 : Conditions non homogènes, sur un disque. Dans un premier temps, considérons l'équation (1.1), avec conditions de Dirichlet non homogènes au bord (i.e. $u = u_D \neq 0$ sur Γ). Le domaine Ω sera le disque centré en $(0.5, 0.5)$ de rayon 0.3125. Pour illustrer l'un des intérêts de l'approche duale par rapport à l'approche directe, le domaine Ω sera décrit par deux fonctions level-set différentes. Dans un premier temps, la version la plus lisse et la plus adaptée à l'approche directe sera utilisée, en définissant la parabole

$$\varphi_1(x, y) = -0.3125^2 + (x - 0.5)^2 + (y - 0.5)^2\,.$$

Dans un second temps, nous utiliserons l'équation correspondant à la distance signée au cercle, c'est-à-dire

$$\varphi_2(x, y) = -0.3125 + \sqrt{(x - 0.5)^2 + (y - 0.5)^2}\,.$$

Les erreurs seront calculées par rapport à une solution très fine FEM standard (calculée avec $h \approx 0.001$). Le second membre est donné par $f(x, y) = -1$ et les conditions de bord sont données par $u_D(x, y) = \cos(\frac{x\pi}{3})\sin(\frac{y\pi}{5})$. On se place enfin dans le cas d'éléments finis $\mathbb{P}^1$ (i.e. $k = 1$).

On représente les résultats obtenus à la Figure 2.1, illustrant que l'ordre de convergence théorique est atteint en norme H^1 et dépassé en norme L^2. Cependant, il est important de distinguer deux cas, puisque les résultats des deux schémas φ-FEM sont comparables lors de l'utilisation de la level-set φ_2 (***traits pleins***), tandis que l'utilisation de la fonction φ_1 (***traits discontinus***) améliore grandement les performances du schéma direct. Cette variation de résultats pour le schéma direct (absente pour le schéma dual, les courbes vertes étant presque superposées) peut notamment s'expliquer par la présence d'une singularité sur le gradient de φ_2 qui n'a pas d'influence sur le schéma dual.

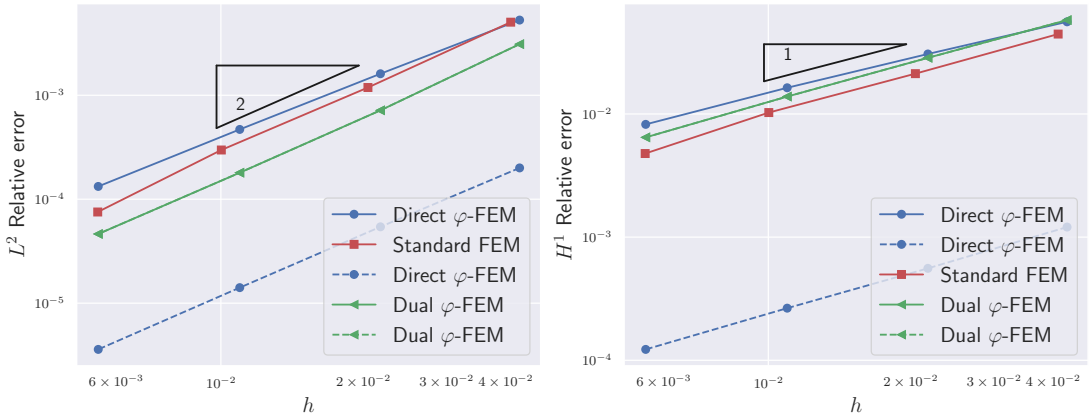

FIGURE 2.1 – **Cas test 1.** Erreurs relatives L^2 (gauche) et H^1 (droite) des trois méthodes, en fonction de la taille de cellule. Pour les méthodes φ-FEM, les pointillés correspondent à φ_1 et les traits pleins à φ_2.

Dans un cas pratique où l'on ne disposerait que d'une distance signée à la frontière, le schéma dual serait ainsi plus adapté.

On représente également à la Figure 2.2 le temps de calcul de chacune des méthodes en fonction de la taille de cellule (gauche) et de l'erreur relative L^2 (droite). Cela permet d'illustrer la différence entre les deux schémas φ-FEM due notamment à l'introduction de la variable auxiliaire p et donc à la résolution d'un système de taille plus élevée pour la version duale. Il est également important de préciser que les implémentations et en particulier les solveurs utilisés diffèrent légèrement de par la nécessité de restreindre les espaces de fonctions pour φ-FEM dual, ce qui est implémenté à l'aide de la librairie *multiphenicsx*[1].

Enfin, un dernier aspect numérique que l'on choisit de vérifier est le conditionnement de la matrice éléments finis associée à chacune des méthodes. On représente à la Figure 2.3 les résultats obtenus par les 3 méthodes, illustrant numériquement que le conditionnement est comme annoncé en théorie (cf. Théorème 2.2) d'ordre 2.

Cas test 2 : Solution manufacturée sur un disque. Il est également intéressant d'étudier le comportement des différentes méthodes lors de l'utilisation d'éléments finis de degré plus élevé. Pour cela, on considère la géométrie précédente, cette fois de rayon $\sqrt{2}/4$ et une solution manufacturée donnée par

$$u_{ex}(x, y) = \sin(R(x, y)) \times \exp(x) \times \sin(y)\,,$$

avec $R(x, y) = -r^2 + (x - 0.5)^2 + (y - 0.5)^2$ et $r = \sqrt{2}/4$. On détermine alors f analytiquement et on impose des conditions homogènes au bord (puisque $u_{ex} = 0$ sur Γ).

Dans un premier temps, pour des éléments finis de degré 1, on compare une nouvelle fois φ-FEM direct et dual avec Standard-FEM. On compare de plus les trois approches à la méthode CutFEM implémentée avec le package Python CutFEMx.

1. `https://multiphenics.github.io/`

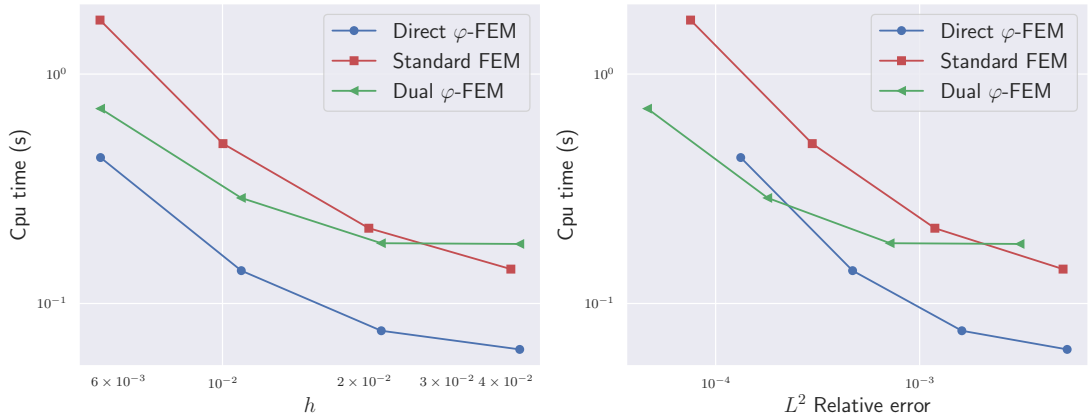

FIGURE 2.2 – **Cas test 1.** Temps de calcul en fonction de la taille de cellule (gauche) et de l'erreur relative L^2 (droite) des trois méthodes.

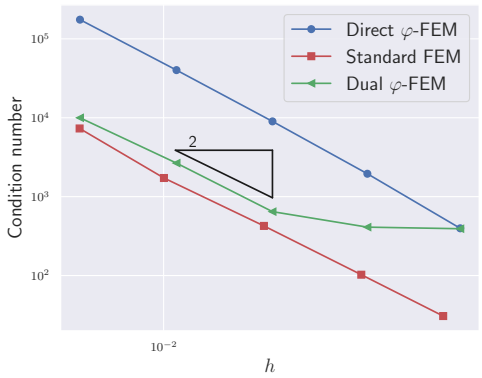

FIGURE 2.3 – **Cas test 1.** Conditionnement des matrices éléments finis associées à chaque méthode, en fonction de h.

Remarque 2.4. Pour ce cas test, dans le cas d'éléments $\mathbb{P}^1$, pour des raisons d'implémentation numérique et une comparaison honnête entre les différentes méthodes et en particulier la méthode CutFEM, l'erreur sera calculée différemment de précédemment et sera donnée par

$$\frac{\left(\frac{1}{N}\sum_{i=1}^{N}(u_{ex}(x_i,y_i)-u_{h,i})^2\right)^{0.5}}{\left(\frac{1}{N}\sum_{i=1}^{N}u_{ex}(x_i,y_i)^2\right)^{0.5}},$$

où (x_i, y_i) sont les coordonnées du nœud i des maillages considérés, et $u_{h,i}$ la solution de chaque méthode au même nœud.

On obtient alors les résultats représentés à la Figure 2.4 qui confirment les ordres de convergence annoncés pour chaque méthode. Les résultats obtenus avec les méthodes CutFEM et la méthode duale φ-FEM sont très proches numériquement, ce qui met en évidence l'intérêt de φ-FEM. Celle-ci permet en effet d'obtenir une précision équivalente

tout en bénéficiant d'une implémentation nettement plus simple, sans utilisation de package spécifique.

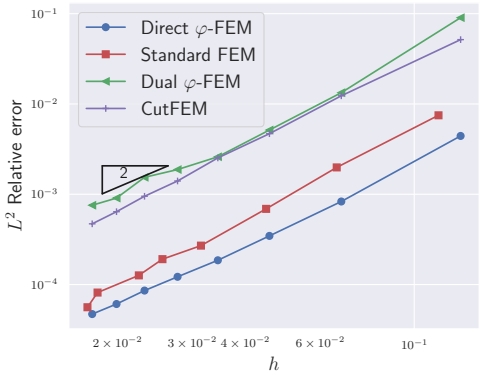

FIGURE 2.4 – **Cas test 2.** Erreurs relatives L^2 des méthodes, en fonction de la taille de cellule.

Pour les éléments finis $\mathbb{P}^2$, les erreurs sont calculées selon (2.10) et (2.11) sur un maillage de référence, avec $u_{\text{ref}} = u_{ex}$. On représente les résultats obtenus à la Figure 2.5, où l'on observe que les ordres de convergence théoriques sont également atteints (et mêmes dépassés en norme L^2) par les deux approches φ-FEM, pour du degré $k = 2$.

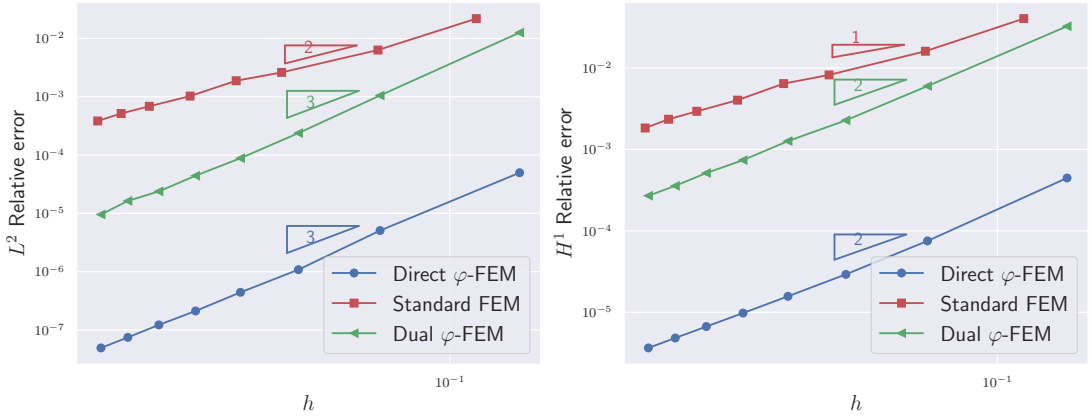

FIGURE 2.5 – **Cas test 2.** Erreurs relatives L^2 (gauche) et H^1 (droite) des trois méthodes, en fonction de la taille de cellule.

Cas test 3 : Une géométrie plus complexe. Pour le troisième cas test, nous allons considérer une situation plus complexe, tant pour les méthodes éléments finis classiques que pour les méthodes non-conformes.
Pour cela, nous considérerons une géométrie définie à partir d'un produit de fonctions gaussiennes, selon l'expression (5.3), présentée en Section 5.1.2.

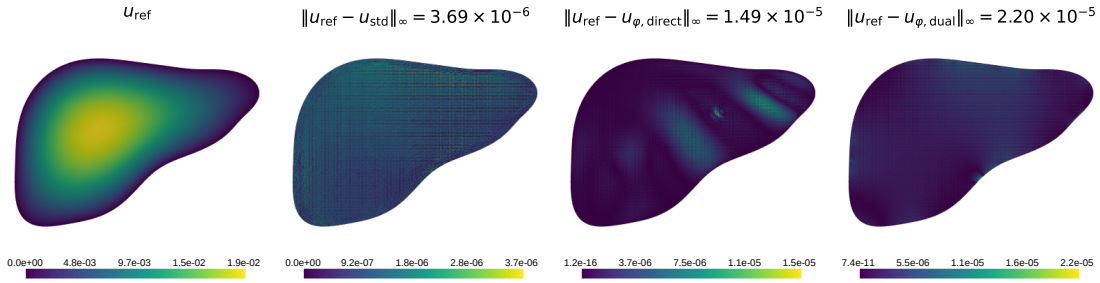

FIGURE 2.6 – **Cas test 3.** À gauche : solution de référence. Puis, de gauche à droite : différence entre la solution de référence et la projection de la solution FEM Standard, de la solution φ-FEM direct, et de la solution φ-FEM duale.

Remarque 2.5 (Construction de maillages sur des géométries complexes). Comme nous l'avons remarqué en introduction, l'une des principales difficultés des méthodes éléments finis classiques est la construction de maillages conformes pour des géométries complexes. Pour construire de tels maillages, notamment à partir de fonctions level-set, nous avons utilisé le package *pymedit*[2] ainsi que Mmg[3]. Plus de détails sur l'approche utilisée et notamment la correction des nœuds de bord sont proposés à la Section 5.1.1.

La solution de référence ainsi que la différence entre les projections sur le maillage de référence des solutions obtenues avec chaque méthode et la solution de référence sont représentées à la Figure 2.6.

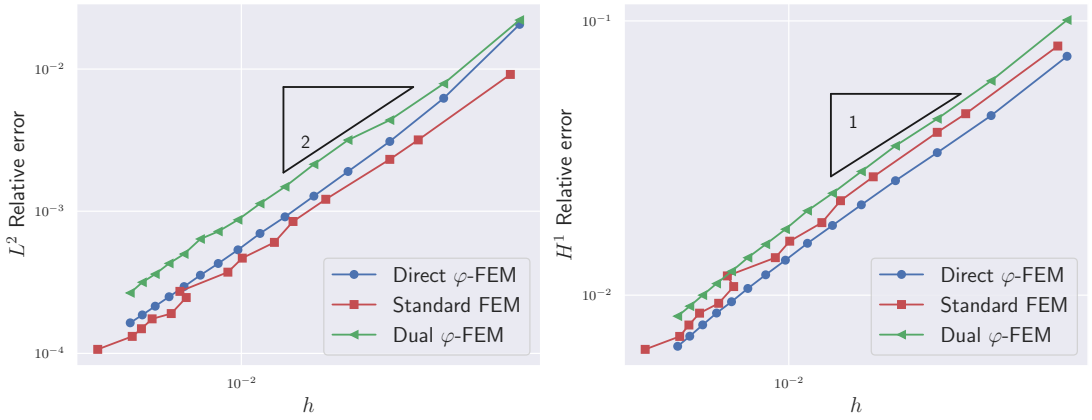

FIGURE 2.7 – **Cas test 3.** Erreurs relatives L^2 (gauche) et H^1 (droite) des trois méthodes, en fonction de la taille de cellule.

Les erreurs des méthodes sont ici calculées par rapport à une solution de référence FEM standard sur un maillage très fin ($h \approx 0.001$). Les résultats obtenus pour les deux schémas φ-FEM et Standard-FEM sont représentés à la Figure 2.7.

2. `https://pypi.org/project/pymedit/`
3. `https://www.mmgtools.org/`

On observe alors des performances très proches pour les trois méthodes à taille de cellule équivalente, chacune suivant les ordres de convergence optimaux en norme L^2 et en norme H^1. On représente également le temps de calcul en fonction de l'erreur relative L^2 à la Figure 2.8 (gauche) et de l'erreur relative H^1 (droite), qui illustrent alors que pour une erreur équivalente, en norme L^2 comme en norme H^1, les résultats sont obtenus bien plus rapidement pour les deux méthodes φ-FEM que pour la méthode standard.

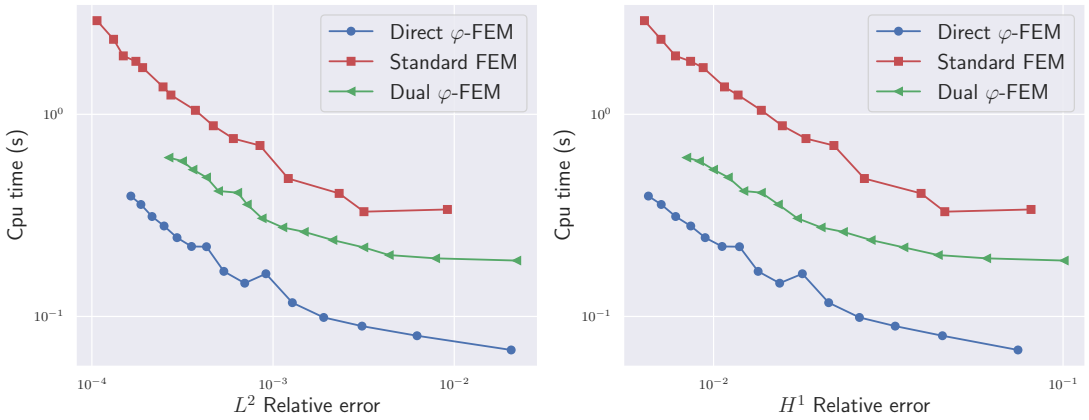

FIGURE 2.8 – **Cas test 3.** Temps de calcul en fonction des erreurs relatives L^2 (gauche) et H^1 (droite) des trois méthodes.

Cas test 4 : un cas 3D. Pour terminer les comparaisons entre les 3 méthodes, nous allons considérer le cas d'une géométrie 3D, donnée par la fonction level-set

$$\varphi(x,y) = -0.3125^2 + (x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2\,,$$

et une solution manufacturée donnée par

$$u_{ex}(x,y) = 1 - \exp(\varphi(x,y)^2)\,,$$

de sorte que $u_{ex} = 0$ sur Γ.

On représente à la Figure 2.9 la solution de référence (i.e. la solution exacte interpolée sur un maillage conforme fin) ainsi que la différence entre les projections sur le maillage de référence des solutions obtenues par chaque méthode et la solution de référence.

Les erreurs en normes relatives L^2 et H^1 sont représentées à la Figure 2.10, où l'on observe que la convergence numérique est une nouvelle fois optimale en norme L^2. On observe une sur-convergence également en norme H^1, puisque les trois méthodes atteignent un ordre 1.5 en norme relative H^1. Cependant, on peut remarquer que dans cette situation, la méthode duale offre des résultats moins précis que les deux autres méthodes.

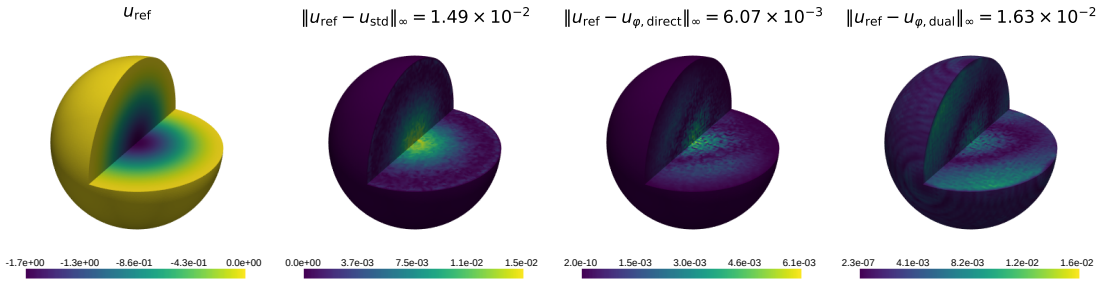

FIGURE 2.9 – **Cas test 4.** À gauche : solution de référence. Puis, de gauche à droite : différence entre la solution de référence et la projection de la solution FEM Standard, de la solution φ-FEM direct, et de la solution φ-FEM dual.

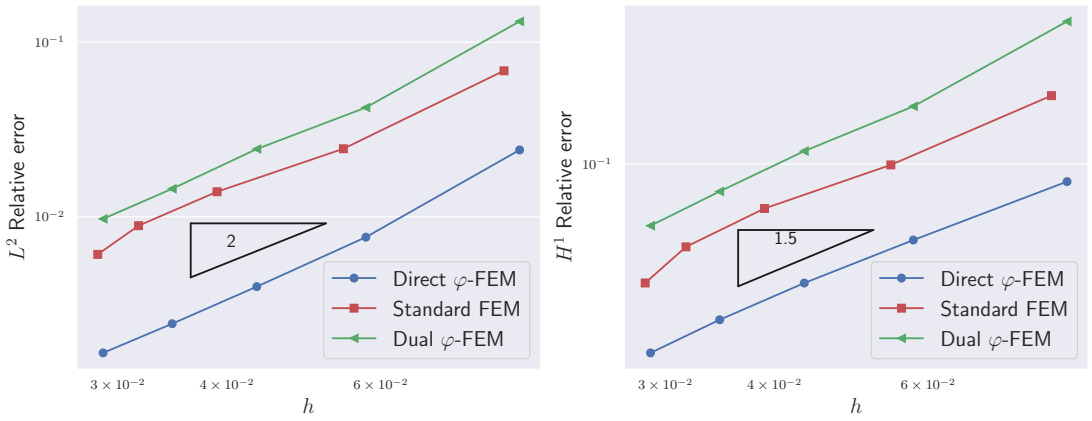

FIGURE 2.10 – **Cas test 4.** Erreurs relatives L^2 (gauche) et H^1 (droite) des trois méthodes, en fonction de la taille de cellule.

2.2 Traitement des conditions mixtes Dirichlet-Neumann

Nous allons maintenant nous intéresser à un cas plus complexe, impliquant un (ou des) changement(s) de conditions de bord. Pour cela, on considère la forme générale de l'équation de Poisson, donnée par

$$\begin{cases} -\Delta u &= f\,, \quad \text{dans } \Omega\,, \\ u &= 0\,, \quad \text{sur } \Gamma_D\,, \\ \nabla u \cdot n &= 0\,, \quad \text{sur } \Gamma_N\,, \end{cases} \tag{2.12}$$

où n est la normale unitaire extérieure à un domaine Ω, de frontière $\Gamma = \Gamma_N \cup \Gamma_D$ avec $\Gamma_N \cap \Gamma_D = \emptyset$ et $f \in L^2(\Omega)$.

Dans cette section, nous allons introduire deux schémas φ-FEM pour résoudre (2.12). Les deux schémas seront étudiés numériquement sur plusieurs cas test, en comparaison avec une méthode éléments finis classique.

Remarque 2.6. Le cas de conditions mixtes Dirichlet-Neumann est particulièrement complexe pour toute méthode non conforme puisque la (ou les) jonction(s) entre la partie Dirichlet et la partie Neumann de la frontière peut (peuvent) intervenir à l'intérieur d'une cellule. Cependant, malgré une complexité plus élevée (autant sur l'aspect théorique que l'aspect numérique), les méthodes non-conformes ont montré leur efficacité, notamment CutFEM, comme dans [17] qui propose des estimations d'erreurs théoriques pour le cas de l'équation (2.12) ou [44] qui propose des résultats numériques pour le cas de l'élasticité linéaire.

Pour le schéma φ-FEM que nous allons construire, nous allons utiliser le schéma dual (2.2) pour imposer les conditions de Dirichlet, et adopter une idée simple : si des cellules contiennent la jonction entre frontière Neumann et frontière Dirichlet, aucune condition de bord ne sera imposée.

En plus de la fonction level-set φ définissant le domaine selon (1.4), on introduit une seconde level-set ψ permettant de séparer la frontière Γ en deux parties Γ_D et Γ_N,

$$\Gamma_D = \Gamma \cap \{\psi < 0\} \quad \text{et} \quad \Gamma_N = \Gamma \cap \{\psi > 0\}\,.$$

On considère une nouvelle fois la boîte $\mathcal{O}$ de $\mathbb{R}^d$ avec $d = 2, 3$ telle que $\Omega \subset \mathcal{O}$. On construit alors les maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ selon (1.6) et (1.7) respectivement. On introduit également les interpolations polynomiales de degré $l \geqslant k$, de φ et ψ sur $\mathcal{T}_h$, notées φ_h et ψ_h.

Les domaines occupés respectivement par $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ sont une nouvelle fois notés Ω_h et Ω_h^Γ. Le maillage $\mathcal{T}_h^\Gamma$ est alors séparé en deux parties, en utilisant la level-set ψ,

$$\mathcal{T}_h^{\Gamma_D} := \{T \in \mathcal{T}_h^\Gamma \,:\, \psi_h \leqslant 0 \text{ sur } T\} \quad \text{et} \quad \mathcal{T}_h^{\Gamma_N} := \{T \in \mathcal{T}_h^\Gamma \,:\, \psi_h \geqslant 0 \text{ sur } T\}\,, \tag{2.13}$$

et on note $\Omega_h^{\Gamma_D}$, $\Omega_h^{\Gamma_N}$ les domaines occupés par $\mathcal{T}_h^{\Gamma_D}$ et $\mathcal{T}_h^{\Gamma_N}$ respectivement.

De plus, il est nécessaire d'ajouter une troisième partie de frontière : en effet, on ne peut pas considérer seulement les cas où la jonction entre Γ_N et Γ_D arrive sur des faces du maillage $\mathcal{T}_h^\Gamma$ (correspondant à la situation illustrée à la Figure 2.12). Il faut aussi considérer que la jonction peut être située à l'intérieur d'une cellule du maillage. Dans cette situation, on choisit de ne pas appliquer de conditions de bord à ces cellules.
On note $\mathcal{T}_h^{\Gamma_{Int}} := \mathcal{T}_h^\Gamma \setminus (\mathcal{T}_h^{\Gamma_D} \cup \mathcal{T}_h^{\Gamma_N})$, et le domaine correspondant à ce sous-maillage est noté $\Omega_h^{\Gamma_{Int}}$ (cf. Figure 2.11, (gauche) pour une représentation graphique des différents sous-maillages). On remarque que

$$\Omega_h^\Gamma = \Omega_h^{\Gamma_D} \cup \Omega_h^{\Gamma_{Int}} \cup \Omega_h^{\Gamma_N}.$$

On note $\Omega_h^i = \Omega_h \backslash \Omega_h^\Gamma$ et $\partial\Omega_h^i$ sa frontière. Soient $\partial\Omega_h^{\Gamma_D}$, $\partial\Omega_h^{\Gamma_N}$ et $\partial\Omega_h^{\Gamma_{Int}}$ les frontières des domaines $\Omega_h^{\Gamma_D}$, $\Omega_h^{\Gamma_N}$ et $\Omega_h^{\Gamma_{Int}}$ intersectées avec $\partial\Omega_h$. On remarque alors que

$$\partial\Omega_h = (\partial\Omega_h^{\Gamma_D} \cap \partial\Omega_h) \cup (\partial\Omega_h^{\Gamma_{Int}} \cap \partial\Omega_h) \cup (\partial\Omega_h^{\Gamma_N} \cap \partial\Omega_h).$$

Soit également l'ensemble de faces $\mathcal{F}_h^\Gamma = \mathcal{F}_h^{\Gamma_D} \cup \mathcal{F}_h^{N_S}$ où

$$\mathcal{F}_h^{\Gamma_D} := \left\{\text{facette de } \mathcal{T}_h^{\Gamma_D} \cup \mathcal{T}_h^{\Gamma_{Int}} \text{ non incluse dans } \partial\Omega_h\right\},$$

et

$$\mathcal{F}_h^{N_S} := \left\{ \text{facette de } \mathcal{T}_h^{\Gamma_N} \text{ incluse dans } \partial\Omega_h^i \right\}.$$

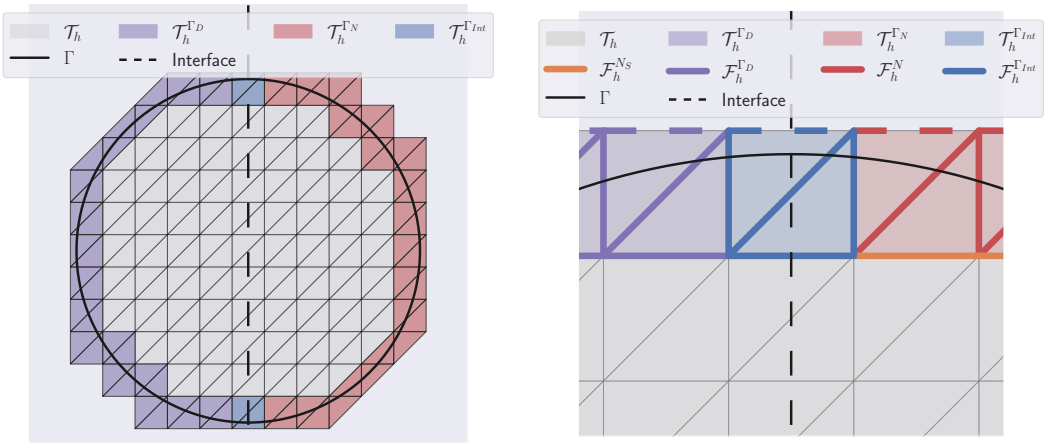

FIGURE 2.11 – Représentation des cellules et faces des différents sous-maillages dans le cas où la jonction entre Γ_D et Γ_N se produit dans une cellule. Sur la figure de droite, les faces en traits pleins correspondent aux faces internes du maillage $\mathcal{T}_h$, celles en pointillés aux faces de bord (i.e. les faces de $\partial\Omega_h$).

Remarque 2.7. Les ensembles $\mathcal{F}_h^{\Gamma_D}$ et $\mathcal{F}_h^{N_S}$ sont les mêmes que ceux introduits dans les précédents schémas φ-FEM, à l'exception qu'ils sont restreints aux sous-maillages correspondant à la partie Dirichlet de la frontière et à la partie Neumann. Les stabilisations imposées dans les différents schémas ne sont pas imposées sur les mêmes faces : pour le schéma Dirichlet (1.10) et (2.2) on considère toutes les faces de Ω_h^Γ tandis que pour Neumann, seulement une partie de ces faces sont considérées. Ainsi, il est important de stabiliser correctement sur chaque portion de la frontière. En particulier, dans la situation de la Figure 2.12, il est important de noter que la facette où la jonction entre Γ_N et Γ_D intervient, est considérée comme n'appartenant ni à $\mathcal{F}_h^{\Gamma_D}$, ni à $\mathcal{F}_h^{N_S}$.

2.2.1 Présentation des schémas

Pour résoudre (2.12), nous proposons 2 méthodes φ-FEM différentes. Le premier schéma suivra l'approche introduite dans [23], rappelée à la Section 1.2 pour l'imposition des conditions de Neumann. Le second introduira lui une nouvelle variante permettant d'imposer les conditions de bord de Neumann.

Premier schéma

Nous allons maintenant construire une combinaison du schéma introduit précédemment pour les conditions de Neumann (voir (1.19)) et du schéma Dual pour les conditions

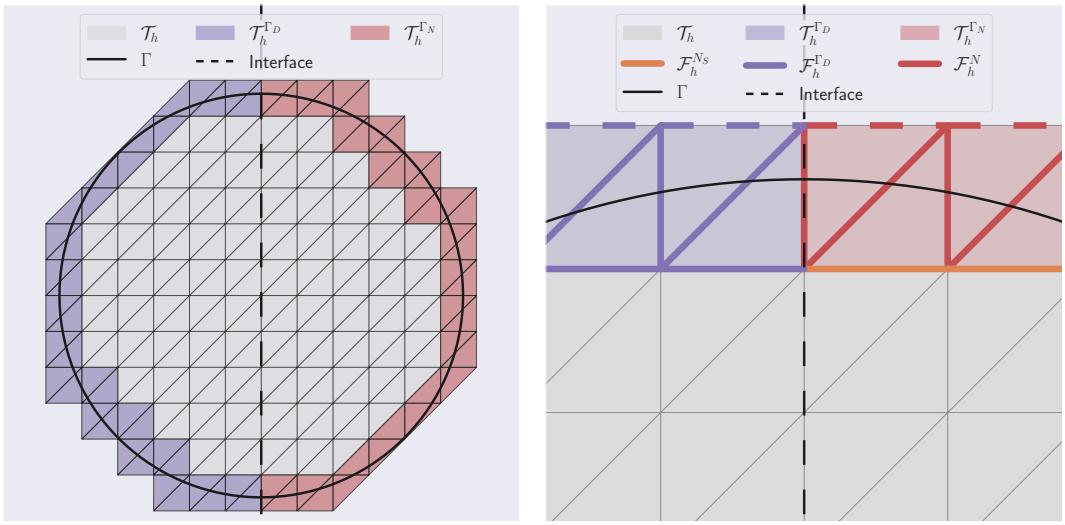

FIGURE 2.12 – Représentation des cellules et faces des différents sous-maillages dans le cas où la jonction entre Γ_D et Γ_N se produit sur une face de $\mathcal{T}_h$.

de Dirichlet (voir (2.2)). Ainsi, on considère u comme l'inconnue primaire sur le domaine Ω_h et on introduit une première variable auxiliaire p_D sur $\Omega_h^{\Gamma_D}$ pour imposer les conditions de Dirichlet, par l'équation

$$u = \varphi p_D \,, \quad \text{sur } \Omega_h^{\Gamma_D} \,.$$

Pour imposer les conditions de Neumann, on introduit comme à la Section 1.2 pour traiter (1.13), une variable auxiliaire y sur $\Omega_h^{\Gamma_N}$, telle que $y = -\nabla u$. En utilisant une nouvelle fois que $n = {}^{\nabla\varphi}/_{|\nabla\varphi|}$ sur Γ, on obtient

$$y \cdot \nabla\varphi = -p_N \varphi \,, \text{ sur } \Omega_h^{\Gamma_N} \,,$$

où p_N est également une variable auxiliaire sur $\Omega_h^{\Gamma_N}$.

On retrouve finalement les trois équations permettant d'imposer les conditions de bord

$$\begin{aligned} u = \varphi p_D \,, &\quad \text{sur } \Omega_h^{\Gamma_D} \,, \\ y + \nabla u = 0 \,, &\quad \text{sur } \Omega_h^{\Gamma_N} \,, \\ y\nabla\varphi + p_N \varphi = 0 \,, &\quad \text{sur } \Omega_h^{\Gamma_N} \,. \end{aligned}$$

Remarque 2.8. On reconnait de manière évidente les équations introduites pour traiter les conditions de bord dans les schémas (1.19) et (2.2). La différence est ici dans les domaines considérés, puisque les variables auxiliaires sont introduites uniquement sur une partie de Ω_h^{Γ}.

Pour discrétiser les différentes variables, on considère alors les espaces éléments finis $V_h^{(k)}$ (cf. (1.9)), $Q_h^{(k)}(\Omega_h^{\Gamma_D})$ (cf. (1.15)), $Z_h^{(k)}(\Omega_h^{\Gamma_N})$ (cf. (1.14)) et $Q_h^{(k-1)}(\Omega_h^{\Gamma_N})$ (cf. (1.15)) et on définit

$$W_h^{(k)} := V_h^{(k)} \times Q_h^{(k)}(\Omega_h^{\Gamma_D}) \times Z_h^{(k)}(\Omega_h^{\Gamma_N}) \times Q_h^{(k-1)}(\Omega_h^{\Gamma_N}) \,.$$

Le schéma φ-FEM pour approcher la solution de (2.12) est finalement donné par : trouver $(u_h, p_{h,D}, y_h, p_{h,N}) \in W_h^{(k)}$ tel que, pour tout $(v_h, q_{h,D}, z_h, q_{h,N}) \in W_h^{(k)}$,

$$\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h \setminus \partial\Omega_{h,N}} \frac{\partial u_h}{\partial n} v_h + a_D(u_h, p_{h,D}; v_h, q_{h,D}) \\ + a_N(u_h, y_h, p_{h,N}; v_h, z_h, q_{h,N}) + G_h(u_h, v_h) = \int_{\Omega_h} f v_h + l_D(v_h) + l_N(z_h)$$

où

$$a_D(u_h, p_{h,D}; v_h, q_{h,D}) = \frac{\gamma}{h^2} \int_{\Omega_h^{\Gamma_D}} (u_h - \frac{1}{h}\varphi_h p_{h,D})(v_h - \frac{1}{h}\varphi_h q_{h,D}) \\ + \sigma_D h^2 \int_{\Omega_h^{\Gamma_D} \cup \Omega_h^{\Gamma_{Int}}} \Delta u_h \Delta v_h,$$

$$a_N(u_h, y_h, p_{h,N}; v_h, z_h, q_{h,N}) = \int_{\partial\Omega_{h,N}} y_h . n v_h + \gamma_u \int_{\Omega_h^{\Gamma_N}} (y_h + \nabla u_h)(z_h + \nabla v_h) \\ + \frac{\gamma_p}{h^2} \int_{\Omega_h^{\Gamma_N}} (y_h \cdot \nabla\varphi_h + \frac{1}{h} p_{h,N}\varphi_h)(z_h \cdot \nabla\varphi_h + \frac{1}{h} q_{h,N}\varphi_h) \\ + \gamma_{div} \int_{\Omega_h^{\Gamma_N}} \operatorname{div} y_h \operatorname{div} z_h,$$

$$G_h(u_h, v_h) := \sigma_D h \sum_{E \in \mathcal{F}_h^{\Gamma_D}} \int_E \left[\frac{\partial u_h}{\partial n}\right] \left[\frac{\partial v_h}{\partial n}\right] + \sigma_N h \sum_{E \in \mathcal{F}_h^{N_S}} \int_E \left[\frac{\partial u_h}{\partial n}\right] \left[\frac{\partial v_h}{\partial n}\right],$$

$$l_D(v_h) = -\sigma_D h^2 \int_{\Omega_h^{\Gamma_D} \cup \Omega_h^{\Gamma_{Int}}} f \Delta v_h \,,$$

et

$$l_N(z_h) = \gamma_{div} \int_{\Omega_h^{\Gamma_N}} f \operatorname{div} z_h.$$

Remarque 2.9 (Conditions non homogènes). Dans le cas de conditions de Dirichlet ou de Neumann non homogènes, on appliquera le même principe que dans les Remarques 2.1 et 1.4, en adaptant les domaines considérés à $\Omega_h^{\Gamma_D}$ et $\Omega_h^{\Gamma_N}$.

Second schéma

Présentons maintenant un second schéma φ-FEM. Ici, les conditions de Dirichlet seront traitées de la même façon, c'est-à-dire via une variable p_D telle que $u = \varphi p_D$ sur $\Omega_h^{\Gamma_D}$. De plus, on définit comme précédemment les maillages $\mathcal{T}_h$, $\mathcal{T}_h^{\Gamma_D}$ et $\mathcal{T}_h^{\Gamma_N}$ ainsi que les domaines Ω_h, $\Omega_h^{\Gamma_D}$ et $\Omega_h^{\Gamma_N}$.
Soient également $\mathcal{F}_h^N$ l'ensemble des facettes de $\mathcal{T}_h^{\Gamma_N}$, ainsi que $\mathcal{F}_h^{N_S}$ et $\mathcal{F}_h^{\Gamma_D}$ définis comme précédemment. Soient p_1 et p_2 définis sur $\Omega_h^{\Gamma_N}$ et considérons

$$\tilde{u}(p_1, p_2) = p_1 + \varphi(g - \nabla p_1 \cdot \nabla\varphi + p_2\varphi) \quad \text{sur } \Omega_h^{\Gamma_N}.$$

On remarque que

$$\frac{\partial \tilde{u}(p_1,p_2)}{\partial n} = g \text{ sur } \Gamma_N.$$

De plus, u peut s'écrire sous la forme $p_1 + \varphi(g - \nabla p_1 \cdot \nabla\varphi + p_2\varphi)$ avec $p_1 = u$ et $p_2 = p$. On cherchera donc u_h sous cette forme sur $\Omega_h^{\Gamma_N}$ par pénalisation.

On introduit alors les espaces éléments finis, comme considérés précédemment : u sera discrétisée dans $V_h^{(k)}$ et p_D dans $Q_h^{(k)}(\Omega_h^{\Gamma_D})$. Finalement, les variables p_1 et p_2 seront elles discrétisées dans $Q_h^{(k+1)}(\Omega_h^{\Gamma_N})$ et $Q_h^{(k)}(\Omega_h^{\Gamma_N})$.
Soit

$$W_h^{(k)} := V_h^{(k)} \times Q_h^{(k)}(\Omega_h^{\Gamma_D}) \times Q_h^{(k+1)}(\Omega_h^{\Gamma_N}) \times Q_h^{(k)}(\Omega_h^{\Gamma_N})\,.$$

Le schéma est alors donné par : trouver $(u_h, p_{h,D}, p_{h,1}, p_{h,2}) \in W_h^{(k)}$ tel que

$$\begin{aligned}
&\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h^N} \nabla\tilde{u}_h \cdot n v_h - \int_{\partial\Omega_h^D \cup \partial\Omega_h^{Int}} \nabla u_h \cdot n v_h + \gamma \frac{1}{h^2} \int_{\Omega_h^{\Gamma_N}} (u_h - \tilde{u}_h)(v_h - \tilde{v}_h) \\
&\qquad + \frac{\sigma_N}{h} \sum_{F \in \mathcal{F}_h^N} \int_F [\nabla\tilde{u}_h \cdot n][\nabla\tilde{v}_h \cdot n] + \gamma \int_{\Omega_h^{\Gamma_N}} (\mathrm{div}(\nabla\tilde{u}_h) + f_h)\mathrm{div}(\nabla\tilde{v}_h) \\
&+ \sigma_N h \sum_{F \in \mathcal{F}_h^{N_s}} \int_F [\nabla u_h \cdot n][\nabla v_h \cdot n] + \frac{\gamma_D}{h^2} \int_{\Omega_h^{\Gamma_D}} (u_h - \frac{1}{h}\varphi_h p_{h,D} - u_D)(v_h - \frac{1}{h}\varphi_h q_{h,D}) \\
&\qquad + \sigma_D h \sum_{F \in \mathcal{F}_h^{\Gamma_D}} \int_F [\nabla u_h \cdot n][\nabla v_h \cdot n] + \gamma_D h^2 \int_{\Omega_h^{\Gamma_D}} (\Delta u_h + f_h)\Delta v_h = \int_{\Omega_h} f_h v_h\,, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall (v_h, q_{h,D}, q_{h,1}, q_{h,2}) \in W_h^{(k)}\,,
\end{aligned}$$

où

$$\tilde{u}_h = p_{h,1} + \varphi_h(g_h - \nabla p_{h,1} \cdot \nabla\varphi_h + p_{h,2}\varphi_h)\,,$$

et

$$\tilde{v}_h = q_{h,1} + \varphi_h(-\nabla q_{h,1} \cdot \nabla\varphi_h + q_{h,2}\varphi_h).$$

Dans la suite de cette section, cette version de φ-FEM sera notée φ-FEM-2.

Remarque 2.10. L'avantage de cette version du schéma est l'absence de la variable vectorielle y. Cependant, en contrepartie, on trouve maintenant une variable p_1 discrétisée dans un espace de degré $k+1$. De plus, ce schéma nécessite plus de termes de stabilisation ainsi qu'un paramètre de stabilisation supplémentaire.

2.2.2 Résultats numériques

Nous allons maintenant étudier numériquement les deux schémas proposés précédemment. Pour cela nous allons considérer différentes situations. Dans un premier temps, le cas le plus simple sans jonction entre les frontières Γ_D et Γ_N sera étudié. Dans ce cas, la solution ne présentera pas de singularité. Dans un second temps, nous considérerons un premier cas présentant deux singularités de changement de conditions de bord, avec

le cas d'un carré tourné. Enfin, nous terminerons cette étude avec le cas d'un disque présentant également deux singularités.
Les trois situations sont représentées à la Figure 2.13.

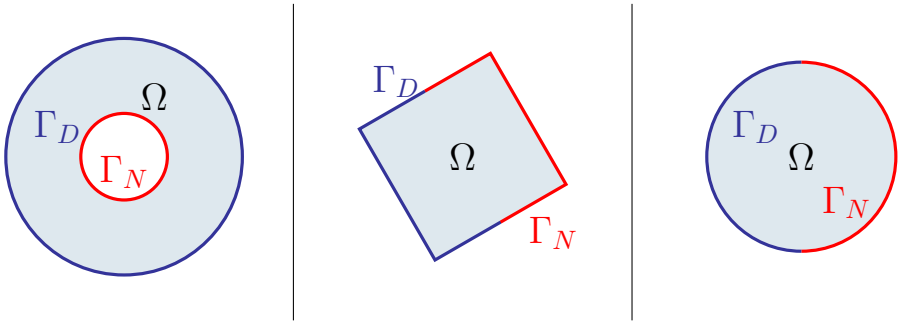

FIGURE 2.13 – Représentation des géométries considérées pour les cas test numériques. Gauche : cas test 1. Centre : cas test 2. Droite : cas test 3.

Les erreurs seront calculées selon les normes relatives L^2 (2.10) et H^1 (2.11), avec une solution de référence éléments finis classique, en utilisant un maillage de référence avec une taille de cellule $h \approx 0.0008$.

Cas test 1 : une solution régulière. Considérons une situation où la solution considérée ne présente pas de singularité, i.e. un cas où $u \in H^2(\Omega)$. Pour cela, on choisit une géométrie sans jonction entre la frontière Dirichlet et la frontière Neumann, représentée à la Figure 2.13 (gauche). Le domaine est donné par la fonction level-set $\varphi(x,y) = \varphi_1(x,y) \times \varphi_2(x,y)$ avec

$$\begin{cases} \varphi_1(x,y) &= -0.391^2 + (x-0.5)^2 + (y-0.5)^2\,, \\ \varphi_2(x,y) &= -0.1431^2 + (x-0.5)^2 + (y-0.5)^2\,. \end{cases}$$

Le terme source de (2.12) est donné par $f = -1$. Enfin, pour détecter le changement de conditions de bord, la fonction level-set ψ est donnée par

$$\psi(x,y) = 0.25^2 - (x-0.5)^2 - (y-0.5)^2\,.$$

Les erreurs en norme L^2 et H^1 sont données à la Figure 2.14. Pour les trois méthodes considérées, on retrouve ici les ordres optimaux de convergence (les ordres attendus sont de 2 pour l'erreur L^2 et 1 pour l'erreur H^1, puisque la solution ne présente pas de singularité), pour les erreurs relatives L^2 et H^1. Les ordres de convergence sont indiqués dans la Table 2.1.

	Optimal	φ-FEM	Std FEM	φ-FEM-2
Erreur L^2	2	2.2	2.04	2.17
Erreur H^1	1	1.31	1.32	1.31

TABLE 2.1 – **Cas test 1.** Ordres de convergence.

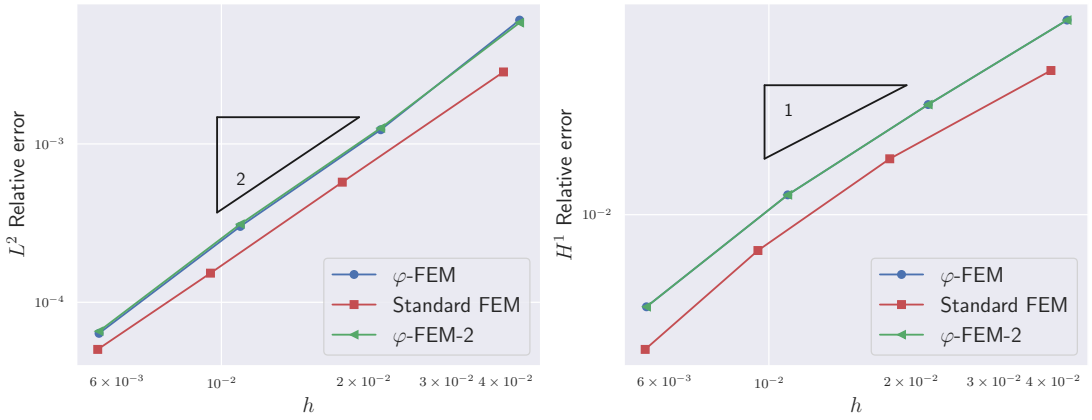

FIGURE 2.14 – **Cas test 1.** Erreurs relatives L^2 (gauche) et H^1 (droite) en l'absence de singularité.

Cas test 2 : singularité sur un carré tourné. Pour le second cas test, la géométrie considérée sera un carré centré au point $(0.5, 0.5)$ de côté 0.5 tourné d'un angle $\pi/6$. La situation considérée est représentée à la Figure 2.13 (centre).
Pour décrire cette géométrie, nous utiliserons une première fonction level-set qui permettra de sélectionner les cellules, définie par

$$\varphi_1(x, y) = \max |R_{(x_0,y_0,\theta)}(x, y) - 0.5| - 0.25\,,$$

où $R_{(x_0,y_0,\theta)}$ est la matrice de rotation centrée en (x_0, y_0), d'angle θ.
Dans les calculs, on choisira une level-set plus lisse, donnée par

$$\varphi_2(x, y) = -((x_R - 0.5) - 0.25) \times ((x_R - 0.5) + 0.25) \\ \times ((y_R - 0.5) - 0.25) \times ((y_R - 0.5) + 0.25)\,,$$

où

$$(x_R, y_R) = R_{(x_0,y_0,\theta)}(x, y)\,.$$

Les résultats des 3 méthodes sont représentés à la Figure 2.15 et les ordres de convergence à la Table 2.2, où l'on remarque que les 3 méthodes convergent de manière optimale en norme L^2 comme en norme H^1. On observe notamment un ordre de convergence plus élevé pour les deux schémas φ-FEM que pour la méthode standard, en particulier en norme L^2.

	Optimal	φ-FEM	Std FEM	φ-FEM-2
Erreur L^2	1	1.19	1.09	1.27
Erreur H^1	0.5	0.56	0.56	0.56

TABLE 2.2 – **Cas test 2.** Ordres de convergence des méthodes.

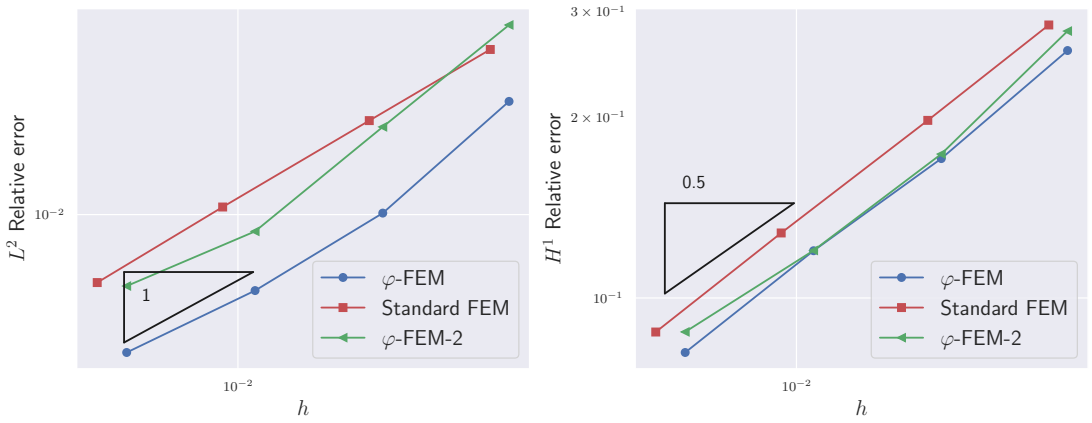

FIGURE 2.15 – **Cas test 2.** Erreurs relatives L^2 et H^1 en fonction de h.

Cas test 3 : singularités sur un disque. Nous allons maintenant considérer le cas d'un disque centré en $(0.5, 0.5)$ de rayon 0.3125, avec une frontière Γ divisée en Γ_D et Γ_N comme représenté à la Figure 2.13 (droite), à l'aide de la level-set $\psi(x, y) = x - 0.5$.

Dans les résultats qui suivent, nous allons distinguer deux cas : le premier cas sera obtenu lorsque l'interface entre la partie Neumann et la partie Dirichlet se produit sur un nœud du maillage standard (analogue à la situation où elle se produit sur une face du maillage φ-FEM). Cette situation correspondra à la dénomination *matching*. Le second cas, moins artificiel sera le cas où cette jonction se produit sur une face du maillage standard (considéré analogue à la situation où la jonction se fait à l'intérieur d'une cellule du maillage φ-FEM), que l'on appellera *not matching*.

Les résultats obtenus dans le cas *matching* sont représentés à la Figure 2.16 ; dans la situation *not matching*, à la Figure 2.17. Les ordres de convergence sont indiqués dans la Table 2.3 pour les deux situations. Dans les deux situations, on observe sur les résultats que les trois méthodes vérifient numériquement les ordres optimaux de convergence : les erreurs L^2 sont d'ordre h et les erreurs H^1 d'ordre $h^{1/2}$. En particulier, on observe que les deux schémas φ-FEM donnent de meilleurs résultats en norme L^2 que la méthode standard. En ce qui concerne la norme H^1, il est intéressant de noter que les trois méthodes donnent des résultats très comparables.

Matching	Optimal	φ-FEM	Standard FEM	φ-FEM-2
Erreur L^2	1	0.98	1.08	1.06
Erreur H^1	0.5	0.5	0.55	0.5
Not Matching	Optimal	φ-FEM	Standard FEM	φ-FEM-2
Erreur L^2	1	1.19	0.98	1.02
Erreur H^1	0.5	0.49	0.51	0.49

TABLE 2.3 – **Cas test 3.** Ordres de convergence des méthodes, dans les deux situations considérées.

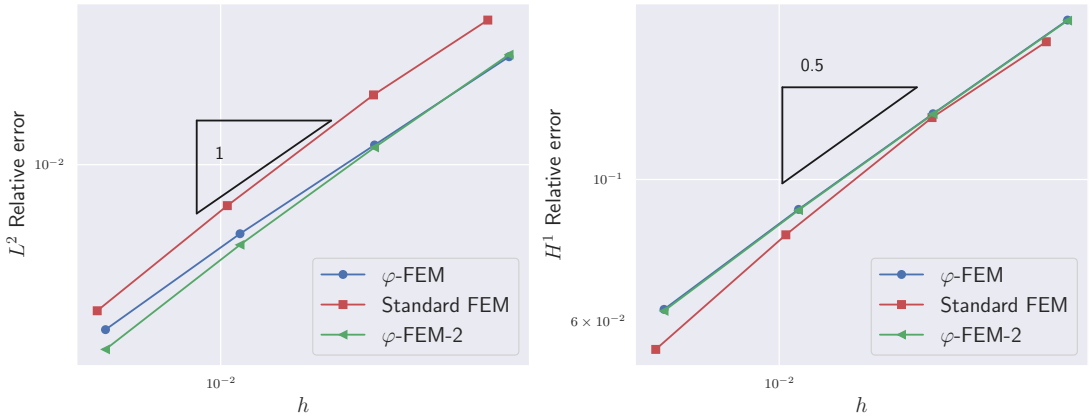

FIGURE 2.16 – **Cas test 3.** Erreur relative L^2 (gauche) et erreur relative H^1 (droite) en fonction de h, pour une interface *matching*.

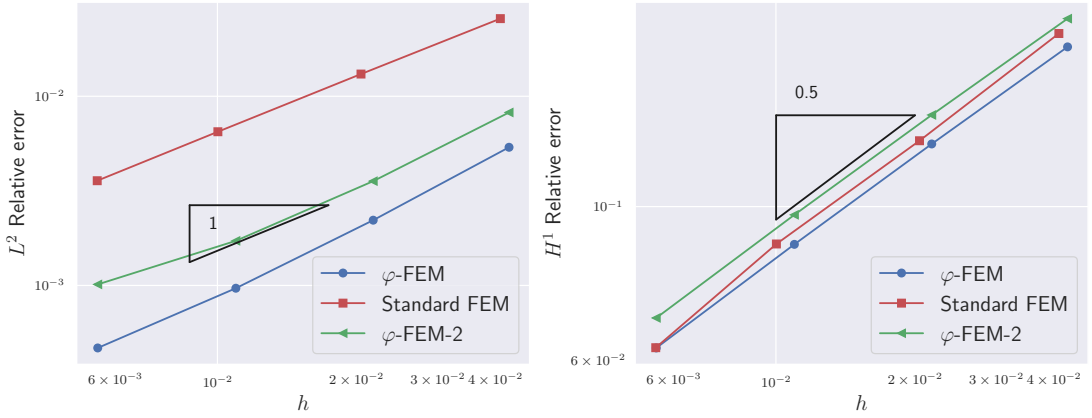

FIGURE 2.17 – **Cas test 3.** Erreur relative L^2 (gauche) et erreur relative H^1 (droite) en fonction de h, pour une interface *not matching*.

2.3 φ-FEM pour l'équation de la chaleur

Nous allons maintenant considérer une équation parabolique dépendant du temps, l'équation de la chaleur avec des conditions de Dirichlet au bord, donnée par

$$\begin{cases} \partial_t u - \Delta u &= f\,, \text{ dans } \Omega \times (0,T)\,, \\ u &= 0\,, \quad \text{sur } \Gamma \times (0,T)\,, \\ u_{|t=0} &= u^0\,, \qquad \text{sur } \Omega, \end{cases} \tag{2.14}$$

avec $T > 0$.

La première partie de cette section, sera consacrée à la présentation d'un schéma φ-FEM pour la résolution de cette équation. Dans la seconde partie, nous proposerons l'analyse théorique de ce schéma. Nous énoncerons alors des estimations d'erreur *a priori*

pour les normes $l^2(H^1)$ et $l^\infty(L^2)$. La troisième partie sera finalement consacrée à l'étude numérique de ce schéma.
Les résultats présentés dans cette section ont été introduits dans [22, 27].

2.3.1 Construction du schéma

Soient $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ définis par (1.6) et (1.7) respectivement. Soit également $\mathcal{F}_h^\Gamma$ donné par (1.8). Soit un temps final $T > 0$. On introduit une partition uniforme de l'intervalle $[0, T]$ en temps t_n, $n = 0, \ldots, N$ tels que $t_n = n\Delta t$ et $t_N = T$.

Pour construire le schéma φ-FEM, nous allons suivre l'idée présentée à la Section 1.2 pour le cas de l'équation (1.1). Cependant, cette fois, nous introduirons une nouvelle inconnue $w = w(x, t)$, au lieu de seulement $w = w(x)$. Ainsi, nous pourrons poser $u = \varphi w$ de sorte que les conditions de Dirichlet $u = 0$ soient automatiquement satisfaites au bord.

La discrétisation en temps de (2.14) sera faite en utilisant un schéma d'Euler implicite. Les évaluations aux temps t_n des fonctions seront notées $f^n(\cdot) = f(\cdot, t_n)$. Ainsi, cela nous permet d'obtenir la discrétisation en temps suivante : pour $u^n = \varphi w^n$ donné, trouver $u^{n+1} = \varphi w^{n+1}$ qui vérifie

$$\frac{\varphi w^{n+1} - \varphi w^n}{\Delta t} - \Delta(\varphi w^{n+1}) = f^{n+1}\,. \tag{2.15}$$

Pour la discrétisation en espace, on considère l'espace éléments finis de degré k, $V_h^{(k)}$ (défini par (1.9)), pour $k \geqslant 1$.

On suppose que les fonctions f et u^0 sont définies sur Ω_h. On rappelle que φ_h est l'interpolation de φ dans $V_h^{(l)}$, pour $l \geqslant k$. Le schéma φ-FEM pour résoudre (2.14) est alors : trouver $w_h^{n+1} \in V_h^{(k)}$, $n = 0, 1, \ldots, N-1$ tel que pour tout $v_h \in V_h^{(k)}$

$$\begin{aligned}
&\int_{\Omega_h} \frac{\varphi_h w_h^{n+1}}{\Delta t}\varphi_h v_h + \int_{\Omega_h} \nabla(\varphi_h w_h^{n+1}) \cdot \nabla(\varphi_h v_h) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\varphi_h w_h^{n+1})\varphi_h v_h \\
&+\sigma_D h \sum_{E \in \mathcal{F}_h^\Gamma} \int_E \left[\frac{\partial(\varphi_h w_h^{n+1})}{\partial n}\right]\left[\frac{\partial(\varphi_h v_h)}{\partial n}\right] - \sigma_D h^2 \sum_{K \in \mathcal{T}_h^\Gamma} \int_K \left(\frac{\varphi_h w_h^{n+1}}{\Delta t} - \Delta(\varphi_h w_h^{n+1})\right)\Delta(\varphi_h v_h) \\
&\qquad = \int_{\Omega_h} \left(\frac{u_h^n}{\Delta t} + f^{n+1}\right)\varphi_h v_h - \sigma_D h^2 \sum_{K \in \mathcal{T}_h^\Gamma} \int_K \left(\frac{u_h^n}{\Delta t} + f^{n+1}\right)\Delta(\varphi_h v_h)\,,
\end{aligned} \tag{2.16}$$

où $u_h^n = \varphi_h w_h^n$ pour $n \geqslant 1$ et $u_h^0 \in V_h^{(k)}$ est l'interpolation de u^0.

Dans (2.16), on retrouve les termes de stabilisation introduits précédemment : la pénalisation fantôme (la somme sur les facettes de $\mathcal{F}_h^\Gamma$) comme introduite dans [12], et la stabilisation d'ordre 2 (les termes multipliés par $\sigma_D h^2$) qui renforce (2.15) sur les cellules de $\mathcal{T}_h^\Gamma$.

Remarque 2.11. Le schéma peut être adapté sans difficulté au cas de conditions de Dirichlet non homogènes $u = u_D$ sur $\Gamma \times (0, T)$. Il suffit alors de considérer $u_h^n = \varphi_h w_h^n + I_h u_g(\cdot, t_n)$ avec u_g un prolongement de u_D de Γ à Ω_h et I_h un interpolant sur $V_h^{(k)}$, comme pour le schéma (1.10). Effectuer les modifications appropriées au schéma (2.16) (i.e. remplacer

$\varphi_h w_h^{n+1}$ par $\varphi_h w_h^{n+1} + I_h u_g(\cdot, t_{n+1})$) introduit alors des termes supplémentaires qui sont tous ajoutés au second membre.

2.3.2 Analyse théorique

Commençons par énoncer le théorème de convergence :

Théorème 2.3 (cf. [27, Théorème 1]). *Supposons que $\Omega \subset \Omega_h$, $l \geq k$, $f \in H^1(0,T;H^{k-1}(\Omega_h))$ et $u \in H^2(0,T;H^{k-1}(\Omega))$ est la solution exacte* (2.14). *De plus, on suppose que $u^n(\cdot) = u(\cdot,t_n)$ et w_h^n sont les solutions de* (2.16) *pour $n = 1,\dots,N$. Enfin, on suppose que les Hypothèses 2.1.1-2.1.2, sont vérifiées. Alors, pour σ_D suffisamment grand, il existe $c > 0$ dépendant seulement de la régularité de $\mathcal{T}_h$ et des constantes des Hypothèses 2.1.1-2.1.2 et $C > 0$ dépendant en plus de T, telles que si $\Delta t \geqslant ch^2$ alors*

$$\left(\sum_{n=0}^{N} \Delta t |u^n - \varphi_h w_h^n|^2_{H^1(\Omega)}\right)^{\frac{1}{2}} \leqslant C\|u^0 - u_h^0\|_{L^2(\Omega_h)} + C(h^k + \Delta t)\left(\|u\|_{H^2(0,T;H^{k-1}(\Omega))} + \|f\|_{H^1(0,T;H^{k-1}(\Omega_h))}\right)$$

et

$$\max_{1\leqslant n\leqslant N} \|u^n - \varphi_h w_h^n\|_{L^2(\Omega)} \leqslant C\|u^0 - u_h^0\|_{L^2(\Omega_h)} + C(h^{k+\frac{1}{2}} + \Delta t)\left(\|u\|_{H^2(0,T;H^{k-1}(\Omega))} + \|f\|_{H^1(0,T;H^{k-1}(\Omega_h))}\right).$$

Remarque 2.12. Si $k = 1$, les normes majorantes des estimations précédentes peuvent être remplacées par la norme de f sur $H^1(0,T;L^2(\Omega_h))$. En effet, puisque $\Omega \subset \Omega_h$, l'hypothèse sur f implique que $u \in H^2(0,T;L^2(\Omega)) \cap H^1(0,T;H^2(\Omega))$, c.f [33, Théorèmes 5 et 6, Chapitre 7.1]. Cependant, imposer cette régularité de u sur Ω ne suffit pas à contrôler l'extension de f sur $\Omega_h \setminus \Omega$, ainsi il est nécessaire d'imposer la régularité sur Ω_h, à la différence des estimations *a priori* classiques des méthodes éléments finis standards (c.f. par exemple [85]).

Avant de démontrer le Théorème 2.3, il est nécessaire de rappeler plusieurs résultats de [28] pour résoudre (1.1).

Lemme 2.6 (cf. [28, Lemme 3.7]). *On considère la forme bilinéaire*

$$a_h(u,v) = \int_{\Omega_h} \nabla u \cdot \nabla v - \int_{\partial\Omega_h} \frac{\partial u}{\partial n} v + \sigma_D h \sum_{E\in\mathcal{F}_h^\Gamma} \int_E \left[\frac{\partial u}{\partial n}\right]\left[\frac{\partial v}{\partial n}\right] + \sum_{K\in\mathcal{T}_h^\Gamma} \sigma_D h^2 \int_K \Delta u\, \Delta v.$$

Pour σ_D assez grand, il existe une constante $\alpha > 0$ indépendante de h telle que

$$a_h(\varphi_h v_h, \varphi_h v_h) \geqslant \alpha |\varphi_h v_h|^2_{H^1(\Omega_h)}, \quad \forall v_h \in V_h^{(k)}.$$

Lemme 2.7 (cf. [28, Théorème 2.3]). *Pour toute fonction $f \in H^{k-1}(\Omega_h)$, soit $w_h \in V_h^{(k)}$ la solution de*

$$a_h(\varphi_h w_h, \varphi_h v_h) = \int_{\Omega_h} f\varphi_h v_h - \sigma_D h^2 \sum_{K\in\mathcal{T}_h^\Gamma} \int_K f\Delta(\varphi_h v_h)$$

et soit $u \in H^{k+1}(\Omega)$ la solution de

$$-\Delta u = f \quad \textit{dans } \Omega, \quad u = 0 \quad \textit{sur } \Gamma$$

étendue à $\tilde{u} \in H^{k+1}(\Omega_h)$ telle que $u = \tilde{u}$ sur Ω et

$$\|\tilde{u}\|_{H^{k+1}(\Omega_h)} \leqslant C\|u\|_{H^{k+1}(\Omega)} \leqslant C\|f\|_{H^{k-1}(\Omega_h)}.$$

Pour σ_D assez grand, il existe une constante $C > 0$ indépendante de h telle que

$$|\tilde{u} - \varphi_h w_h|_{H^1(\Omega_h)} \leqslant Ch^k\|f\|_{H^{k-1}(\Omega_h)} \quad \text{et} \quad \|\tilde{u} - \varphi_h w_h\|_{L^2(\Omega_h)} \leqslant Ch^{k+\frac{1}{2}}\|f\|_{H^{k-1}(\Omega_h)}.$$

Il est également nécessaire d'introduire le résultat suivant :

Lemme 2.8. *Pour tout $v_h \in V_h^{(k)}$, il existe une constante $C_P > 0$ telle que*

$$\|\varphi_h v_h\|_{L^2(\Omega_h)} \leqslant C_P|\varphi_h v_h|_{H^1(\Omega_h)}.$$

Preuve. Soit $\tilde{\Omega}_h = \{\varphi_h < 0\}$. En utilisant l'inégalité de Poincaré,

$$\|\varphi_h v_h\|_{L^2(\tilde{\Omega}_h)} \leqslant C\mathrm{diam}(\tilde{\Omega}_h)|\varphi_h v_h|_{H^1(\tilde{\Omega}_h)},$$

et $\mathrm{diam}(\tilde{\Omega}_h) \leqslant \mathrm{diam}(\mathcal{O})$.
De plus, par [28, Lemme 3.4],

$$\|\varphi_h v_h\|_{L^2(\Omega_h\setminus\tilde{\Omega}_h)} \leqslant \|\varphi_h v_h\|_{L^2(\Omega_h^\Gamma)} \leqslant Ch|\varphi_h v_h|_{H^1(\Omega_h^\Gamma)},$$

où Ω_h^Γ est le domaine occupé par $\mathcal{T}_h^\Gamma$ (définis par (1.7)).
En notant $\Omega \subset \tilde{\Omega}_h \cup \Omega_h^\Gamma$, on obtient le résultat désiré. □

Preuve du Théorème 2.3. Il existe une extension $\tilde{u} \in H^2(0,T;H^{k-1}(\Omega_h))$, de u à Ω_h, telle que

$$\|\tilde{u}\|_{H^2(0,T;H^{k-1}(\Omega_h))} \leqslant C\|u\|_{H^2(0,T;H^{k-1}(\Omega))}. \tag{2.17}$$

Soit w_h^n la solution obtenue par le schéma φ-FEM (2.16), qui peut être réécrit sous la forme

$$\int_{\Omega_h} \varphi_h \frac{w_h^{n+1} - w_h^n}{\Delta t}\varphi_h v_h + a_h(\varphi_h w_h^{n+1}, \varphi_h v_h) - \sum_{K\in\mathcal{T}_h^\Gamma} \sigma_D h^2 \int_K \varphi_h \frac{w_h^{n+1} - w_h^n}{\Delta t}\Delta(\varphi_h v_h)$$
$$= \int_{\Omega_h} f^{n+1}\varphi_h v_h - \sum_{K\in\mathcal{T}_h^\Gamma} \sigma_D h^2 \int_K f^{n+1}\Delta(\varphi_h v_h) \tag{2.18}$$

pour $n \geqslant 1$ où $\varphi_h w_h^0$ sera remplacé par u_h^0 pour $n = 0$.

À chaque temps $t \in [0, T]$, on introduit $\tilde{w}_h(\cdot, t) = \tilde{w}_h \in V_h^{(k)}$, comme dans le Lemme 2.7, où f est remplacé par $f - \partial_t \tilde{u}$ évalué à chaque temps t :

$$a_h(\varphi_h \tilde{w}_h, \varphi_h v_h) = \int_{\Omega_h} (f - \partial_t \tilde{u}) \varphi_h v_h - \sigma_D h^2 \sum_{K \in \mathcal{T}_h^\Gamma} \int_K (f - \partial_t \tilde{u}) \Delta(\varphi_h v_h). \tag{2.19}$$

Soient $\tilde{w}_h^n = \tilde{w}_h(t_n)$ et $e_h^n := \varphi_h(w_h^n - \tilde{w}_h^n)$ pour $n \geqslant 1$ avec $e_h^0 := u_h^0 - \varphi_h \tilde{w}_h^0$.
On considère la différence entre (2.18) et (2.19) au temps t_{n+1}, et on obtient

$$\int_{\Omega_h} \frac{e_h^{n+1} - e_h^n}{\Delta t} \varphi_h v_h + a_h(e_h^{n+1}, \varphi_h v_h) - \sum_{K \in \mathcal{T}_h^\Gamma} \sigma_D h^2 \int_K \frac{e_h^{n+1} - e_h^n}{\Delta t} \Delta(\varphi_h v_h)$$

$$= \int_{\Omega_h} \left(\partial_t \tilde{u}^{n+1} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t} \right) \varphi_h v_h$$
$$- \sum_{K \in \mathcal{T}_h^\Gamma} \sigma_D h^2 \int_K \left(\partial_t \tilde{u}^{n+1} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t} \right) \Delta(\varphi_h v_h).$$

En prenant $v_h = w_h^{n+1} - \tilde{w}_h^{n+1}$, i.e. $\varphi_h v_h = e_h^{n+1}$, et en combinant l'égalité

$$\|e_h^{n+1}\|_{L^2(\Omega_h)}^2 - (e_h^n, e_h^{n+1})_{L^2(\Omega_h)} = \frac{\|e_h^{n+1}\|_{L^2(\Omega_h)}^2 - \|e_h^n\|_{L^2(\Omega_h)}^2 + \|e_h^{n+1} - e_h^n\|_{L^2(\Omega_h)}^2}{2},$$

et les estimations des termes du second membre (avec les inégalités de Cauchy-Schwarz et inverse : $\|\Delta e_h^{n+1}\|_{L^2(T)} \leqslant C h^{-2} \|e_h^{n+1}\|_{L^2(T)}$), on obtient

$$\frac{\|e_h^{n+1}\|_{L^2(\Omega_h)}^2 - \|e_h^n\|_{L^2(\Omega_h)}^2 + \|e_h^{n+1} - e_h^n\|_{L^2(\Omega_h)}^2}{2\Delta t} + \overbrace{a_h(e_h^{n+1}, e_h^{n+1})}^{(I)}$$
$$\overbrace{- \sigma_D h^2 \int_{\Omega_h^\Gamma} \frac{e_h^{n+1} - e_h^n}{\Delta t} \Delta e_h^{n+1}}^{(II)} \leqslant \underbrace{C \left\| \partial_t \tilde{u}^{n+1} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t} \right\|_{L^2(\Omega_h)}}_{(III)} \|e_h^{n+1}\|_{L^2(\Omega_h)}. \tag{2.20}$$

D'après le lemme de coercivité 2.6, on peut minorer (I) par $\alpha |e_h^{n+1}|_{H^1(\Omega_h)}^2$. En utilisant l'inégalité de Young (pour $\varepsilon > 0$) et l'inégalité inverse $\|\Delta e_h^{n+1}\|_{L^2(T)} \leqslant C_I h^{-1} |e_h^{n+1}|_{H^1(T)}$,

$$(I) - (II) \geqslant \alpha |e_h^{n+1}|_{H^1(\Omega_h)}^2 - \frac{\sigma_D h^2}{2\epsilon(\Delta t)^2} \|e_h^{n+1} - e_h^n\|_{L^2(\Omega_h^\Gamma)}^2 - \frac{\epsilon \sigma_D C_I^2}{2} |e_h^{n+1}|_{H^1(\Omega_h^\Gamma)}^2$$
$$\geqslant \frac{3}{4} \alpha |e_h^{n+1}|_{H^1(\Omega_h)}^2 - \frac{1}{2\Delta t} \|e_h^{n+1} - e_h^n\|_{L^2(\Omega_h^\Gamma)}^2, \tag{2.21}$$

où ϵ est choisi tel que $\epsilon\sigma_D C_I^2/2 = \alpha/4$ et où l'on suppose que $\sigma_D h^2/(\epsilon\Delta t) \leqslant 1$. Cela nous permet de contrôler le terme négatif de (2.21) avec le terme similaire positif de (2.20). On obtient alors la contrainte $\Delta t \geqslant ch^2$ où $c = \sigma_D/\epsilon$.

On considère maintenant le terme (III) de (2.20). Par inégalité triangulaire,

$$\left\|\partial_t \tilde{u}^{n+1} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t}\right\|_{L^2(\Omega_h)} \leqslant \left\|\partial_t \tilde{u}^{n+1} - \frac{\tilde{u}^{n+1} - \tilde{u}^n}{\Delta t}\right\|_{L^2(\Omega_h)} + \left\|\frac{\tilde{u}^{n+1} - \tilde{u}^n}{\Delta t} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t}\right\|_{L^2(\Omega_h)}. \quad (2.22)$$

Par le théorème de Taylor avec reste intégral,

$$\tilde{u}^n(\cdot) = \tilde{u}^{n+1}(\cdot) - \Delta t \partial_t \tilde{u}^{n+1}(\cdot) - \int_{t_n}^{t_{n+1}} \partial_{tt}\tilde{u}(t,\cdot)(t_n - t)\,\mathrm{d}\,t\,.$$

Ainsi,

$$\begin{aligned}\left\|\partial_t \tilde{u}^{n+1} - \frac{\tilde{u}^{n+1} - \tilde{u}^n}{\Delta t}\right\|_{L^2(\Omega_h)} &= \frac{1}{\Delta t}\left\|\int_{t_n}^{t_{n+1}} \partial_{tt}\tilde{u}(t,\cdot)(t_n - t)\,\mathrm{d}\,t\right\|_{L^2(\Omega_h)} \\ &\leqslant \sqrt{\Delta t}\|\partial_{tt}\tilde{u}\|_{L^2(t_n,t_{n+1};L^2(\Omega_h))}\,.\end{aligned}$$

Dériver $-\Delta u = f - \partial_t u$ et (2.19) en temps, entraîne alors, par le Lemme 2.7,

$$\|\partial_t(\tilde{u}(t) - \varphi_h \tilde{w}_h)(t)\|_{L^2(\Omega_h)} \leqslant Ch^{k+\frac{1}{2}}\|(\partial_t f - \partial_{tt}\tilde{u})(t)\|_{H^{k-1}(\Omega_h)}.$$

Alors, pour le second terme de (2.22), on obtient finalement

$$\begin{aligned}\left\|\frac{\tilde{u}^{n+1} - \tilde{u}^n}{\Delta t} - \varphi_h \frac{\tilde{w}_h^{n+1} - \tilde{w}_h^n}{\Delta t}\right\|_{L^2(\Omega_h)} &= \frac{1}{\Delta t}\left\|\int_{t_n}^{t_{n+1}} \partial_t(\tilde{u}(t,\cdot) - \varphi_h \tilde{w}_h(t,\cdot))\,\mathrm{dt}\right\|_{L^2(\Omega_h)} \\ &\leqslant \frac{Ch^{k+\frac{1}{2}}}{\sqrt{\Delta t}}\|\partial_t f - \partial_{tt}\tilde{u}\|_{L^2(t_n,t_{n+1};H^{k-1}(\Omega_h))}.\end{aligned}$$

En utilisant toutes les estimations et en appliquant l'inégalité de Young avec $\delta > 0$ ainsi que l'inégalité de Poincaré du Lemme 2.8,

$$(III) \leqslant \frac{C}{\delta}\left(\Delta t\|\partial_{tt}\tilde{u}\|^2_{L^2(t_n,t_{n+1};L^2(\Omega_h))} + \frac{h^{2k+1}}{\Delta t}\|\partial_t f - \partial_{tt}\tilde{u}\|^2_{L^2(t_n,t_{n+1};H^{k-1}(\Omega_h))}\right) + \frac{\delta C_P^2}{2}|e_h^{n+1}|^2_{H^1(\Omega_h)}. \quad (2.23)$$

En remplaçant (2.21) et (2.23) dans (2.20) et en prenant δ tel que $\delta C_P^2 = \alpha/2$, on obtient

$$\begin{aligned}&\frac{\|e_h^{n+1}\|^2_{L^2(\Omega_h)} - \|e_h^n\|^2_{L^2(\Omega_h)}}{2\Delta t} + \frac{\alpha}{2}|e_h^{n+1}|^2_{H^1(\Omega_h)} \\ &\qquad\leqslant C\left(\Delta t\|\partial_{tt}\tilde{u}\|^2_{L^2(t_n,t_{n+1};L^2(\Omega_h))} + \frac{h^{2k+1}}{\Delta t}\|\partial_t f - \partial_{tt}\tilde{u}\|^2_{L^2(t_n,t_{n+1};H^{k-1}(\Omega_h))}\right),\end{aligned}$$

ce qui, multiplié par $2\Delta t$ et sommé sur l'ensemble des $n = 0, \dots, N-1$, donne

$$\|e_h^N\|^2_{L^2(\Omega_h)} + \alpha\Delta t \sum_{n=1}^{N} |e_h^n|^2_{H^1(\Omega_h)} \\ \leqslant \|e_h^0\|^2_{L^2(\Omega_h)} + C(\Delta t^2 \|\partial_{tt}\tilde{u}\|^2_{L^2(0,T;L^2(\Omega_h))} + h^{2k+1}\|\partial_t f - \partial_{tt}\tilde{u}\|^2_{L^2(0,T;H^{k-1}(\Omega_h))}).$$

Alors, en observant que la somme peut être arrêtée pour tout $n \leqslant N$,

$$\max_{n=1,\dots,N} \|e_h^n\|_{L^2(\Omega_h)} + \left(\Delta t \sum_{n=1}^{N} |e_h^n|^2_{H^1(\Omega_h)}\right)^{\frac{1}{2}} \\ \leqslant C\|e_h^0\|_{L^2(\Omega_h)} + C\left(\Delta t \|\partial_{tt}\tilde{u}\|_{L^2(0,T;L^2(\Omega_h))} + h^{k+\frac{1}{2}}\|\partial_t f - \partial_{tt}\tilde{u}\|_{L^2(0,T;H^{k-1}(\Omega_h))}\right).$$

Le Lemme 2.7 appliqué à $-\Delta u = f - \partial_t u$ dans Ω au temps t_n donne alors

$$\max_{n=0,\dots,N} \|\tilde{u}^n - \varphi_h \tilde{w}_h^n\|_{L^2(\Omega_h)} \leqslant Ch^{k+1/2}\|f - \partial_t\tilde{u}\|_{C([0,T],H^{k-1}(\Omega_h))},$$

$$\left(\Delta t \sum_{n=1}^{N} |\tilde{u}^n - \varphi_h \tilde{w}_h^n|^2_{H^1(\Omega_h)}\right)^{\frac{1}{2}} \leqslant Ch^k\|f - \partial_t\tilde{u}\|_{C([0,T],H^{k-1}(\Omega_h))}.$$

En particulier,

$$\|e_h^0\|_{L^2(\Omega_h)} \leqslant \|u^0 - u_h^0\|_{L^2(\Omega_h)} + \|u^0 - \varphi_h\tilde{w}_h^0\|_{L^2(\Omega_h)} \\ \leqslant \|u^0 - u_h^0\|_{L^2(\Omega_h)} + Ch^{k+1/2}\|f - \partial_t\tilde{u}\|_{C([0,T],H^{k-1}(\Omega_h))}.$$

Cela combiné avec la régularité de f et de $\tilde{u}$, cf. (2.17), ainsi qu'avec la majoration $\|\cdot\|_{C([0,T],\cdot)} \leqslant C\|\cdot\|_{H^1(0,T;\cdot)}$ (où C dépend de T) nous donne finalement le résultat annoncé. □

2.3.3 Résultats numériques

Dans cette partie, nous allons valider numériquement les performances de notre méthode sur deux cas test [4]. Les implémentations sont faites avec *FEniCS* [2]. Les codes python des simulations sont disponibles dans le repository Github

`https://github.com/PhiFEM/publication_Heat-Equation_fenics`

Dans les simulations suivantes, si la convergence espérée est d'ordre $C_1h^p + C_2\Delta t^m$, nous fixerons $\Delta t = h^{p/m}$ de sorte qu'il soit suffisant d'observer si l'erreur est d'ordre h^p.

Remarque 2.13 (Normes utilisées pendant les simulations). Pour illustrer la convergence des méthodes, nous considérerons les normes suivantes :

$$\frac{\|u_h - u_{\text{ref}}\|^2_{l^2(0,T,H_0^1(\Omega_{\text{ref}}))}}{\|u_{\text{ref}}\|^2_{l^2(0,T,H_0^1(\Omega_{\text{ref}}))}} \approx \frac{\sum_{n=0}^{N}\Delta t\int_{\Omega_{\text{ref}}}|\nabla u_h(.,t_n) - \nabla u_{\text{ref}}(.,t_n)|^2\mathrm{d}x}{\sum_{n=0}^{N}\Delta t\int_{\Omega_{\text{ref}}}|\nabla u_{\text{ref}}(.,t_n)|^2\mathrm{d}x},$$

4. Pour le premier cas test, nous utiliserons le solveur linéaire par défaut de *FEniCS*. Pour le second, le solveur linéaire *GMRES* sera utilisé, combiné au préconditionneur *hypre_amg*.

et

$$\frac{\|u_h - u_{\text{ref}}\|^2_{l^\infty(0,T,L^2(\Omega_{\text{ref}}))}}{\|u_{\text{ref}}\|^2_{l^\infty(0,T,L^2(\Omega_{\text{ref}}))}} \approx \frac{\max_{n=0,\dots,N} \int_{\Omega_{\text{ref}}} (u_h(.,t_n) - u_{\text{ref}}(.,t_n))^2 \mathrm{d}x}{\max_{n=0,\dots,N} \int_{\Omega_{\text{ref}}} (u_{\text{ref}}(.,t_n))^2 \mathrm{d}x},$$

où l'on note u_h une approximation de la projection orthogonale L^2 de la solution calculée, sur un maillage de référence $\mathcal{T}_{\text{ref}}$ du domaine Ω_{ref} et u_{ref} la solution de référence.

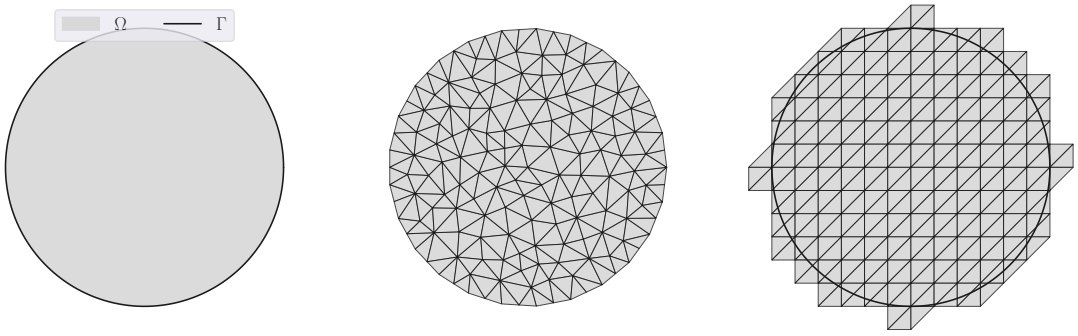

FIGURE 2.18 – **Cas test 1.** Gauche : domaine considéré. Centre : maillage conforme pour FEM standard. Droite : maillage cartésien uniforme pour φ-FEM ($\mathcal{T}_h$).

Premier cas test : solution manufacturée. Pour ce premier cas test, nous considérons un domaine simple : le cercle centré en $(0,0)$, de rayon 1, comme représenté à la Figure 2.18. La fonction level-set φ est donnée en utilisant l'équation d'un cercle, i.e. $\varphi(x,y) = -1 + x^2 + y^2$. Son approximation φ_h est l'interpolation de φ avec des éléments finis $\mathbb{P}_{k+1}$, hormis pour les résultats présentés à la Figure. 2.24 (droite).

La solution manufacturée $u_{\text{ref}} = \cos\left(\frac{1}{2}\pi(x^2+y^2)\right)\exp(x)\sin(t)$ est telle que u_{ref} vérifie $u_{\text{ref}}(t=0) = u^0_{\text{ref}} = 0$ et $u_{\text{ref}} = 0$ sur $\Gamma \times (0,T)$. Ici, le maillage de référence sera le maillage considéré à chaque résolution φ-FEM et FEM standard (i.e. il n'y a pas d'interpolation sur un maillage plus fin pour le calcul de l'erreur).

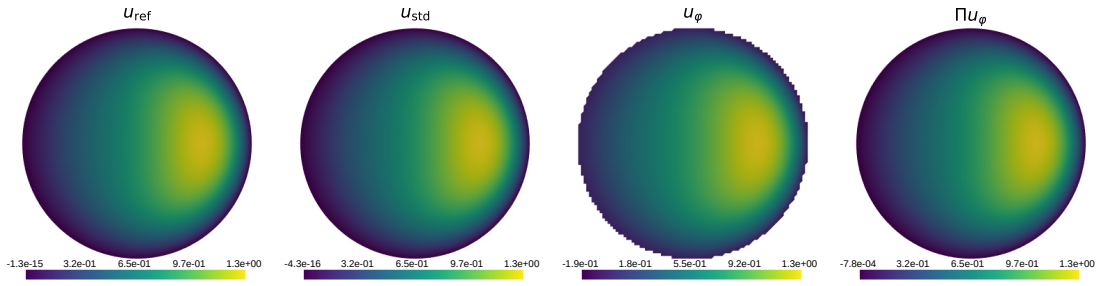

FIGURE 2.19 – **Cas test 1.** Représentations des solutions au temps final. De gauche à droite : solution de référence FEM standard, solution FEM standard, solution φ-FEM, projection de la solution φ-FEM sur le maillage de référence.

On représente à la Figure 2.19 u_{ref} calculée sur un maillage fin, au temps final, ainsi qu'une solution éléments finis et une solution φ-FEM toutes deux au temps final. On représente également la projection de la solution φ-FEM sur un maillage fin conforme.

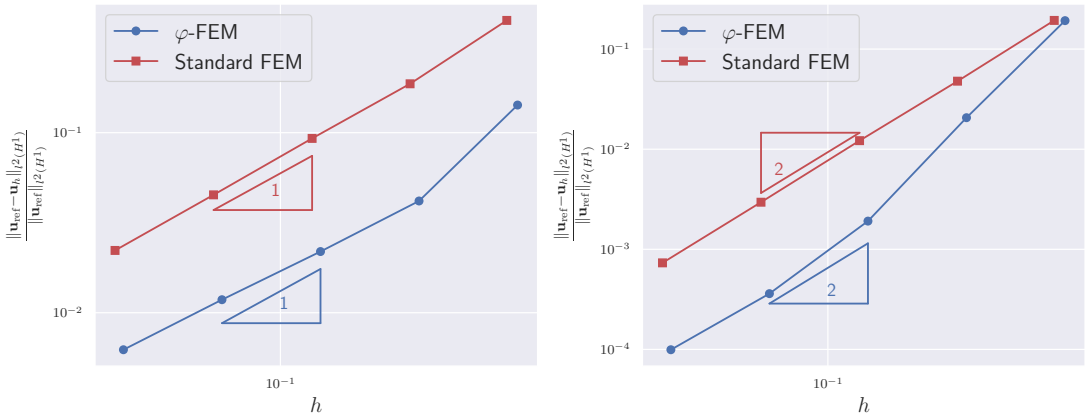

FIGURE 2.20 – **Cas test 1.** Erreurs relatives $l^2(0,T;H^1(\Omega))$ en fonction de h pour des éléments finis $\mathbb{P}_1$ et $\Delta t = h$ (gauche) et $\mathbb{P}_2$ avec $\Delta t = h^2$ (droite).

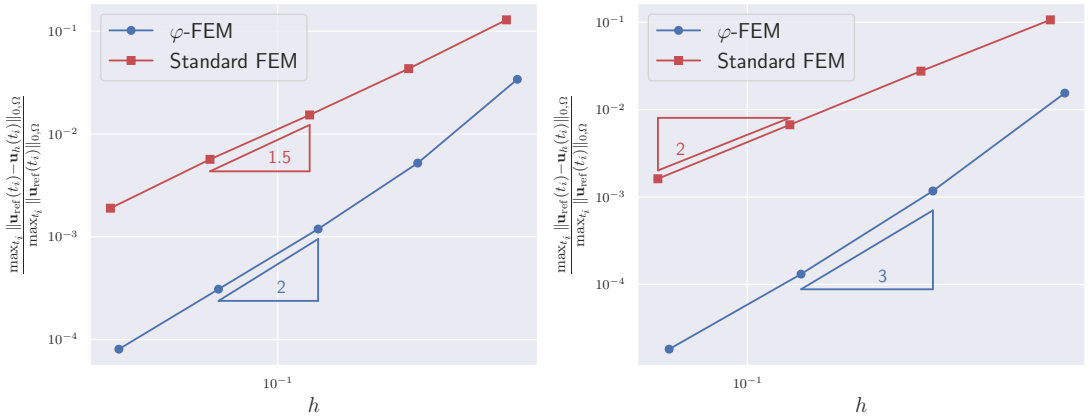

FIGURE 2.21 – **Cas test 1.** Erreurs relatives $l^\infty(0,T;L^2(\Omega))$ en fonction de h pour des éléments finis $\mathbb{P}_1$ et $\Delta t = h^2$ (gauche) et $\mathbb{P}_2$ avec $\Delta t = h^3$ (droite).

On représente l'erreur en norme $l^2(H^1)$ à la Figure 2.20 et en norme $l^\infty(L^2)$ sur la Figure 2.21, dans les deux cas pour des éléments finis $\mathbb{P}_1$ et $\mathbb{P}_2$ ($k = 1$ et $k = 2$).
Les résultats numériques correspondent bien à l'ordre de convergence théorique annoncé dans le Théorème 2.3 et se comportent même mieux puisque l'on observe des convergences d'ordre 2 et 3 en norme $l^\infty(L^2)$ au lieu de 1.5 et 2.5 respectivement. Il est intéressant de remarquer que la contrainte théorique $\Delta t \geqslant ch^2$ n'est pas satisfaite pour les éléments finis $\mathbb{P}^2$, ce qui n'affecte pas la convergence numérique. On représente également les erreurs en normes $l^2(H^1)$ et $l^\infty(L^2)$ en fonction du temps de calcul (ici, la somme du temps

d'assemblage de la matrice éléments finis et du temps de résolution du système linéaire à chaque pas de temps, sans prendre en compte les temps de construction des maillages) à la Figure 2.22. On observe alors que φ-FEM est significativement plus rapide qu'une méthode éléments finis classique pour obtenir une solution à seuil d'erreur fixé.

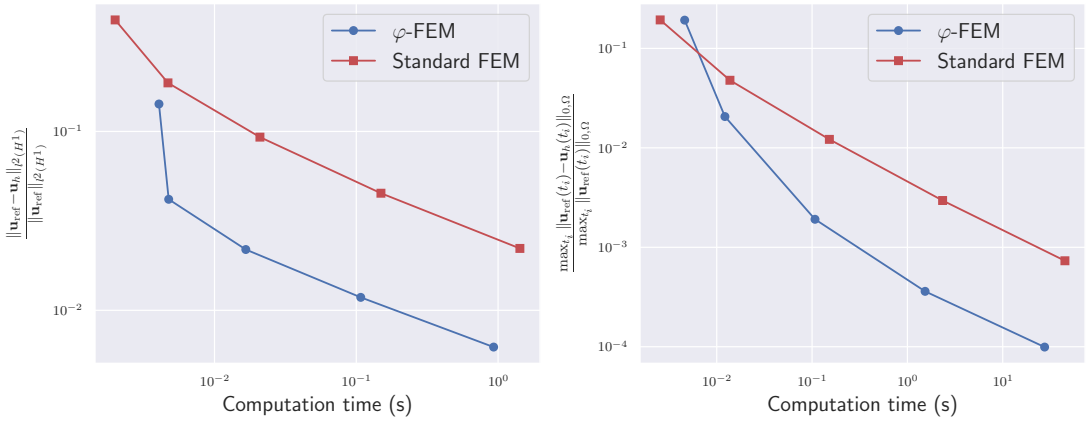

FIGURE 2.22 – **Cas test 1.** Erreurs relatives $l^2(0,T;H^1(\Omega))$ avec $\Delta t = h$ (gauche) et $l^\infty(0,T;L^2(\Omega))$ avec $\Delta t = h^2$ (droite) en fonction du temps de calcul.

La Figure 2.23 (gauche), représente l'erreur $l^2(H^1)$ et la Figure 2.23 (droite) l'erreur $l^\infty(L^2)$, dans les deux cas en fonction du paramètre de stabilisation σ_D. Cela permet d'illustrer l'influence de σ_D sur la stabilité de l'erreur, ainsi que de valider le choix de la valeur $\sigma_D = 1$ dans les autres simulations.

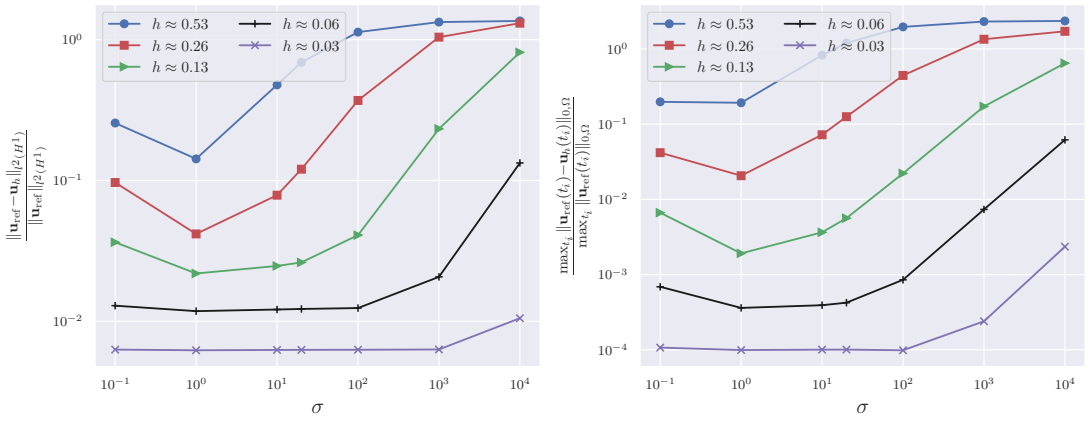

FIGURE 2.23 – **Cas test 1.** Gauche : Erreurs relatives $l^2(0,T;H^1(\Omega))$ en fonction de σ_D différentes tailles de maillage h, avec $\Delta t = h$. Droite : Erreurs relatives $l^\infty(0,T;L^2(\Omega))$ en fonction de σ_D, avec $\Delta t = h^2$.

Enfin, la Figure 2.24, permet de justifier le choix du degré d'interpolation de φ puisque dans l'analyse théorique, $\mathbb{P}_k$ est suffisant, mais on observe que l'erreur de la méthode est

plus faible pour $l = 2$. Ici, puisque l'interpolation est exacte à partir de $l = 2$, il n'est pas nécessaire de comparer les résultats avec un plus haut degré d'interpolation de φ.

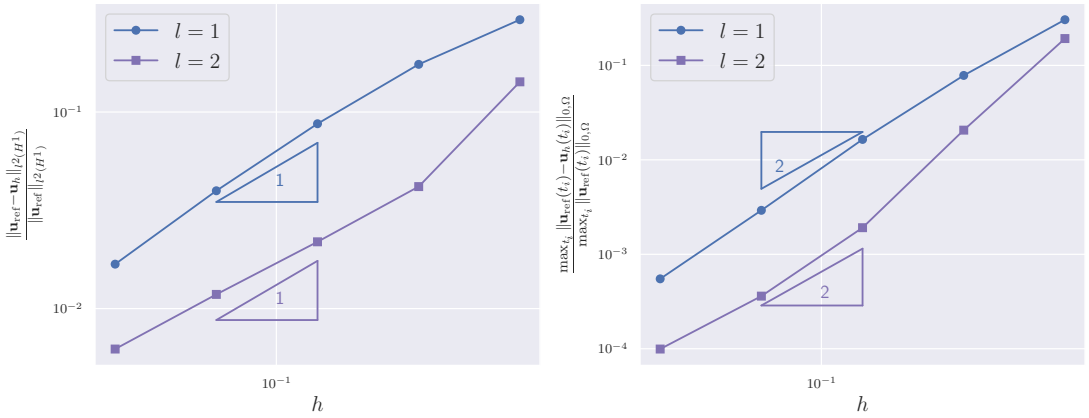

FIGURE 2.24 – **Cas test 1.** Erreurs relatives $l^2(0,T;H^1(\Omega))$ en fonction de h pour différentes valeurs de l, $\Delta t = h$ (gauche) et erreurs relatives $l^\infty(0,T;L^2(\Omega))$ avec $\Delta t = h^2$ (droite).

Second cas test : terme source donné. On considère maintenant un cas test plus réaliste où l'on applique un terme source connu et cherche à déterminer la distribution de la chaleur dans le domaine considéré. Plus précisément, on impose $u = 0$ sur $\Gamma \times (0,T)$. La condition initiale est donnée par $u^0 = 0$ dans Ω on définit un terme source $f(x,y,z,t) = \exp\left(-\frac{(x-\mu_1)^2+(y-\mu_2)^2+(z-\mu_3)^2}{2\sigma_0^2}\right)$ pour tout $(x,y,z,t) \in \Omega \times (0,T)$, avec $(\mu_1,\mu_2,\mu_3,\sigma_0) = (0.2, 0.3, -0.1, 0.3)$. Le temps final est fixé à $T = 1$.
De plus, pour ce cas test le domaine considéré sera un domaine 3D plus complexe, issu de [13], donné par

$$\varphi(x,y,z) = x^2 + y^2 + z^2 - r_0^2 - A\sum_{k=0}^{11} \exp\left(-\frac{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2}{\sigma_0^2}\right),$$

où

$$(x_k,y_k,z_k) = \frac{r_0}{\sqrt{5}}\left(2\cos\left(\frac{2k\pi}{5}\right), 2\sin\left(\frac{2k\pi}{5}\right), 1\right), \quad 0 \leqslant k \leqslant 4,$$
$$(x_k,y_k,z_k) = \frac{r_0}{\sqrt{5}}\left(2\cos\left(\frac{(2(k-5)-1)\pi}{5}\right), 2\sin\left(\frac{(2(k-5)-1)\pi}{5}\right), -1\right), \quad 5 \leqslant k \leqslant 9,$$
$$(x_k,y_k,z_k) = (0,0,r_0), \quad k = 10,$$
$$(x_k,y_k,z_k) = (0,0,-r_0), \quad k = 11,$$

avec $r_0 = 0.6$, $\sigma = 0.3$ et $A = 1.5$.

Le domaine et des exemples de maillages (Standard-FEM et φ-FEM) construits sont représentés à la Figure 2.25.

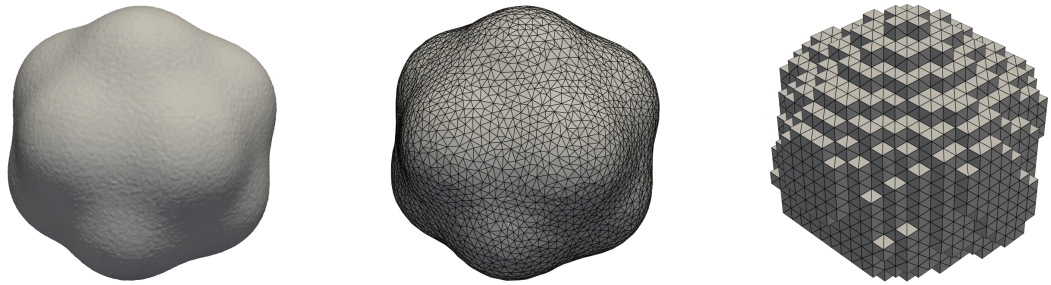

FIGURE 2.25 – **Cas test 2.** Gauche : domaine considéré. Centre : maillage conforme standard FEM. Droite : maillage uniforme cartésien $\mathcal{T}_h$ pour φ-FEM.

Ici, on notera u_{ref} la solution obtenue par Standard-FEM sur un maillage conforme très fin $\mathcal{T}_{\text{ref}}$ du domaine de référence Ω_{ref}. En particulier, on introduit une partition de l'intervalle $[0,T]$ en pas de temps $0 = t_0^{\text{ref}} < t_1^{\text{ref}} < \cdots < t_M^{\text{ref}} = T$ avec $t_n^{\text{ref}} = n\Delta t^{\text{ref}}$ et $\Delta t^{\text{ref}} = h_{\text{ref}}^{p/m}$, où h_{ref} est la taille de cellules de $\mathcal{T}_{\text{ref}}$. Ainsi, dans les simulations, chaque discrétisation est construite de sorte que $\{t_n\}_{n=0,\dots,N}$ soit un sous-ensemble de $\left\{t_n^{\text{ref}}\right\}_{n=0,\dots,M}$.
On représente à la Figure 2.27 u_{ref}, au temps final, ainsi qu'une solution éléments finis et une solution φ-FEM toutes deux au temps final. On représente également la projection de la solution φ-FEM sur un maillage fin.

Pour la Figure 2.26, on considère des éléments finis $\mathbb{P}_1$ ($k = 1$), et φ_h est l'interpolation $\mathbb{P}_2$ de φ ($l = 2$). On compare les erreurs relatives en normes $l^2(H^1)$, $l^\infty(L^2)$ entre les solutions du schéma φ-FEM (2.16) et les solutions avec FEM classique. Dans ce cas également, les résultats numériques correspondent aux résultats théoriques énoncés dans le Théorème 2.3, c'est-à-dire, l'ordre 1 pour la norme $l^2(H^1)$ et l'ordre 2 pour la norme $l^\infty(L^2)$.

2.4 Résolution de problèmes d'élasticité linéaire

Dans cette section, nous allons introduire plusieurs schémas φ-FEM permettant de résoudre différents problèmes d'élasticité linéaire. Dans un premier temps, nous considérerons un problème générique d'élasticité linéaire avec des conditions de Dirichlet ou mixtes de Dirichlet/Neumann. Ensuite, nous verrons comment résoudre un problème d'élasticité impliquant plusieurs matériaux, dans le cas de problèmes avec interface. Nous traiterons également le cas de matériaux élastiques contenant une fracture. Ces résultats ont fait l'objet de la publication [22]. Enfin, nous proposerons de nouveaux résultats numériques illustrant l'intérêt de notre approche dans le cas de problèmes plus réalistes pouvant notamment présenter des singularités.

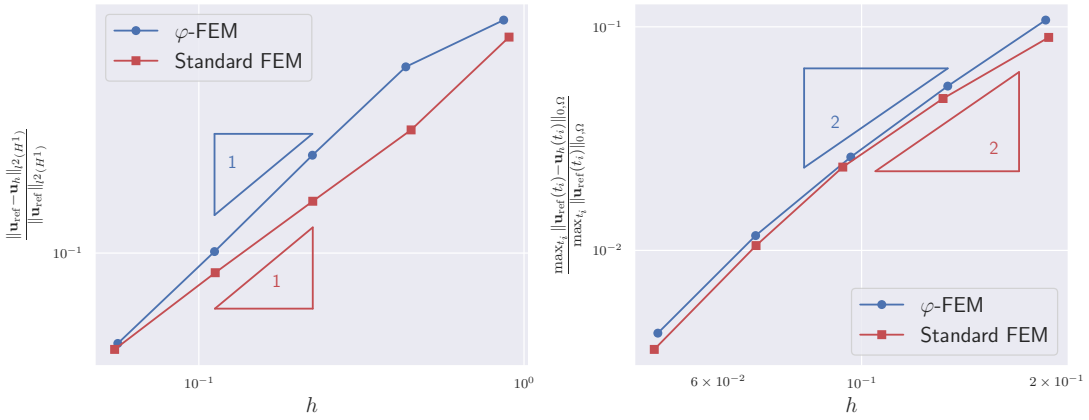

FIGURE 2.26 – **Cas test 2.** Erreurs relatives $l^2(0,T;H^1(\Omega))$ en fonction de h avec $\Delta t = h$ (gauche) et erreurs relatives $l^\infty(0,T;L^2(\Omega))$ en fonction de h, avec $\Delta t = h^2$ (droite).

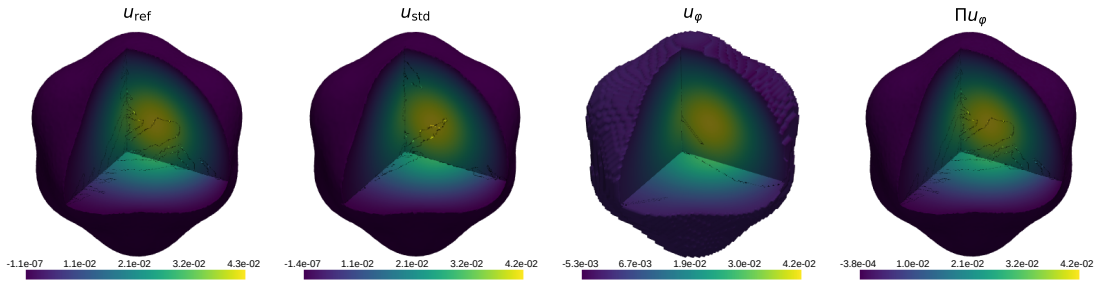

FIGURE 2.27 – **Cas test 2.** Représentation des solutions du cas test 2 au temps final. De gauche à Droite : solution de référence, solution FEM standard, solution φ-FEM et solution φ projetée sur le maillage de référence.

2.4.1 L'élasticité linéaire avec conditions Dirichlet et mixtes Dirichlet/Neumann

Considérons premièrement le cas de l'élasticité linéaire statique pour des matériaux homogènes et isotropes. Le problème consiste à trouver un déplacement $\boldsymbol{u} \in \mathbb{R}^d$ pour un déplacement donné $\boldsymbol{u}^g$ sur Γ_D (conditions de Dirichlet), une traction $\boldsymbol{g}$ sur Γ_N (conditions de Neumann) et une force interne $\boldsymbol{f}$ dans Ω, vérifiant

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}) + \boldsymbol{f} &= 0\,, \quad \text{dans } \Omega\,, \\ \boldsymbol{u} &= \boldsymbol{u}^g\,, \quad \text{sur } \Gamma_D\,, \\ \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \boldsymbol{g}\,, \quad \text{sur } \Gamma_N\,, \end{cases} \tag{2.24}$$

où le tenseur des contraintes $\boldsymbol{\sigma}(\boldsymbol{u})$ est donné par

$$\boldsymbol{\sigma}(\boldsymbol{u}) = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda(\operatorname{div}\boldsymbol{u})I,$$

avec $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ le tenseur de déformation et les paramètres de Lamé λ et μ dépendant du module de Young E et du coefficient de Poisson ν,

$$\mu = \frac{E}{2(1+\nu)} \text{ et } \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}\,. \tag{2.25}$$

Rappelons premièrement la formulation faible associée à l'équation (2.24). Pour cela, on suit l'approche classique : on multiplie l'équation par une fonction test $\boldsymbol{v}$ et on intègre par parties sur Ω. On cherche alors le champ de vecteur $\boldsymbol{u}$ dans Ω vérifiant $\boldsymbol{u}|_{\Gamma_D} = \boldsymbol{u}^g$ et

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \nabla \boldsymbol{v} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} + \int_{\Gamma_N} \boldsymbol{g} \cdot \boldsymbol{v}, \quad \forall \boldsymbol{v} \text{ dans } \Omega \text{ tel que } \boldsymbol{v}|_{\Gamma_D} = 0.$$

Cette formulation sera utilisée pour construire les schémas éléments finis classiques utilisés dans les simulations numériques de cette section.

Une fois de plus, on considère le cas où Ω est inscrit dans une boite $\mathcal{O}$, couverte par le maillage $\mathcal{T}_h^{\mathcal{O}}$. De plus, on construit les maillages $\mathcal{T}_h$ (c.f. (1.6)) et $\mathcal{T}_h^\Gamma$ (c.f. (1.7)). Enfin, on suppose que l'on connait les différentes fonctions sur Ω_h plutôt que seulement sur Ω.

On peut alors, comme pour les précédents schémas φ-FEM étendre la formulation (2.24) à Ω_h. Alors, multiplier par une fonction test $\boldsymbol{v}$ et intégrer par parties sur Ω_h, donne la formulation : trouver $\boldsymbol{u}$ dans Ω_h tel que

$$\int_{\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}) : \nabla \boldsymbol{v} - \int_{\partial\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} \cdot \boldsymbol{v} = \int_{\Omega_h} \boldsymbol{f} \cdot \boldsymbol{v}, \quad \forall \boldsymbol{v} \text{ dans } \Omega_h\,.$$

Conditions de Dirichlet

On considère premièrement le cas de conditions de bord de Dirichlet pures, c'est-à-dire lorsque $\Gamma = \Gamma_D$. Comme pour le problème de Poisson, nous allons proposer deux versions du schéma : la version directe ($\boldsymbol{u} = \varphi \boldsymbol{w} + \boldsymbol{u}^g$ dans tout Ω_h) et la version duale ($\boldsymbol{u} = \varphi \boldsymbol{p} + \boldsymbol{u}^g$ uniquement sur les cellules « proches » de Γ).

Introduisons premièrement les espaces éléments finis adaptés aux problèmes d'élasticité dans lesquels les variables seront discrétisées. Pour $k \geqslant 1$, soit

$$\boldsymbol{V}_h := \left\{ \boldsymbol{v}_h : \Omega_h \to \mathbb{R}^d : \boldsymbol{v}_{h|T} \in \mathbb{P}^k(T)^d \;\; \forall T \in \mathcal{T}_h,\; \boldsymbol{v}_h \text{ continue sur } \Omega_h \right\}, \tag{2.26}$$

l'espace de discrétisation des variables « principales ».

Comme nous l'avons fait dans le cas du schéma dual pour Poisson-Dirichlet, il est nécessaire d'introduire la version locale de cet espace, défini pour tout maillage $\mathcal{M}_h$ couvrant un domaine M_h et pour $l \geqslant 0$, par

$$\boldsymbol{Q}_h^k(M_h) := \{\boldsymbol{q}_h : M_h \to \mathbb{R}^d : \boldsymbol{q}_{h|T} \in \mathbb{P}^k(T)^d \;\; \forall T \in \mathcal{M}_h,\; \boldsymbol{q}_h \text{ continue sur } M_h \text{ si } k \geqslant 0\}. \tag{2.27}$$

En particulier, nous aurons besoin de $\boldsymbol{Q}_h^k(\Omega_h^\Gamma)$ sur le sous-maillage $\mathcal{T}_h^\Gamma$ pour la version duale.

Maintenant que les espaces éléments finis sont définis, on peut alors introduire les deux schémas φ-FEM permettant de résoudre (2.24) avec des conditions de Dirichlet pures :

- **φ-FEM direct Dirichlet** : le schéma direct est donné par, trouver $\boldsymbol{w}_h \in \boldsymbol{V}_h$ tel que

$$\begin{aligned}\int_{\Omega_h} \boldsymbol{\sigma}(\varphi_h \boldsymbol{w}_h) : \nabla(\varphi_h \boldsymbol{z}_h) - \int_{\partial\Omega_h} \boldsymbol{\sigma}(\varphi_h \boldsymbol{w}_h)\boldsymbol{n} \cdot \varphi_h \boldsymbol{z}_h + G_h(\varphi_h \boldsymbol{w}_h, \varphi_h \boldsymbol{z}_h) \\ + J_h^{lhs}(\varphi_h \boldsymbol{w}_h, \varphi_h \boldsymbol{z}_h) = \int_{\Omega_h} \boldsymbol{f} \cdot \varphi_h \boldsymbol{z}_h - \int_{\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}_h^g) : \nabla(\varphi_h \boldsymbol{z}_h) \\ + \int_{\partial\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}_h^g)\boldsymbol{n} \cdot \varphi_h \boldsymbol{z}_h, + J_h^{rhs}(\varphi_h \boldsymbol{z}_h), \quad \forall \boldsymbol{z}_h \in \boldsymbol{V_h}\end{aligned}$$

avec $\boldsymbol{u}_h = \boldsymbol{u}_h^g + \varphi_h \boldsymbol{w}_h$. Ici, φ_h et $\boldsymbol{u}_h^g$ sont les approximations éléments finis de φ et $\boldsymbol{u}^g$ sur Ω_h. De plus, G_h, J_h^{lhs} et J_h^{rhs} sont les termes de stabilisation définis par

$$G_h(\boldsymbol{u}, \boldsymbol{v}) := \sigma_D h \sum_{E \in \mathcal{F}_h^\Gamma} \int_E [\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n}] \cdot [\boldsymbol{\sigma}(\boldsymbol{v})\boldsymbol{n}] \,, \tag{2.28}$$

$$J_h^{lhs}(\boldsymbol{u}, \boldsymbol{v}) := \sigma_D h^2 \sum_{T \in \mathcal{T}_h^\Gamma} \int_T \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{v}) \,, \tag{2.29}$$

$$J_h^{rhs}(\boldsymbol{v}) := -\sigma_D h^2 \sum_{T \in \mathcal{T}_h^\Gamma} \int_T \boldsymbol{f} \cdot \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{v}) \,. \tag{2.30}$$

Ici, G_h est une adaptation aux équations d'élasticité de la « ghost penalty » introduite à l'équation (1.11) pour le problème de Poisson-Dirichlet, avec $\sigma_D > 0$. Cependant, dans ce cas, on choisit de pénaliser le saut des forces élastiques internes (en suivant l'approche [21]), et donc de contrôler les combinaisons appropriées des dérivées plutôt que les dérivées normales directement. Une représentation des faces sur lesquelles cette stabilisation est appliquée est donnée à la Figure 1.4, puisque l'ensemble $\mathcal{F}_h^\Gamma$ est défini par (1.8). Les stabilisations d'ordre 2 sont introduites de sorte à imposer l'équation (2.24) aux moindres carrés sur les cellules coupées par la frontière.

- **φ-FEM dual Dirichlet** : le schéma dual est lui défini par, trouver $\boldsymbol{u}_h \in \boldsymbol{V}_h$, $\boldsymbol{p}_h \in \boldsymbol{Q}_h^k(\Omega_h^\Gamma)$ tels que

$$\begin{aligned}\int_{\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h - \int_{\partial\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}_h)\boldsymbol{n} \cdot \boldsymbol{v}_h + \frac{\gamma}{h^2} \int_{\Omega_h^\Gamma} (\boldsymbol{u}_h - \frac{1}{h}\varphi_h \boldsymbol{p}_h) \cdot (\boldsymbol{v}_h - \frac{1}{h}\varphi_h \boldsymbol{q}_h) \\ + G_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + J_h^{lhs}(\boldsymbol{u}_h, \boldsymbol{v}_h) = \int_{\Omega_h} \boldsymbol{f} \cdot \boldsymbol{v}_h \\ + \frac{\gamma}{h^2} \int_{\Omega_h^\Gamma} \boldsymbol{u}_h^g \cdot (\boldsymbol{v}_h - \frac{1}{h}\varphi_h \boldsymbol{q}_h) + J_h^{rhs}(\boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h,\ \boldsymbol{q}_h \in \boldsymbol{Q}_h^k(\Omega_h^\Gamma).\end{aligned} \tag{2.31}$$

Les termes de stabilisation G_h, J_h^{lhs} et J_h^{rhs} sont définis respectivement par (2.28), (2.29) et (2.30).

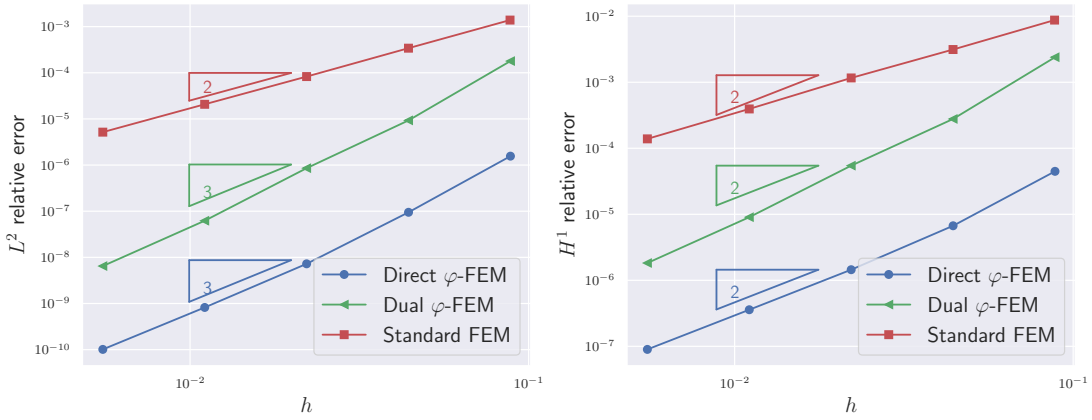

FIGURE 2.28 – **Cas test 1.** (Conditions de Dirichlet). Erreur relative L^2 (gauche), erreur relative H^1 (droite).

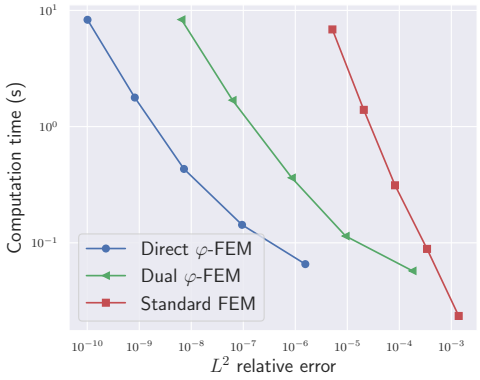

FIGURE 2.29 – **Cas test 1.** (Conditions de Dirichlet). Temps de calcul (en secondes) en fonction de l'erreur relative L^2.

Cas test 1. Soit $\mathcal{O}$ le carré $(0,1)^2$ et soit $\mathcal{T}_h^{\mathcal{O}}$ un maillage uniforme de $\mathcal{O}$. Soit Ω le cercle de centre $(0.5, 0.5)$ de rayon $\frac{\sqrt{2}}{4}$, défini par la fonction level-set

$$\varphi(x,y) = -\frac{1}{8} + (x-0.5)^2 + (y-0.5)^2 \,. \tag{2.32}$$

Les paramètres d'élasticité seront fixés à $E = 2$ et $\nu = 0.3$ et les paramètres de stabilisation à $\gamma = \sigma_D = 20.0$. Des éléments finis $\mathbb{P}^2$ seront utilisés pour $\boldsymbol{V}_h$ et $\boldsymbol{Q}_h$, i.e. $k = 2$ dans (2.26) et (2.27). Finalement, on considérera une solution manufacturée donnée par

$$\boldsymbol{u} = \boldsymbol{u}_{ex} := (\sin(x)\exp(y), \sin(y)\exp(x)) \,. \tag{2.33}$$

Le second membre $\boldsymbol{f}$ de (2.24) est alors calculé analytiquement et les conditions de bord $\boldsymbol{u}^g$ sont données par $\boldsymbol{u}^g = \boldsymbol{u}_{ex}$ sur Γ. Afin d'éviter d'utiliser cette expression sur l'ensemble de Ω_h (schéma direct) ou sur Ω_h^Γ (schéma dual), on perturbera légèrement

cette condition de bord, et on imposera plutôt

$$\boldsymbol{u}^g = \boldsymbol{u}_{ex}(1+\varphi), \quad \text{dans } \Omega_h \text{ ou dans } \Omega_h^\Gamma.$$

Remarque 2.14. Les représentations des maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ peuvent être trouvées à la Figure 1.4. De plus, un maillage conforme pour une méthode éléments finis dans le cas considéré ici est représenté à la Figure 1.1.

Nous allons dans un premier temps étudier la convergence de nos deux schémas ainsi que celle de la méthode éléments finis classique. Pour cela, nous mesurons les erreurs L^2 et H^1, qui sont représentées à la Figure 2.28. On remarque que les deux schémas φ-FEM atteignent l'ordre optimal espéré pour les deux normes : h^2 pour la semi-norme H^1 et h^3 pour la norme L^2. De plus, les deux méthodes sont significativement meilleures que l'approche Standard, qui est sous-optimale en norme L^2.

L'efficacité de φ-FEM par rapport à Standard-FEM est également confirmée par la Figure 2.29, où l'on représente le temps de calcul en fonction de l'erreur relative L^2. Les temps de calcul considérés ne prennent en compte que les temps d'assemblage des matrices éléments finis et les temps de résolution des systèmes linéaires. Ainsi, pour une erreur fixée, les résultats sont obtenus significativement plus rapidement avec φ-FEM (direct comme dual), qu'avec Standard-FEM.

Remarque 2.15 (Temps de calcul). Il a été ici choisi de ne pas prendre en compte le temps de génération des différents maillages puisque pour ce cas test, les simulations ont été réalisées avec *FEniCS* qui ne permettait pas de sélectionner les cellules des sous-maillages φ-FEM de manière optimale.

Conditions de bord mixtes

Considérons maintenant le cas plus complexe de conditions mixtes Dirichlet-Neumann au bord sur $\Gamma = \Gamma_N \cup \Gamma_D$ où $\Gamma_D \neq \emptyset$ et $\Gamma_N \neq \emptyset$.

Comme dans le cas du problème de Poisson avec conditions mixtes, on considère une fonction level-set ψ nous permettant de caractériser la partie Neumann et la partie Dirichlet du bord Γ :

$$\Gamma_D = \Gamma \cap \{\psi \leqslant 0\} \quad \text{et } \Gamma_N = \Gamma \cap \{\psi > 0\}.$$

On peut à nouveau introduire les maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$, cf. (1.6) et (1.7) (représentés aux Figures 2.11 et 2.12). La level-set ψ nous permet alors de définir une nouvelle fois les sous-maillages $\mathcal{T}_h^{\Gamma_D}$ et $\mathcal{T}_h^{\Gamma_N}$ cf. (2.13) que l'on rappelle :

$$\mathcal{T}_h^{\Gamma_D} := \{T \in \mathcal{T}_h^\Gamma : \psi \leqslant 0 \text{ sur } T\} \qquad \text{et} \qquad \mathcal{T}_h^{\Gamma_N} := \{T \in \mathcal{T}_h^\Gamma : \psi \geqslant 0 \text{ sur } T\}.$$

On notera Ω_h, Ω_h^Γ, $\Omega_h^{\Gamma_D}$ et $\Omega_h^{\Gamma_N}$ les domaines occupés par les maillages $\mathcal{T}_h$, $\mathcal{T}_h^\Gamma$, $\mathcal{T}_h^{\Gamma_D}$ et $\mathcal{T}_h^{\Gamma_N}$. On rappelle comme dans le cas de l'équation (2.12), Section 2.2, que certaines cellules de $\mathcal{T}_h^\Gamma$ peuvent appartenir aux deux maillages $\mathcal{T}_h^{\Gamma_D}$ et $\mathcal{T}_h^{\Gamma_N}$ ou à aucun (comme représenté à la Figure 2.11). Dans ces deux situations, ces cellules seront considérées comme des cellules d'interface, pour lesquelles aucune condition de bord ne sera appliquée.

Des exemples des maillages construits sont représentés aux Figures 2.11 et 2.12 pour une jonction Neumann/Dirichlet censée intervenir pour $x = 0.5$, i.e. pour une level-set $\psi(x, y) = 0.5 - x$.

Les cellules intersectées par Γ appartiennent soit à la partie Dirichlet (et forment donc $\mathcal{T}_h^{\Gamma_D}$, et sont colorées en violet), ou à la partie Neumann (et forment $\mathcal{T}_h^{\Gamma_N}$, cellules colorées en rouge), ou à la partie d'interface et sont alors colorées en bleu.

On suppose une nouvelle fois que $\boldsymbol{u}$, solution de (2.24) peut être étendue de Ω à Ω_h comme solution de la même équation. On introduit alors le schéma φ-FEM en combinant la version duale φ-FEM Dirichlet (2.31) et l'adaptation au cas de l'élasticité du schéma Poisson-Neumann proposé dans [23], rappelé en Section 1.2 (le schéma étant rappelé à l'équation (1.19)).

Pour imposer les conditions de Dirichlet, on utilisera l'équation

$$\boldsymbol{u} = \boldsymbol{u}^g + \varphi \boldsymbol{p}_D, \text{ sur } \Omega_h^{\Gamma_D},$$

où l'on suppose que $\boldsymbol{u}^g$ est étendue de Γ_D à $\Omega_h^{\Gamma_D}$.

Les conditions de Neumann seront imposées via l'introduction de deux variables auxiliaires (comme détaillé en Sections 1.2 et 2.2). Introduisons premièrement une variable tensorielle $\boldsymbol{y}$ sur $\Omega_h^{\Gamma_N}$, telle que $\boldsymbol{y} = -\boldsymbol{\sigma}(\boldsymbol{u})$. Pour imposer $\boldsymbol{y}\boldsymbol{n} = -\boldsymbol{g}$ sur Γ_N, on rappelle que la normale extérieure unitaire $\boldsymbol{n}$ est donnée sur Γ par $\boldsymbol{n} = \frac{1}{|\nabla\varphi|}\nabla\varphi$. Ainsi, les conditions de Neumann sont imposées en introduisant une seconde variable auxiliaire (vectorielle), telle que $\boldsymbol{y}\nabla\varphi + \boldsymbol{g}|\nabla\varphi| = -\boldsymbol{p}_N\varphi$ sur $\Omega_h^{\Gamma_N}$. Alors, on obtient les équations suivantes :

$$\boldsymbol{y} + \boldsymbol{\sigma}(\boldsymbol{u}) = 0, \quad \text{sur } \Omega_h^{\Gamma_N}, \tag{2.34a}$$

$$\boldsymbol{y}\nabla\varphi + \boldsymbol{p}_N\varphi = -\boldsymbol{g}|\nabla\varphi|, \quad \text{sur } \Omega_h^{\Gamma_N}. \tag{2.34b}$$

Il ne reste alors plus qu'à introduire les espaces éléments finis permettant de discrétiser les différentes variables auxiliaires avant de construire le schéma. Une nouvelle fois, on pose $k \geqslant 1$, et on considère l'espace $\boldsymbol{V}_h$ défini par (2.26) comme espace de discrétisation pour l'approximation $\boldsymbol{u}_h$ de $\boldsymbol{u}$. Les variables $\boldsymbol{p}_{h,D}$ et $\boldsymbol{p}_{h,N}$, approximations de $\boldsymbol{p}_D$ et $\boldsymbol{p}_N$, seront choisies dans $\boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_N})$ et $\boldsymbol{Q}_h^k(\Omega_h^{\Gamma_D})$ respectivement (où $\boldsymbol{Q}_h^k(M_h)$ est défini par (2.27)).

Enfin, la variable $\boldsymbol{y}$ sera approchée par une variable $\boldsymbol{y}_h \in Z_h(\Omega_h^{\Gamma_N})$ où

$$\boldsymbol{Z}_h(M_h) := \{\boldsymbol{z}_h : M_h \to \mathbb{R}^{(d\times d)} : \boldsymbol{z}_{h|T} \in \mathbb{P}^k(T)^{(d\times d)} \ \ \forall T \in \mathcal{M}_h, \\ \boldsymbol{z}_h \text{ continue sur } M_h\}. \tag{2.35}$$

Finalement, on obtient le schéma : trouver $\boldsymbol{u}_h \in \boldsymbol{V}_h$, $\boldsymbol{p}_{h,D} \in \boldsymbol{Q}_h^k(\Omega_h^{\Gamma_D})$, $\boldsymbol{y}_h \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_N})$ et $\boldsymbol{p}_{h,N} \in \boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_N})$ tels que

$$\begin{aligned}
\int_{\Omega_h} \boldsymbol{\sigma}(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h - \int_{\partial\Omega_h \setminus \partial\Omega_{h,N}} \boldsymbol{\sigma}(\boldsymbol{u}_h)\boldsymbol{n} \cdot \boldsymbol{v}_h + \int_{\partial\Omega_{h,N}} \boldsymbol{y}_h \boldsymbol{n} \cdot \boldsymbol{v}_h \\
+ \gamma_u \int_{\Omega_h^{\Gamma_N}} (\boldsymbol{y}_h + \boldsymbol{\sigma}(\boldsymbol{u}_h)) : (\boldsymbol{z}_h + \boldsymbol{\sigma}(\boldsymbol{v}_h)) \\
+ \frac{\gamma_p}{h^2} \int_{\Omega_h^{\Gamma_N}} \left(\boldsymbol{y}_h \nabla \varphi_h + \frac{1}{h} \boldsymbol{p}_{h,N} \varphi_h \right) \cdot \left(\boldsymbol{z}_h \nabla \varphi_h + \frac{1}{h} \boldsymbol{q}_{h,N} \varphi_h \right) \\
+ \frac{\gamma}{h^2} \int_{\Omega_h^{\Gamma_D}} (\boldsymbol{u}_h - \frac{1}{h} \varphi_h \boldsymbol{p}_{h,D}) \cdot (\boldsymbol{v}_h - \frac{1}{h} \varphi_h \boldsymbol{q}_{h,D}) + G_h(\boldsymbol{u}_h, \boldsymbol{v}_h) \\
+ J_h^{lhs,D}(\boldsymbol{u}_h, \boldsymbol{v}_h) + J_h^{lhs,N}(\boldsymbol{y}_h, \boldsymbol{z}_h) = \int_{\Omega_h} \boldsymbol{f} \cdot \boldsymbol{v}_h \\
+ \frac{\gamma}{h^2} \int_{\Omega_h^D} \boldsymbol{u}_h^g \cdot (\boldsymbol{v}_h - \frac{1}{h} \varphi_h \boldsymbol{q}_{h,D}) - \frac{\gamma_p}{h^2} \int_{\Omega_h^{\Gamma_N}} \boldsymbol{g} \cdot |\nabla \varphi_h| (\boldsymbol{z}_h \cdot \nabla \varphi_h + \frac{1}{h} \boldsymbol{q}_{h,N} \varphi_h) \\
+ J_h^{rhs,D}(\boldsymbol{v}_h) + J_h^{rhs,N}(\boldsymbol{z}_h) \\
\forall \boldsymbol{v}_h \in \boldsymbol{V}_h, \boldsymbol{q}_{h,D} \in \boldsymbol{Q}_h^k(\Omega_h^{\Gamma_D}), \boldsymbol{z}_h \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_N}), \boldsymbol{q}_{h,N} \in \boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_N}) \,, \quad (2.36)
\end{aligned}$$

où G_h est définie par :

$$\begin{aligned}
G_h(\boldsymbol{u}, \boldsymbol{v}) := \sigma_D h \sum_{E \in \mathcal{F}_h^{\Gamma_D}} \int_E [\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n}] \cdot [\boldsymbol{\sigma}(\boldsymbol{v})\boldsymbol{n}] \\
+ \sigma_N h \sum_{E \in \mathcal{F}_h^{\Gamma_{N_s}}} \int_E [\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n}] \cdot [\boldsymbol{\sigma}(\boldsymbol{v})\boldsymbol{n}] \,,
\end{aligned}$$

avec $\mathcal{F}_h^{\Gamma_D}$ l'ensemble des facettes de $\Omega_h^{\Gamma_D}$ et $\mathcal{F}_h^{\Gamma_{N_s}}$ les facettes de $(\mathcal{T}_h \setminus \mathcal{T}_h^\Gamma) \cap \mathcal{T}_h^{\Gamma_N}$ (voir Figures 2.11 et 2.12 pour des exemples de représentations graphiques). Les termes de stabilisation J_h^{lhs} et J_h^{rhs} sont eux adaptés de (2.29) et (2.30), séparés en termes agissant sur $\boldsymbol{u}_h$ sur les cellules de la partie Dirichlet (et d'interface) de $\mathcal{T}_h^\Gamma$, et les termes agissant sur $\boldsymbol{y}_h$ sur la partie Neumann :

$$J_h^{lhs,D}(\boldsymbol{u}, \boldsymbol{v}) = \sigma_D h^2 \sum_{T \in \mathcal{T}_h^\Gamma \setminus \mathcal{T}_h^{\Gamma_N}} \int_T \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{v}) \,, \quad (2.37)$$

$$J_h^{rhs,D}(\boldsymbol{v}) = -\sigma_D h^2 \sum_{T \in \mathcal{T}_h^\Gamma \setminus \mathcal{T}_h^{\Gamma_N}} \int_T \boldsymbol{f} \cdot \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{v}) \,, \quad (2.38)$$

$$J_h^{lhs,N}(\boldsymbol{y}, \boldsymbol{z}) = \gamma_{div} \int_{\Omega_h^{\Gamma_N}} \operatorname{div} \boldsymbol{y} \cdot \operatorname{div} \boldsymbol{z} \,, \quad (2.39)$$

$$J_h^{rhs,N}(\boldsymbol{z}) = \gamma_{div} \int_{\Omega_h^{\Gamma_N}} \boldsymbol{f} \cdot \operatorname{div} \boldsymbol{z}. \quad (2.40)$$

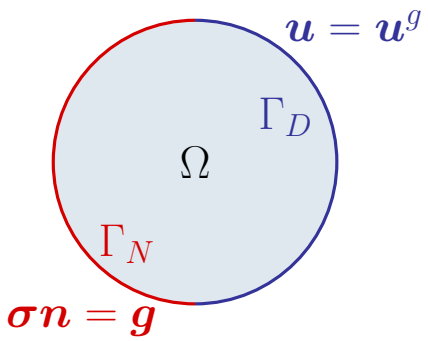

FIGURE 2.30 – Représentation de la géométrie considérée pour les cas test 1 ($\Gamma = \Gamma_D$) et 2.

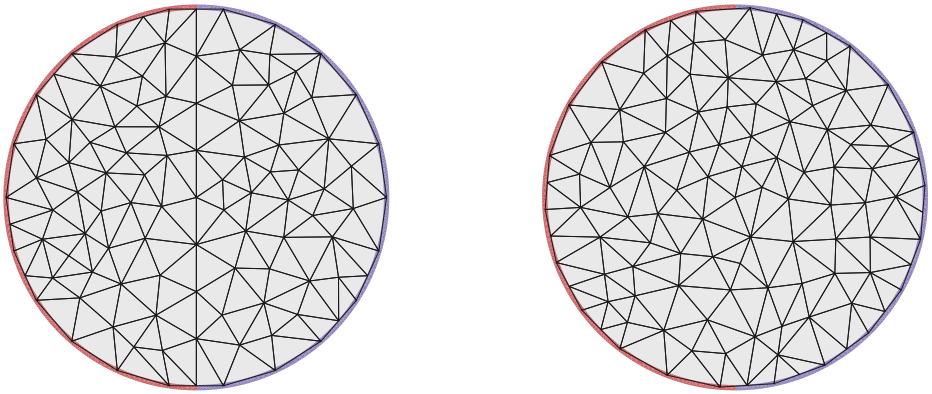

FIGURE 2.31 – **Cas test 2**. Maillages Standard-FEM. Gauche : changement de conditions de bord conforme. Droite : changement de conditions de bord non-conforme.

Cas test 2. Nous allons maintenant présenter des résultats numériques pour la méthode (2.36), que nous comparerons à la méthode standard FEM.

Remarque 2.16. Les ordres de convergence optimaux sont ici 3 pour la norme L^2 et 2 pour la semi-norme H^1 puisque l'on se place dans la situation d'éléments finis $\mathbb{P}^2$, en considérant une solution manufacturée au moins H^2, et donc très régulière.

Pour ce cas test, nous considérerons la géométrie définie par (2.32) (i.e. le cercle centré en $(0.5, 0.5)$, de rayon $\sqrt{2}/4$), les mêmes paramètres d'élasticité ainsi que la même solution manufacturée (2.33) que pour le premier cas test de cette section. Des conditions de Dirichlet seront imposées sur $\Gamma \cap \{x \geqslant 0.5\}$ et des conditions de Neumann pour $x < 0.5$, c.f. Figure 2.30, i.e. $\psi(x, y) = 0.5 - x$. Les conditions de bord $\boldsymbol{u}^g$ et $\boldsymbol{g}$ sont calculées à

partir de la solution manufacturée $\boldsymbol{u}_{ex}$. Pour φ-FEM, elles sont étendues de Γ à $\Omega_h^{\Gamma_D}$ et $\Omega_h^{\Gamma_N}$ respectivement. Elles sont définies par :

$$\begin{cases} \boldsymbol{u}^g & = \boldsymbol{u}_{ex}(1+\varphi), & \text{sur } \Omega_h^\Gamma \cap \{x \geqslant 0.5\}, \\ \boldsymbol{g} & = \boldsymbol{\sigma}(\boldsymbol{u}_{ex}) \cdot \frac{\nabla\varphi}{\|\nabla\varphi\|} + \boldsymbol{u}_{ex}\varphi, & \text{sur } \Omega_h^\Gamma \cap \{x < 0.5\}. \end{cases}$$

Les expressions sont ici une nouvelle fois perturbées lorsque l'on s'éloigne de Γ pour s'approcher d'un cas plus réaliste où l'on ne disposerait des données que sur Γ. Les paramètres de stabilisation sont fixés à $\gamma_{div} = \gamma_u = \gamma_p = 1.0$, $\sigma = 0.01$ et $\gamma = \sigma_D = 20.0$. La solution $\boldsymbol{u}_{ex}$ ainsi que la solution éléments finis classique et la solution φ-FEM (en plus de sa projection sur un maillage conforme) sont représentées à la Figure 2.32.

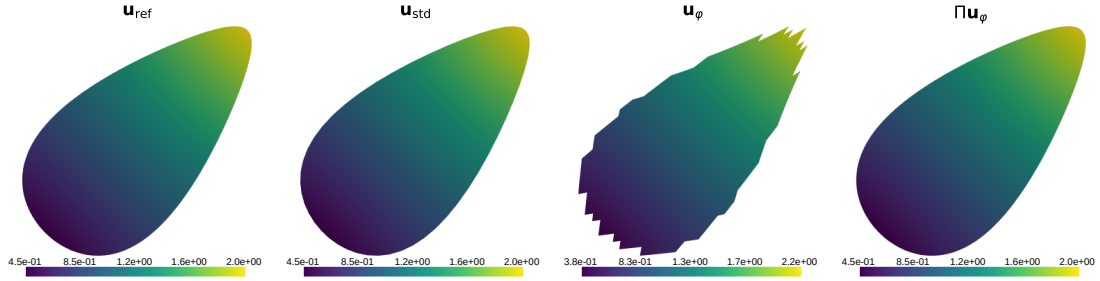

FIGURE 2.32 – **Cas test 2.** De gauche à droite : solution manufacturée sur un maillage fin, solution éléments finis, solution φ-FEM et projection sur un maillage conforme de la solution φ-FEM.

Comme nous l'avons fait pour le Cas test 3 de la Section 2.2.2, nous allons séparer l'étude numérique en deux cas : le cas *matching* et le cas *not matching*.

Cas de changement « conforme ». Commençons par étudier les cas où le changement de conditions de bord intervient sur des faces du maillage $\mathcal{T}_h^\Gamma$, que l'on compare au cas où le changement intervient sur un nœud d'un maillage standard. Ce cas sera considéré comme un « changement de conditions de bord conforme » et correspond aux Figures 2.12 et 2.31 (gauche). Ici, pour φ-FEM toutes les cellules de $\mathcal{T}_h^\Gamma$ sont bien attribuées soit à $\mathcal{T}_h^{\Gamma_N}$ ou à $\mathcal{T}_h^{\Gamma_D}$, et il n'y a donc pas de cellules d'interface. Pour ce cas, les résultats obtenus par φ-FEM et Standard FEM, tous deux avec des éléments finis $\mathbb{P}^2$ pour $\boldsymbol{u}_h$, sont présentés à la Figure 2.33. Les erreurs relatives L^2 et H^1 en fonction de h sont représentées sur la partie gauche. On observe alors que les ordres de convergence optimaux sont atteints pour φ-FEM, tandis que la convergence en norme L^2 est sous-optimale pour Standard-FEM. Dans ce cas, φ-FEM est toujours plus précis que l'approche standard, en norme L^2 comme H^1. De plus, la Figure 2.33 (droite) illustre qu'à nouveau, pour un seuil d'erreur fixé, les résultats seront obtenus plus rapidement qu'avec une méthode standard.

Cas de changement « non conforme ». Considérons maintenant un cas moins artificiel concernant la jonction Dirichlet/Neumann, laquelle pouvant intervenir à l'intérieur

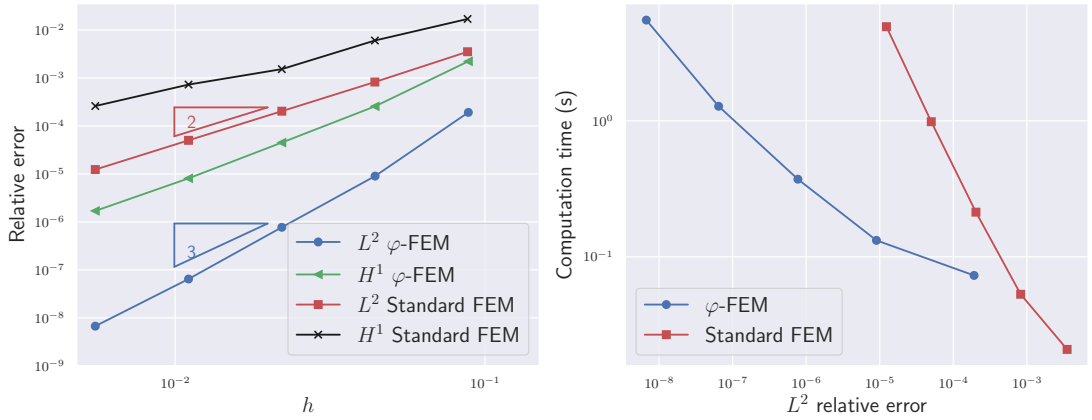

FIGURE 2.33 – **Cas test 2**. Cas de maillages avec changement de conditions de bord conforme. Gauche : erreurs relatives L^2 et H^1, en fonction des tailles de maillages. Droite : temps de calcul en fonction de l'erreur relative L^2.

d'une cellule de $\mathcal{T}_h^\Gamma$ ou d'une face du maillage conforme FEM standard. Ce cas correspond à la situation présentée aux Figures 2.11 et 2.31 (droite).

Les résultats numériques obtenus dans cette situation sont présentés à la Figure 2.34. En comparaison avec les résultats obtenus Figure 2.33, on observe que le comportement du schéma φ-FEM (2.36) n'est que très peu affecté par les cellules d'interface, puisque les courbes de convergence sont seulement légèrement moins lisses. En particulier, les conclusions faites précédemment sont toujours valables : la méthode φ-FEM est plus précise sur des maillages comparables et moins coûteuse en temps de calcul pour une erreur donnée que Standard-FEM.

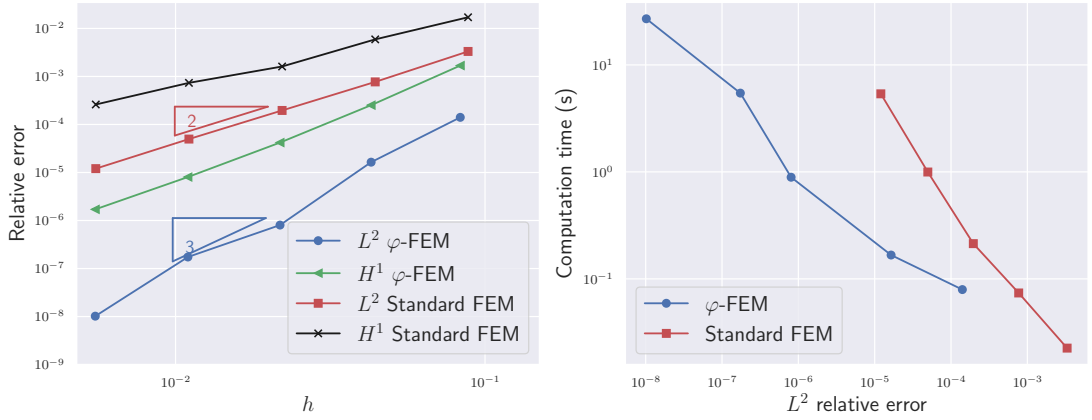

FIGURE 2.34 – **Cas test 2**. Cas de maillages avec changement de conditions de bord non-conforme. Gauche : erreurs relatives L^2 et H^1, en fonction des tailles de maillages. Droite : temps de calcul en fonction de l'erreur relative L^2.

2.4.2 Élasticité linéaire avec plusieurs matériaux.

Nous allons maintenant traiter le cas de problèmes avec interfaces, que nous modéliserons par une structure composée de deux matériaux, avec des paramètres d'élasticité différents. Cette situation a été traitée par les méthodes XFEM [18, 4, 92, 90], CutFEM [14, 43, 42, 53], et SBM [54]. Notre objectif est ici de démontrer l'applicabilité de notre approche dans ce contexte. Pour cela, supposons que la structure considérée occupe un domaine Ω, et est constituée de deux matériaux qui occupent des domaines Ω_1 et Ω_2, séparés par une interface Γ. On suppose de plus que le matériau Ω_1 est inclus dans le domaine Ω. Ainsi, $\Gamma = \partial\Omega_1$, comme illustré à la Figure 2.35. On suppose également que le déplacement $\boldsymbol{u}$ est donné à la frontière externe ($\partial\Omega$).
Le problème considéré est finalement de trouver $\boldsymbol{u}$ tel que

$$\begin{cases} -\operatorname{div}\boldsymbol{\sigma}(\boldsymbol{u}) &= \boldsymbol{f}\,,\ \text{sur}\ \ \Omega\backslash\Gamma\,, \\ \boldsymbol{u} &= \boldsymbol{u}^g\,,\ \text{sur}\ \ \partial\Omega\,, \\ [\boldsymbol{u}] &= 0\,,\ \text{sur}\ \ \Gamma\,, \\ [\boldsymbol{\sigma}(\boldsymbol{u})\cdot\boldsymbol{n}] &= 0\,,\ \text{sur}\ \ \Gamma\,, \end{cases} \tag{2.41}$$

où $\boldsymbol{n}$ est la normale unitaire de Ω_1 vers Ω_2, et $[\cdot]$ est le saut sur Γ. Les paramètres d'élasticité sont supposés constants sur chaque domaine, mais différents entre les deux domaines. Le tenseur des contraintes est donné par

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \begin{cases} \boldsymbol{\sigma}_1(\boldsymbol{u}) = 2\mu_1\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda_1(\operatorname{div}\boldsymbol{u})I\,,\ \text{sur}\ \Omega_1\,, \\ \boldsymbol{\sigma}_2(\boldsymbol{u}) = 2\mu_2\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda_2(\operatorname{div}\boldsymbol{u})I\,,\ \text{sur}\ \Omega_2\,, \end{cases}$$

avec les paramètres de Lamé λ_i et μ_i définis par (2.25), avec E_i, ν_i, $i = 1, 2$ donnés. En introduisant les déplacements $\boldsymbol{u}_i = \boldsymbol{u}|_{\Omega_i}$, $i = 1, 2$ sur Ω_1 et Ω_2 séparément, le problème (2.41) peut être réécrit sous la forme de deux problèmes couplés :

$$\begin{cases} -\operatorname{div}\boldsymbol{\sigma}_i(\boldsymbol{u}_i) &= \boldsymbol{f}\,,\ \text{sur}\ \ \Omega_i\,,\ i = 1, 2, \\ \boldsymbol{u}_2 &= \boldsymbol{u}^g\,,\ \text{sur}\ \ \partial\Omega\,, \\ \boldsymbol{u}_1 &= \boldsymbol{u}_2\,,\ \text{sur}\ \ \Gamma\,, \\ \boldsymbol{\sigma}_1(\boldsymbol{u}_1)\boldsymbol{n} &= \boldsymbol{\sigma}_2(\boldsymbol{u}_2)\boldsymbol{n}\,,\ \text{sur}\ \ \Gamma\,. \end{cases} \tag{2.42}$$

Supposons que le domaine Ω ait une forme suffisamment simple, de sorte qu'un maillage conforme $\mathcal{T}_h$ soit simple à générer précisément, par exemple un carré.

Remarque 2.17. Cette condition n'est pas particulièrement restrictive. En effet, dans le cas d'une géométrie complexe, il sera possible de traiter les conditions de bord de Ω à l'aide de φ-FEM.

Cependant, on suppose que le maillage $\mathcal{T}_h$ n'est pas conforme à l'interface Γ. Nous allons maintenant adapter la méthode φ-FEM à une telle situation. Le point de départ de cette nouvelle version est la réécriture du problème sous la forme (2.42). Ainsi, nous allons discrétiser séparément $\boldsymbol{u}_1$ dans Ω_1 et $\boldsymbol{u}_2$ dans Ω_2. Pour cela, commençons par introduire deux maillages actifs $\mathcal{T}_{h,1}$ et $\mathcal{T}_{h,2}$, sous-maillages de $\mathcal{T}_h$, construits de sorte que $\mathcal{T}_{h,i}$

contienne les cellules de $\mathcal{T}_h$ en intersection avec Ω_i. En pratique, ces deux sous-maillages sont définis par une fonction level-set φ :

$$\Omega_1 = \{\varphi > 0\} \cap \Omega, \qquad \Omega_2 = \{\varphi < 0\}, \qquad \Gamma = \{\varphi = 0\} \cap \Omega\,,$$

et ainsi $\mathcal{T}_{h,i}$, peuvent être construits en utilisant une interpolation φ_h de φ, par :

$$\mathcal{T}_{h,1} := \{T \in \mathcal{T}_h : T \cap \{\varphi_h > 0\} \neq \emptyset\} \text{ et } \mathcal{T}_{h,2} := \{T \in \mathcal{T}_h : T \cap \{\varphi_h < 0\} \neq \emptyset\}\,. \quad (2.43)$$

Le sous-maillage $\mathcal{T}_h^\Gamma$ est défini comme l'intersection $\mathcal{T}_{h,1} \cap \mathcal{T}_{h,2}$ et $\Omega_{h,1}$, $\Omega_{h,2}$, Ω_h^Γ sont les domaines couvrant les maillages $\mathcal{T}_{h,1}$, $\mathcal{T}_{h,2}$, $\mathcal{T}_h^\Gamma$ respectivement.

Les inconnues $\boldsymbol{u}_1$ et $\boldsymbol{u}_2$ seront discrétisées sur les domaines $\Omega_{h,1}$ et $\Omega_{h,2}$, en introduisant des extensions sur les parties additionnelles proches de Γ. Pour traiter cette situation, plusieurs variables auxiliaires seront nécessaires, proches de l'interface, i.e. sur Ω_h^Γ.

En prolongeant $\boldsymbol{u}_i$ aux domaines Ω_h^i, on peut alors écrire une formulation faible au niveau continu, donnée par :

$$\int_{\Omega_{h,i}} \boldsymbol{\sigma}_i(\boldsymbol{u}_i) : \nabla \boldsymbol{v}_i - \int_{\partial\Omega_{h,i}} \boldsymbol{\sigma}_i(\boldsymbol{u}_i)\boldsymbol{n}_i \cdot \boldsymbol{v}_i = \int_{\Omega_{h,i}} \boldsymbol{f} \cdot \boldsymbol{v}_i,$$
$$\forall \boldsymbol{v}_i \text{ sur } \Omega_{h_i} \text{ tel que } \boldsymbol{v}_i = \boldsymbol{0} \text{ sur } \partial\Omega\,. \quad (2.44)$$

Par la suite, par abus de notation la partie de la frontière de $\Omega_{h,i}$ autre que $\partial\Omega$ sera notée $\partial\Omega_{h,i}$ et $\boldsymbol{n}_i$ correspondra à la normale unitaire sur $\partial\Omega_{h,i}$ extérieure à $\Omega_{h,i}$. Les conditions de bord sur $\partial\Omega$, i.e. la deuxième équation dans (2.42), seront imposées fortement. Les autres conditions, sur l'interface Γ seront imposées via φ-FEM, en introduisant des variables auxiliaires sur Ω_h^Γ : la variable vectorielle $\boldsymbol{p}$ (similaire à celle introduite pour les conditions de Dirichlet précédemment) et les variables tensorielles $\boldsymbol{y}_1$ et $\boldsymbol{y}_2$ (similaires à la variable y introduite pour les conditions de Neumann). Cela donne alors (cf. les deux dernières équations de (2.42)) :

$$\boldsymbol{u}_1 - \boldsymbol{u}_2 + \boldsymbol{p}\varphi = 0\,, \quad \text{sur } \Omega_h^\Gamma, \quad (2.45)$$
$$\boldsymbol{y}_i + \boldsymbol{\sigma}_i(\boldsymbol{u}_i) = 0\,, \quad \text{sur } \Omega_h^\Gamma,\ i = 1, 2, \quad (2.46)$$
$$\boldsymbol{y}_1 \nabla\varphi - \boldsymbol{y}_2 \nabla\varphi = 0\,, \quad \text{sur } \Omega_h^\Gamma\,. \quad (2.47)$$

L'équation (2.47) prolonge la dernière équation de (2.42) de l'interface Γ au domaine Ω_h^Γ.

Discrétisons maintenant les équations (2.44)–(2.47).
Pour cela, on considérera $k \geqslant 1$, et

$$\boldsymbol{V}_{h,i} := \{\boldsymbol{v}_h : \Omega_{h,i} \to \mathbb{R}^d : \boldsymbol{v}_{h|T} \in \mathbb{P}^k(T)^d \ \ \forall T \in \mathcal{T}_h,$$
$$\boldsymbol{v}_h \text{ continue sur } \Omega_{h,i}\,, \text{ et } \boldsymbol{v}_h = I_h \boldsymbol{u}^g \ \text{ sur } \partial\Omega\} \quad (2.48)$$

où I_h est l'interpolant éléments finis classique, ainsi que les versions homogènes correspondantes : $\boldsymbol{V}_{h,i}^0$ avec la contrainte $\boldsymbol{v}_h = \boldsymbol{0}$ sur $\partial\Omega$, espaces utilisés pour les fonctions test.

De plus, on considérera les espaces $\boldsymbol{Q}_h^{(k)}(\Omega_h^\Gamma)$ et $\boldsymbol{Z}_h(\Omega_h^\Gamma)$ définis respectivement par (2.27) et (2.35) pour discrétiser les variables auxiliaires. En combinant (2.44) avec (2.45)–(2.47) introduites sous la forme des moindres carrés, on obtient le schéma suivant : trouver $\boldsymbol{u}_{h,1} \in \boldsymbol{V}_{h,1}$, $\boldsymbol{u}_{h,2} \in \boldsymbol{V}_{h,2}$, $\boldsymbol{p}_h \in \boldsymbol{Q}_h^k(\Omega_h^\Gamma)$, $\boldsymbol{y}_{h,1}, \boldsymbol{y}_{h,2} \in \boldsymbol{Z}_h(\Omega_h^\Gamma)$ tels que,

$$\begin{aligned}
&\sum_{i=1}^{2} \int_{\Omega_{h,i}} \boldsymbol{\sigma}_i(\boldsymbol{u}_{h,i}) : \nabla \boldsymbol{v}_{h,i} + \sum_{i=1}^{2} \int_{\partial\Omega_{h,i}} \boldsymbol{y}_{h,i}\boldsymbol{n} \cdot \boldsymbol{v}_h \\
&\qquad + \frac{\gamma_p}{h^2} \int_{\Omega_h^\Gamma} (\boldsymbol{u}_{h,1} - \boldsymbol{u}_{h,2} + \frac{1}{h}\boldsymbol{p}_h\varphi_h) \cdot (\boldsymbol{v}_{h,1} - \boldsymbol{v}_{h,2} + \frac{1}{h}\boldsymbol{q}_h\varphi_h) \\
&\qquad\qquad + \gamma_u \sum_{i=1}^{2} \int_{\Omega_h^\Gamma} (\boldsymbol{y}_{h,i} + \boldsymbol{\sigma}_i(\boldsymbol{u}_{h,i})) : (\boldsymbol{z}_{h,i} + \boldsymbol{\sigma}_i(\boldsymbol{v}_{h,i})) \\
&\qquad + \frac{\gamma_y}{h^2} \int_{\Omega_h^\Gamma} (\boldsymbol{y}_{h,1}\nabla\varphi_h - \boldsymbol{y}_{h,2}\nabla\varphi_h) \cdot (\boldsymbol{z}_{h,1}\nabla\varphi_h - \boldsymbol{z}_{h,2}\nabla\varphi_h) \\
&+ \sum_{i=1}^{2} \Big(G_h(\boldsymbol{u}_{h,i}, \boldsymbol{v}_{h,i}) + J_h^{lhs,N}(\boldsymbol{y}_{h,i}, \boldsymbol{z}_{h,i}) \Big) = \sum_{i=1}^{2} \int_{\Omega_{h,i}} \boldsymbol{f} \cdot \boldsymbol{v}_{h,i} + \sum_{i=1}^{2} J_h^{rhs,N}(\boldsymbol{z}_{h,i}) \,, \\
&\qquad\qquad \forall\, \boldsymbol{v}_{h,1} \in \boldsymbol{V}_{h,1}^0, \boldsymbol{v}_{h,2} \in \boldsymbol{V}_{h,2}^0, \boldsymbol{q}_h \in \boldsymbol{Q}_h^k(\Omega_h^\Gamma), \boldsymbol{z}_{h,1}, \boldsymbol{z}_{h,2} \in \boldsymbol{Z}_h(\Omega_h^\Gamma) \,. \quad (2.49)
\end{aligned}$$

Comme précédemment, les termes de stabilisation ont été ajoutés, avec G_h défini par (2.28) et $J_h^{rhs,N}$ par (2.40) avec $\Omega_h^{\Gamma_N}$ remplacé par Ω_h^Γ et en imposant $\operatorname{div} \boldsymbol{y}_i = \boldsymbol{f}$ sur Ω_h^Γ à la manière des moindres carrés.

Cas test 3. On considère $\Omega = (0,1)^2$ et Ω_1, Ω_2 définis par φ

$$\varphi(x,y) = -R^2 + (x-0.5)^2 + (y-0.5)^2 \,,$$

avec $R = 0.3$ comme illustré à la Figure 2.35. Une nouvelle fois, pour calculer l'erreur, nous utiliserons une solution manufacturée, définie par

$$\boldsymbol{u} = \boldsymbol{u}_{ex} = \begin{cases} \frac{1}{E_1}(\cos(r) - \cos(R))(1,1)^T & \text{si } r < R, \\ \frac{1}{E_2}(\cos(r) - \cos(R))(1,1)^T & \text{sinon,} \end{cases}$$

où $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. On détermine alors $\boldsymbol{f}$ de manière analytique et on impose $\boldsymbol{u}_g = \boldsymbol{u}_{ex}$ sur $\partial\Omega$.

Les paramètres d'élasticité sont donnés par $E_1 = 7$, $E_2 = 2.28$ et $\nu_1 = \nu_2 = 0.3$. Une représentation des maillages considérés pour φ-FEM et pour Standard-FEM est donnée à la Figure 2.36.

Pour la méthode standard, la solution $\boldsymbol{u}_h \in V_h$ est obtenue grâce au schéma

$$\sum_{i=1}^{2} \int_{\Omega_{h,i}} \boldsymbol{\sigma}_i(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}_h, \ \forall\ \boldsymbol{v}_h \in \boldsymbol{V}_h^0 \,, \quad (2.50)$$

où $\boldsymbol{V}_h$ est l'espace éléments finis de degré $\mathbb{P}^k$ approchant $\boldsymbol{u}_g$ sur $\partial\Omega$ et $\boldsymbol{V}_h^0$ est son analogue homogène.

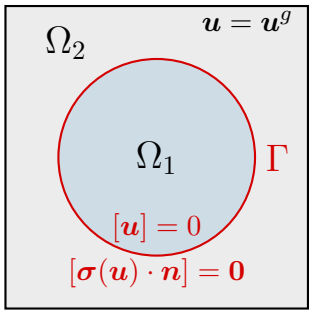

FIGURE 2.35 – Représentation de la géométrie considérée pour le cas test 3.

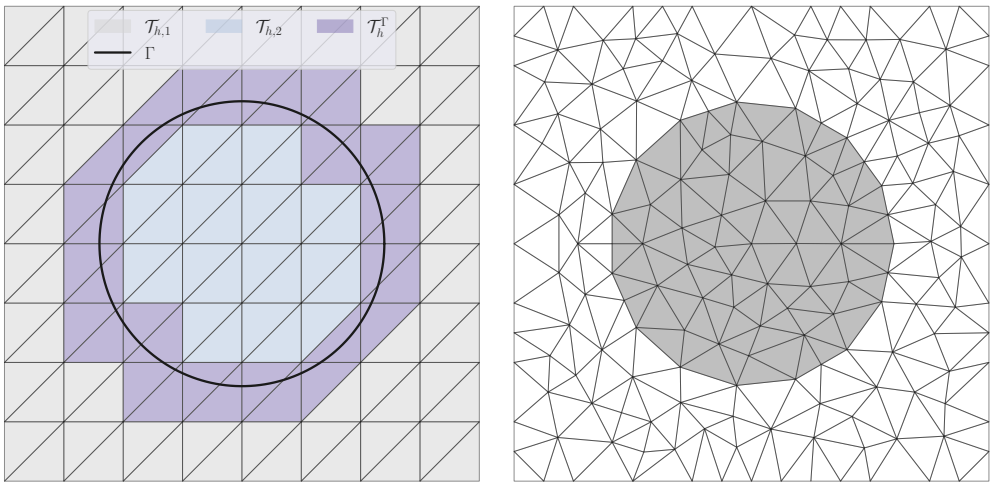

FIGURE 2.36 – **Cas test 3 : problème d'interface.** Gauche : maillage φ-FEM, avec $\mathcal{T}_h^\Gamma$ représenté en violet. Droite : Maillage standard, conforme à l'interface Γ.

Les résultats obtenus avec le schéma φ-FEM (2.49) et la méthode standard (2.50), pour des éléments finis $\mathbb{P}^2$ sont présentés à la Figure 2.38. On représente également à la figure 2.37 les déplacements obtenus par les deux méthodes ainsi que la solution de référence et la projection sur un maillage conforme de la solution φ-FEM. Les conclusions sont une nouvelle fois les mêmes que pour les deux cas test précédents : dans ce cas, φ-FEM est plus précise que la méthode standard sur maillages de tailles comparables et moins coûteuse en temps de calcul pour une erreur fixée.

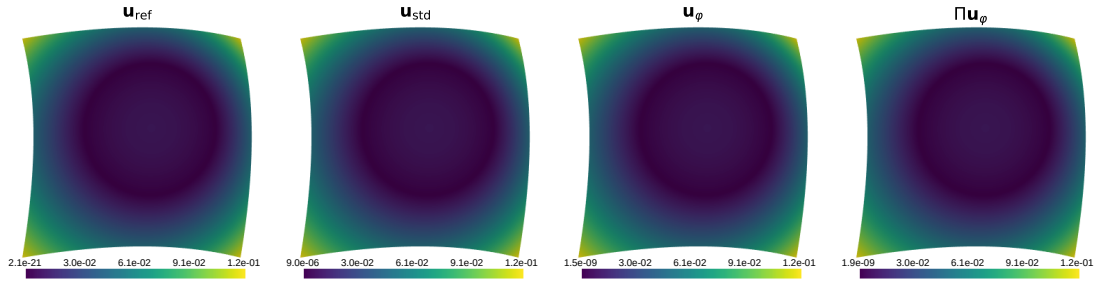

FIGURE 2.37 – **Cas test 3 : problème d'interface.** De gauche à Droite : solution manufacturée sur un maillage fin, solution éléments finis, solution φ-FEM et projection sur un maillage conforme de la solution φ-FEM.

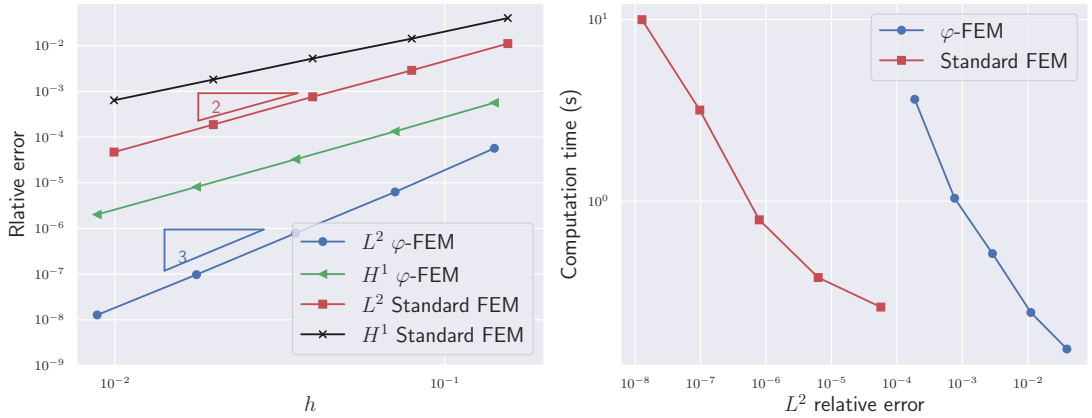

FIGURE 2.38 – **Cas test 3 : problème d'interface.** Gauche : erreurs relatives H^1 et L^2 en fonction de la taille de maillage. Droite : temps de calcul en fonction de l'erreur relative L^2.

2.4.3 Problèmes avec des fractures

Considérons maintenant le cas d'un problème d'élasticité linéaire posé sur un domaine avec une fracture, $\Omega \setminus \Gamma_f$ où Γ_f est une fracture (une courbe en 2D, une surface en 3D) à l'intérieur du domaine Ω :

$$\begin{cases} -\operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}) & = \boldsymbol{f}\,, \text{ sur } \Omega \setminus \Gamma_f\,, \\ \boldsymbol{u} & = \boldsymbol{u}^g\,, \quad \text{sur } \partial\Omega\,, \\ \boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} & = \boldsymbol{g}\,, \quad \text{sur } \Gamma_f\,. \end{cases} \tag{2.51}$$

Ce type de problème est le domaine d'application original de la méthode XFEM, cf. [69]. Notre objectif va être d'adapter l'approche φ-FEM à ce type de problème.

En pratique, la géométrie de la fracture sera donnée par une première fonction level-set φ (qui permettra de localiser la fracture en 2D et sa surface en 3D) et une seconde level-set ψ qui localisera les extrémités de cette fracture :

$$\Gamma_f := \Omega \cap \{\varphi = 0\} \cap \{\psi < 0\}\,.$$

Supposons que la courbe (surface) $\Gamma := \{\varphi = 0\}$ sépare Ω en deux sous-domaines Ω_1 et Ω_2, caractérisés respectivement par $\{\varphi < 0\}$ et $\{\varphi > 0\}$, comme représenté à la Figure 2.39. L'interface Γ consiste alors en Γ_f et la partie (fictive) restante, Γ_{int} :

$$\Gamma_{int} := \Omega \cap \{\varphi = 0\} \cap \{\psi > 0\}\,.$$

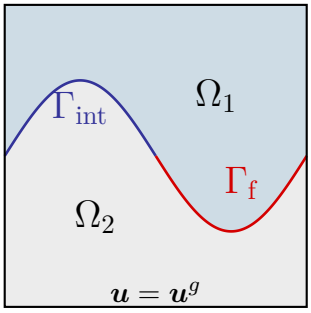

FIGURE 2.39 – Représentation de la géométrie considérée pour le cas test 4.

Afin, de réutiliser le schéma φ-FEM (2.49) introduit précédemment pour les problèmes d'interface, on peut reformuler le problème (2.51) en utilisant deux inconnues $\boldsymbol{u}_i = \boldsymbol{u}|_{\Omega_i}$, $i = 1, 2$, sous la forme :

$$\begin{cases} -\operatorname{div}\boldsymbol{\sigma}(\boldsymbol{u}_i) &= \boldsymbol{f}\,, \quad \text{sur} \quad \Omega_i\,, \\ \boldsymbol{u}_i &= \boldsymbol{u}^g\,, \ \text{sur} \quad \partial\Omega\,, \\ [\boldsymbol{u}] &= 0\,, \ \text{sur} \quad \Gamma_{int}\,, \\ [\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n}] &= 0\,, \ \text{sur} \quad \Gamma_{int}\,, \\ \boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} &= \boldsymbol{g}\,, \quad \text{sur} \quad \Gamma_f\,. \end{cases} \tag{2.52}$$

On suppose que la géométrie est suffisamment simple pour être maillée de façon conforme par un maillage cartésien $\mathcal{T}_h$, ce dernier n'étant pas conforme à l'interface Γ. Les variables $\boldsymbol{u}_1$ et $\boldsymbol{u}_2$ seront discrétisées séparément dans Ω_1 et Ω_2, en utilisant la forme (2.52) comme point de départ. On introduit alors une nouvelle fois deux sous-maillages $\mathcal{T}_{h,1}$ et $\mathcal{T}_{h,2}$, donnés par (2.43). De plus, on introduit un maillage sur l'interface, donné par $\mathcal{T}_h^\Gamma = \mathcal{T}_{h,1} \cap \mathcal{T}_{h,2}$, séparé lui en deux sous-maillages en fonction de ψ :

$$\mathcal{T}_h^{\Gamma_f} := \{T \in \mathcal{T}_h^\Gamma : \psi \leqslant 0 \text{ sur } T\} \qquad \text{et} \qquad \mathcal{T}_h^{\Gamma_{int}} := \{T \in \mathcal{T}_h^\Gamma : \psi \geqslant 0 \text{ sur } T\}\,.$$

Comme dans le cas des conditions mixtes Dirichlet/Neumann, cette définition peut laisser quelques cellules n'appartenant à aucun des sous-maillages ou aux deux, comme illustré à la Figure 2.40, où les cellules de $\mathcal{T}_h^{\Gamma_f}$ et $\mathcal{T}_h^{\Gamma_{int}}$ sont représentées respectivement en rouge et en bleu. La situation décrite est représentée à la Figure 2.40 (droite) où les cellules roses représentent les cellules restantes. Ces cellules correspondent aux cellules en intersection avec la droite $\{\psi = 0\}$, correspondant à la droite caractérisant l'extrémité interne de la fracture.

Maintenant que les différents maillages sont définis, nous allons pouvoir construire le schéma φ-FEM correspondant. Soit une nouvelle fois $k \geq 1$. On considère $\boldsymbol{V}_{h,1}$, $\boldsymbol{V}_{h,2}$ les espaces éléments finis de degré k sur $\mathcal{T}_{h,1}$ et $\mathcal{T}_{h,2}$, ainsi que les espaces homogènes correspondant, $\boldsymbol{V}^0_{h,1}$, $\boldsymbol{V}^0_{h,2}$ (cf. (2.48)) pour approcher $\boldsymbol{u}_1$ et $\boldsymbol{u}_2$. Ces espaces seront utilisés pour la discrétisation de la formulation variationnelle de la première équation de (2.52) et appliquer les conditions de bord sur $\partial\Omega$. Les équations restantes dans (2.52), i.e. les sauts sur Γ_{int} et les conditions de Neumann sur Γ_f seront traitées via φ-FEM et donc en introduisant des variables auxiliaires sur les parties appropriées de Ω_h^Γ (i.e. le domaine recouvrant le maillage $\mathcal{T}_h^\Gamma$) :

- les inconnues $\boldsymbol{p}$ et $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ sur $\Omega_h^{\Gamma_{int}}$ (domaine recouvrant $\mathcal{T}_h^{\Gamma_{int}}$) serviront à imposer la continuité du déplacement et des forces normales sur Γ_{int} avec les équations

$$\begin{aligned} \boldsymbol{u}_1 - \boldsymbol{u}_2 + \boldsymbol{p}\varphi = 0\,, \quad &\text{sur } \Omega_h^{\Gamma_{int}}\,, \\ \boldsymbol{y}_i = -\boldsymbol{\sigma}(\boldsymbol{u}_i)\,, \quad &\text{sur } \Omega_h^{\Gamma_{int}}\,, \\ \boldsymbol{y}_1 \cdot \nabla\varphi - \boldsymbol{y}_2 \cdot \nabla\varphi = 0\,, \quad &\text{sur } \Omega_h^{\Gamma_{int}}\,, \end{aligned}$$

  qui sont les mêmes que celles introduites dans (2.45)–(2.47), à la seule différence qu'elles ne sont imposées que sur une portion de Ω_h^Γ. Ces variables seront donc discrétisées dans les espaces $\boldsymbol{Q}_h^k(\Omega_h^{\Gamma_{int}})$ (cf. (2.27)) pour $\boldsymbol{p}$ et $\boldsymbol{Z}_h(\Omega_h^{\Gamma_{int}})$ (cf. (2.35)) pour $\boldsymbol{y}_1$,$\boldsymbol{y}_2$.

- les inconnues $\boldsymbol{p}_i^N$ et $\boldsymbol{y}_i^N$, $i = 1, 2$ définies sur $\Omega_h^{\Gamma_f}$ (domaine couvrant $\mathcal{T}_h^{\Gamma_f}$) serviront elles à imposer les conditions de Neumann sur Γ_f, avec les équations

$$\begin{aligned} \boldsymbol{y}_i^N = -\boldsymbol{\sigma}(\boldsymbol{u}_i)\,, \quad &\text{sur } \Omega_h^{\Gamma_f}\,, \\ \boldsymbol{y}_i^N \nabla\varphi + \boldsymbol{p}_i^N \varphi + \boldsymbol{g}|\nabla\varphi| = 0\,, \quad &\text{sur } \Omega_h^{\Gamma_f}\,, \end{aligned}$$

  les mêmes que (2.34)(a-b) cette fois seulement sur $\Omega_h^{\Gamma_f}$ au lieu de $\Omega_h^{\Gamma_N}$.
  Les variables $\boldsymbol{p}_i^N$ seront discrétisées dans $\boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_f})$ et $\boldsymbol{y}_i^N$ dans $\boldsymbol{Z}_h(\Omega_h^{\Gamma_f})$.

Remarque 2.18. Cette combinaison d'équations n'impose pas exactement les conditions d'interface sur l'ensemble de Γ puisque cette dernière peut ne pas être complètement couverte par $\Omega_h^{\Gamma_f} \cup \Omega_h^{\Gamma_{int}}$. Ce défaut dans la formulation continue, sera comblé dans la formulation discrète en introduisant les termes de stabilisation appropriés, comme nous avons pu le voir pour le cas des conditions mixtes Dirichlet/Neumann.

Tout cela donne finalement le schéma : trouver $\boldsymbol{u}_{h,1} \in \boldsymbol{V}_{h,1}$, $\boldsymbol{u}_{h,2} \in \boldsymbol{V}_{h,2}$, $\boldsymbol{p}_h \in \boldsymbol{Q}_h^k(\Omega_h^{\Gamma_{int}})$, $\boldsymbol{y}_{h,1}, \boldsymbol{y}_{h,2} \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_{int}})$, $\boldsymbol{p}_{h,1}^N, \boldsymbol{p}_{h,2}^N \in \boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_f})$, $\boldsymbol{y}_{h,1}^N, \boldsymbol{y}_{h,2}^N \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_f})$ tels

que

$$
\begin{aligned}
\sum_{i=1}^{2}\Big(\int_{\Omega_{h,i}}\boldsymbol{\sigma}(\boldsymbol{u}_{h,i}):\nabla\boldsymbol{v}_{h,i}+\int_{\partial\Omega_{h,i}^{int}}\boldsymbol{y}_{h,i}\boldsymbol{n}\cdot\boldsymbol{v}_{h,i}+\int_{\partial\Omega_{h,i}^{f}}\boldsymbol{y}_{h,i}^{N}\boldsymbol{n}\cdot\boldsymbol{v}_{h,i}\\
-\int_{\partial\Omega_{h,i}\setminus(\partial\Omega_{h,i,int}\cup\partial\Omega_{h,i,f})}\boldsymbol{\sigma}(\boldsymbol{u}_{h,i})\boldsymbol{n}\cdot\boldsymbol{v}_{h,i}\Big)\\
+\frac{\gamma_p}{h^2}\int_{\Omega_h^{\Gamma_{int}}}(\boldsymbol{u}_{h,1}-\boldsymbol{u}_{h,2}+\frac{1}{h}\boldsymbol{p}_h\varphi_h)\cdot(\boldsymbol{v}_{h,1}-\boldsymbol{v}_{h,2}+\frac{1}{h}\boldsymbol{q}_h\varphi_h)\\
+\gamma_u\sum_{i=1}^{2}\int_{\Omega_h^{\Gamma_{int}}}(\boldsymbol{y}_{h,i}+\boldsymbol{\sigma}(\boldsymbol{u}_{h,i})):(\boldsymbol{z}_{h,i}+\boldsymbol{\sigma}(\boldsymbol{v}_{h,i}))\\
+\frac{\gamma_y}{h^2}\int_{\Omega_h^{\Gamma_{int}}}(\boldsymbol{y}_{h,1}\nabla\varphi_h-\boldsymbol{y}_{h,2}\nabla\varphi_h)\cdot(\boldsymbol{z}_{h,1}\nabla\varphi_h-\boldsymbol{z}_{h,2}\nabla\varphi_h)\\
+\gamma_{u,N}\sum_{i=1}^{2}\int_{\Omega_h^{\Gamma_f}}(\boldsymbol{y}_{h,i}^{N}+\boldsymbol{\sigma}(\boldsymbol{u}_{h,i})):(\boldsymbol{z}_{h,i}^{N}+\boldsymbol{\sigma}(\boldsymbol{v}_{h,i}))\\
+\frac{\gamma_{p,N}}{h^2}\sum_{i=1}^{2}\int_{\Omega_h^{\Gamma_f}}(\boldsymbol{y}_{h,i}^{N}\nabla\varphi_h+\frac{1}{h}\boldsymbol{p}_{h,i}^{N}\varphi_h)\cdot(\boldsymbol{z}_{h,i}^{N}\nabla\varphi_h+\frac{1}{h}\boldsymbol{q}_{h,i}^{N}\varphi_h)\\
+\sum_{i=1}^{2}\Big(G_h\left(\boldsymbol{u}_{h,i},\boldsymbol{v}_{h,i}\right)+J_h^{lhs,int}\left(\boldsymbol{y}_{h,i},\boldsymbol{z}_{h,i}\right)+J_h^{lhs,f}\left(\boldsymbol{y}_{h,i}^{N},\boldsymbol{z}_{h,i}^{N}\right)\Big)\\
=\sum_{i=1}^{2}\int_{\Omega_{h,i}}\boldsymbol{f}\cdot\boldsymbol{v}_{h,i}-\frac{\gamma_{p,N}}{h^2}\sum_{i=1}^{2}\int_{\Omega_h^{\Gamma_f}}\boldsymbol{g}|\nabla\varphi_h|(\boldsymbol{z}_{h,i}^{N}\nabla\varphi_h+\frac{1}{h}\boldsymbol{q}_{h,i}^{N}\varphi_h)\\
+\sum_{i=1}^{2}\Big(J_h^{rhs,int}\left(\boldsymbol{z}_{h,i}\right)+J_h^{rhs,f}\left(\boldsymbol{z}_{h,i}^{N}\right)\Big),\\
\forall\boldsymbol{v}_{h,1}\in\boldsymbol{V}_{h,1}^{0},\boldsymbol{v}_{h,2}\in\boldsymbol{V}_{h,2}^{0},\boldsymbol{q}_h\in\boldsymbol{Q}_h^{k}(\Omega_h^{\Gamma_{int}}),\boldsymbol{z}_{h,1},\boldsymbol{z}_{h,2}\in\boldsymbol{Z}_h(\Omega_h^{\Gamma_{int}}),\\
\boldsymbol{q}_{h,1}^{N},\boldsymbol{q}_{h,2}^{N}\in\boldsymbol{Q}_h^{k-1}(\Omega_h^{\Gamma_f}),\boldsymbol{z}_{h,1}^{N},\boldsymbol{z}_{h,2}^{N}\in\boldsymbol{Z}_h(\Omega_h^{\Gamma_f})\,. \quad (2.53)
\end{aligned}
$$

Comme précédemment, G_h (2.28) a été ajoutée. De plus $J_h^{lhs,int}$, $J_h^{lhs,f}$ (ainsi que leurs analogues dans le second membre) ont été adaptés de $J_h^{lhs,N}$ (cf. (2.39)) pour correspondre aux bons sous-maillages :

$$
J_h^{lhs,int}(\boldsymbol{y},\boldsymbol{z})=\gamma_{div}\int_{\Omega_h^{\Gamma_{int}}}\operatorname{div}\boldsymbol{y}\cdot\operatorname{div}\boldsymbol{z}\,,\quad J_h^{lhs,f}(\boldsymbol{y},\boldsymbol{z})=\gamma_{div}\int_{\Omega_h^{\Gamma_f}}\operatorname{div}\boldsymbol{y}\cdot\operatorname{div}\boldsymbol{z}\,.
$$

Comme introduit pour les problèmes d'interface, nous avons ici noté $\partial\Omega_{h,i}$ les parties de frontières de $\Omega_{h,i}$ autres que $\partial\Omega$. De plus $\partial\Omega_{h,i}^{int}$, partie de $\partial\Omega_{h,i}$ est formée par les faces de $\mathcal{T}_{h,i}$ appartenant aux cellules de $\mathcal{T}_h^{\Gamma_{int}}$ et $\partial\Omega_{h,i}^{f}$ a été construit de la même façon.

Cas test 4. Soit $\Omega=(0,1)^2$ avec l'interface Γ donnée par la level-set

$$
\varphi(x,y)=y-\frac{1}{4}\sin(2\pi x)-\frac{1}{2}\,.
$$

L'extrémité interne de la fracture sera au point d'abscisse $x = 0.5$, ainsi

$$\Gamma_{int} := \{\varphi = 0\} \cap \{x < 0.5\} \qquad \text{et} \qquad \Gamma_f := \{\varphi = 0\} \ \cap \{x > 0.5\}\,.$$

Cette situation est représentée à la Figure 2.39.
Le schéma φ-FEM (2.53) est utilisé pour résoudre (2.51), avec la solution manufacturée

$$\boldsymbol{u} = \boldsymbol{u}_{ex} = (\sin(x) \times \exp(y), \sin(y) \times \exp(x))^T$$

définissant $\boldsymbol{f}$, $\boldsymbol{g}$, et $\boldsymbol{u}^g$.
Les forces $\boldsymbol{g}$ sur la fracture sont étendues à un voisinage de Γ_f, construit par

$$\boldsymbol{g} = \boldsymbol{\sigma}(\boldsymbol{u}_{ex})\frac{\nabla\varphi}{\|\nabla\varphi\|} + \varphi\boldsymbol{u}_{ex}\,.$$

Les paramètres de stabilisation sont fixés à $\gamma_u = \gamma_p = \gamma_{div} = \gamma_{u,N} = \gamma_{p,N} = \gamma_{div,N} = 1.0$, $\sigma_p = 1.0$ et $\sigma_D = 20.0$.

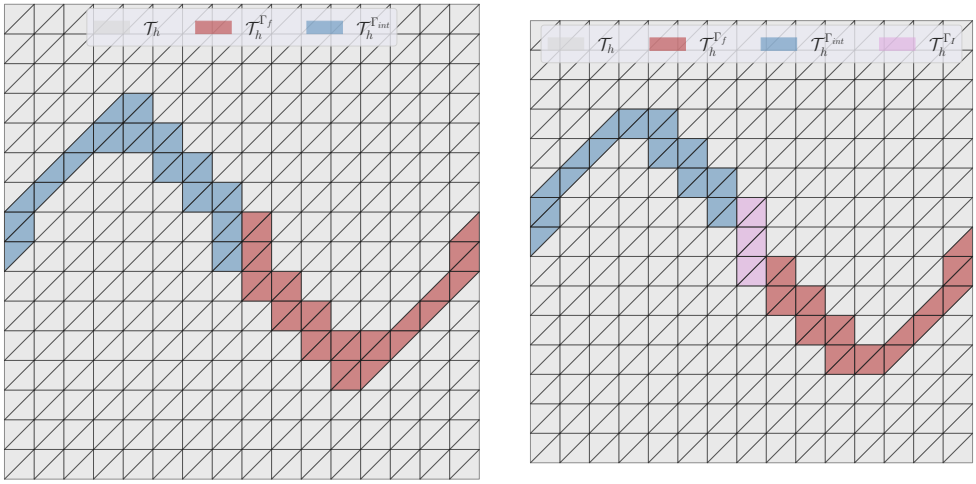

FIGURE 2.40 – **Cas test 4 : cas d'une fracture.** Maillages φ-FEM. Gauche : maillage « conforme » à l'extrémité de la fracture ; les cellules de $\mathcal{T}_h^{\Gamma_{int}}$ sont en bleu ; celles de $\mathcal{T}_h^{\Gamma_f}$ en rouge. Droite : maillage « non-conforme » à l'extrémité de la fracture ; en rose les cellules de l'interface entre $\mathcal{T}_h^{\Gamma_{int}}$ et $\mathcal{T}_h^{\Gamma_f}$.

Deux séries de simulations ont été réalisées pour étudier les résultats de φ-FEM (2.53), avec des éléments finis $\mathbb{P}^2$: premièrement pour le cas de la Figure 2.40 (gauche) où l'extrémité de la fracture intervient sur une face du maillage et deuxièmement, lorsque celle-ci est à l'intérieur d'une cellule du maillage, i.e. Figure 2.40 (droite).
Les résultats présentés Figure 2.41, indiquent que la convergence de φ-FEM est optimale, dans les deux situations.

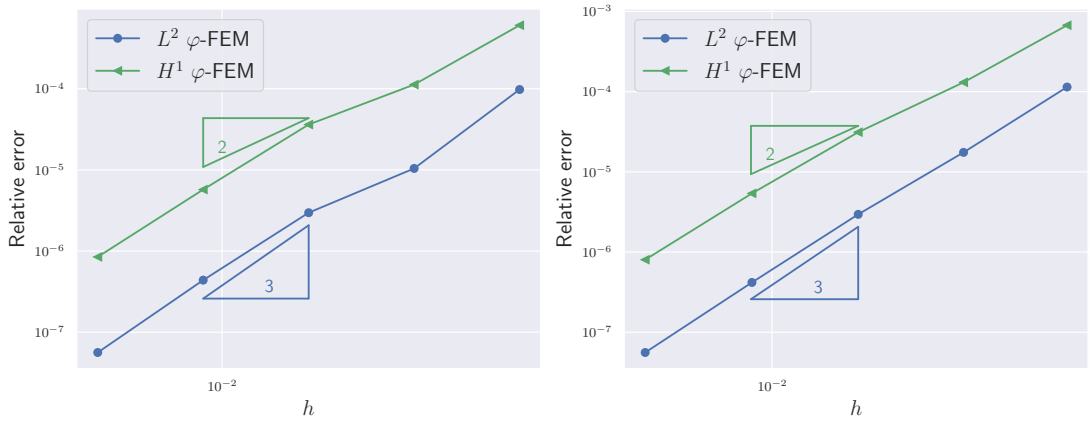

FIGURE 2.41 – **Cas test 4 : cas d'une fracture.** Erreurs relatives H^1 et L^2 en fonction de la taille de cellule. Gauche : maillages « conformes » à l'extrémité de la fracture. Droite : maillages « non-conformes » à l'extrémité de la fracture.

2.4.4 Nouveaux résultats pour des conditions mixtes

Dans les cas test précédents, nous avons étudié numériquement la convergence des schémas proposés. En particulier, pour le cas de l'équation (2.24), et donc du schéma (2.36), nous n'avons considéré que des solutions manufacturées. L'avantage de ces solutions est la facilité de calcul de l'erreur commise par les méthodes numériques. Cependant, comme nous l'avons vu à la Section 2.2, la plus grosse difficulté dans le cas de conditions mixtes est le traitement de la singularité de changement de conditions de bord. Or, les solutions manufacturées présentées ne présentent pas de telle difficulté. Il est donc important pour appuyer la validation numérique de notre méthode de considérer des cas test supplémentaires, plus réalistes. Nous allons ainsi proposer deux cas test numériques sans solution manufacturée : le premier ne présentera pas de singularité, le second en comportera 2.

Cas test 6 : anneau. Dans un premier temps, on se place dans la situation du premier cas test de la Section 2.2, représentée à la Figure 2.13 (gauche), et on considère l'équation (2.24), avec $\boldsymbol{f} = (0, -\rho g)$ avec $\rho = 0.6$ et $g = 9.81$. De plus, on fixe $\boldsymbol{u}^g = (0, 0)$ sur Γ_D et $\boldsymbol{g} = (0, 0)$ sur Γ_N. On applique alors le schéma (2.36) à ce problème et on le compare à une méthode éléments finis classique. Pour calculer l'erreur, la solution de référence sera obtenue par une méthode éléments finis classique, sur un maillage conforme, avec une taille de cellule $h \approx 0.001$.

Les configurations initiale et déformées pour ce cas test, sont représentées à la Figure 2.42, ainsi que la différence entre les solutions approchées et la solution de référence.

On représente à la Figure 2.43 les résultats obtenus pour φ-FEM et une méthode éléments finis classique. Les erreurs relatives L^2 et H^1 sont représentées à la Figure 2.43, semblant confirmer ceux obtenus précédemment, pour des solutions manufacturées. Ainsi, les ordres de convergence optimaux sont atteints pour φ-FEM alors que la méthode

standard est sous-optimale pour les deux normes.

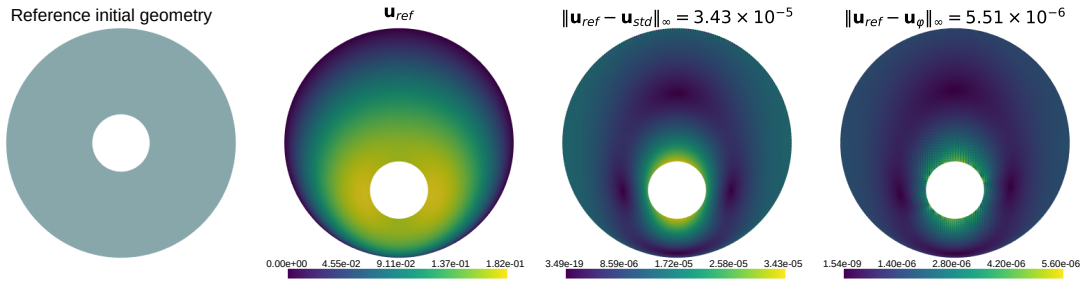

FIGURE 2.42 – **Cas test 6.** Configurations déformées pour les solutions obtenues. Les nuances de couleurs représentent le déplacement pour la solution de référence et l'erreur en norme L^2 (en chaque point) pour φ-FEM et FEM standard.

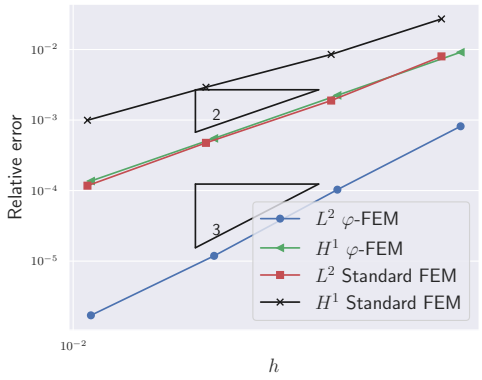

FIGURE 2.43 – **Cas test 6.** Erreurs relatives L^2 et H^1 en fonction de h.

Cas test 7 : disque avec une singularité. Enfin, un cas test supplémentaire important pour valider notre méthode est le cas où une singularité de changement de conditions de bord est présente. Dans ce cas, comme nous l'avons vu pour le problème de Poisson dans la Section 2.2, la solution est au plus $H^{3/2}$ et donc la convergence espérée est d'ordre 1 en norme L^2 et d'ordre 0.5 en norme H^1. Pour vérifier que ces ordres sont atteints, nous considérerons le cas d'un disque fixé sur sa partie haute, et sans contrainte sur la moitié basse, soumis à la gravité, i.e. avec des conditions de Dirichlet sur $\Gamma \cap \{y > 0.5\}$ et de Neumann homogènes sur $\Gamma \cap \{y \leqslant 0.5\}$ et un second membre donné par $\boldsymbol{f} = (0, -\rho g)$ avec $\rho = 0.6$ et $g = 9.81$.

Pour le calcul d'erreur, la solution de référence sera obtenue par une méthode éléments finis classique, sur un maillage conforme très fin. Comme pour le cas test 2 de cette section, nous considérerons 2 situations différentes : dans le premier cas, le changement de conditions de bord sera sur un nœud du maillage standard et par analogie sur une face du maillage φ-FEM ; dans le second cas, le changement sera situé sur une face du maillage standard et à l'intérieur d'une cellule du maillage φ-FEM. Les configurations

initiales et déformées pour un tel cas test, dans la dernière situation, sont représentées à la Figure 2.44, ainsi que la différence entre les solutions approchées et la solution de référence.

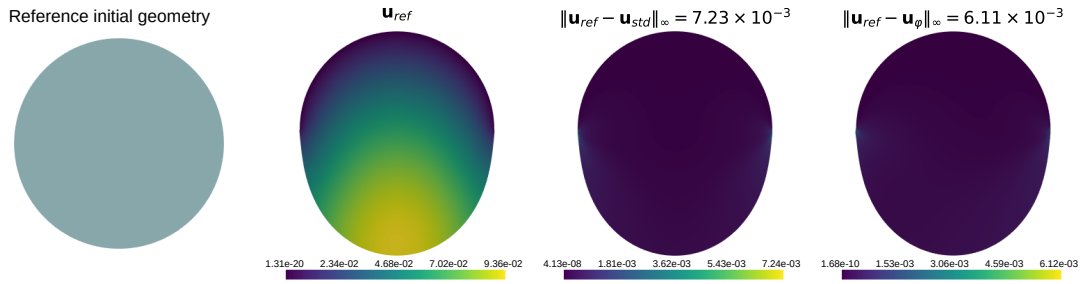

FIGURE 2.44 – **Cas test 7.** Configurations déformées pour les solutions obtenues. Les nuances de couleurs représentent le déplacement pour la solution de référence et l'erreur en norme L^2 (en chaque point) pour φ-FEM et FEM standard.

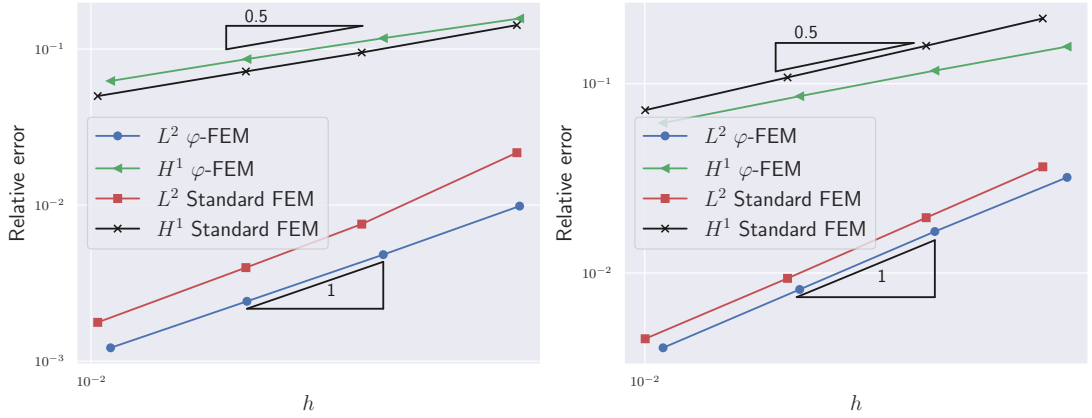

FIGURE 2.45 – **Cas test 7.** Erreurs relatives L^2 et H^1 en fonction de h. Gauche : cas où la jonction intervient sur une face de Ω_h^Γ. Droite : cas où la jonction est à l'intérieur d'une cellule de Ω_h^Γ.

On représente à la Figure 2.45 les résultats obtenus pour φ-FEM et une méthode éléments finis classique. Dans les deux situations, l'ordre optimal est atteint pour la norme L^2 ainsi que pour la norme H^1. De plus, dans les deux cas, l'erreur obtenue en norme L^2 est plus faible pour φ-FEM que pour l'approche standard. Cependant, dans la première situation (Figure 2.45, gauche) la méthode standard donne de meilleurs résultats que la méthode φ-FEM en norme H^1.

2.5 φ-FEM pour l'élasticité non-linéaire

Enfin, le dernier problème type qui sera considéré dans ce manuscrit sera la déformation de matériaux élastiques non-linéaires. Ces équations sont proches de (2.24) à la différence

que le tenseur considéré n'est pas linéaire. Ainsi, le problème sera de trouver $\boldsymbol{u} \in \mathbb{R}^d$ vérifiant

$$\begin{cases} -\operatorname{div} \boldsymbol{P}(\boldsymbol{u}) & = \boldsymbol{f}\,, \quad \text{dans } \Omega\,, \\ \boldsymbol{u} & = \boldsymbol{u}_D\,, \text{ sur } \Gamma_D\,, \\ \boldsymbol{P}(\boldsymbol{u}) \cdot \boldsymbol{n} & = \boldsymbol{g}\,, \quad \text{sur } \Gamma_N\,, \end{cases} \tag{2.54}$$

pour $\boldsymbol{f}$, $\boldsymbol{u}_D$ et $\boldsymbol{g}$ données.

Pour modéliser ce problème, il est possible de considérer différents types de matériaux, modélisant de manière plus ou moins précise (et ainsi plus ou moins complexe) les grandes déformations. Ainsi, l'expression du tenseur $\boldsymbol{P}$ sera modifiée en accord avec le type de matériau choisi. Par exemple, considérant un matériau compressible modélisé avec une loi Néo-Hookéenne, le premier tenseur des contraintes de Piola-Kirchhoff $\boldsymbol{P}$, est donné par (c.f. [48, eq. (6.1)]) :

$$\boldsymbol{P}(F(\boldsymbol{u})) = \frac{\partial W(F(\boldsymbol{u}))}{\partial F},$$

que l'on notera par la suite $\boldsymbol{P}(\boldsymbol{u})$, où la fonction W est définie par (c.f. [8]) :

$$W = \frac{\mu}{2}\left(I_1 - 3 - 2\ln(J)\right) + \frac{\lambda}{2}\ln(J)^2\,.$$

Ici, $I_1 = \operatorname{tr}(C)$ est le premier invariant du tenseur de déformation de Cauchy-Green, C, donné par $C = F^T \cdot F$, où $F = I + \nabla \boldsymbol{u}$ est le tenseur de déformation. Enfin, $J = \det F$ est le déterminant Jacobien.

Les paramètres μ et λ sont définis de la même manière que pour les équations d'élasticité linéaire vues précédemment.

Remarque 2.19. On peut également considérer d'autres lois, telles que la loi de Saint-Venant-Kirchhoff, plus proche de l'élasticité linéaire, pour laquelle W est donnée par

$$W = \frac{1}{2}\lambda(\operatorname{tr}E)^2 + \mu(E : E)\,.$$

2.5.1 Construction du schéma

On se place dans le contexte de l'équation (2.54). Le schéma φ-FEM sera construit de la même manière que le schéma (2.36), en introduisant des variables y, p_N et p_D sur $\Omega_h^{\Gamma_N}$ et $\Omega_h^{\Gamma_D}$ qui sont eux construits comme précédemment. Il suffit alors d'adapter les équations (2.34) au nouveau cas considéré, ce qui donne ainsi

$$\begin{aligned} \boldsymbol{y} + \boldsymbol{P}(\boldsymbol{u}) = 0, \quad & \text{sur } \Omega_h^{\Gamma_N}\,, \\ \boldsymbol{y}\nabla\varphi + \boldsymbol{p}_N\varphi = -\boldsymbol{g}|\nabla\varphi|, \quad & \text{sur } \Omega_h^{\Gamma_N}\,. \end{aligned}$$

Le schéma φ-FEM pour résoudre (2.54) est finalement donné par :
trouver $\boldsymbol{u}_h \in \boldsymbol{V}_h^k$, $\boldsymbol{p}_{h,N} \in \boldsymbol{Q}_h^{(k-1)}(\Omega_h^{\Gamma_N})$, $\boldsymbol{y}_h \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_N})$ et $\boldsymbol{p}_{h,D} \in \boldsymbol{Q}_h^{(k)}(\Omega_h^{\Gamma_D})$ tels que

$$\begin{aligned}
\int_{\Omega_h} \boldsymbol{P}(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h + \int_{\partial\Omega_h^{\Gamma_N}} \boldsymbol{y}_h \boldsymbol{n} \cdot \boldsymbol{v}_h - \int_{\partial\Omega_h \setminus \partial\Omega_h^{\Gamma_N}} \nabla \boldsymbol{u}_h \boldsymbol{n} \cdot \boldsymbol{v}_h - \int_{\Omega_h} \boldsymbol{f}_h \boldsymbol{v}_h \\
+ \gamma_D \int_{\Omega_h^{\Gamma_D}} (\boldsymbol{u}_h - \frac{1}{h}\varphi_h \boldsymbol{p}_{h,D} - \boldsymbol{u}_{h,D})(\boldsymbol{v}_h - \frac{1}{h}\varphi_h \boldsymbol{q}_{h,D}) \\
+ \sigma_D h^2 \sum_{T \in \mathcal{T}_h^{\Gamma_D} \cup \mathcal{T}_h^{\Gamma_{Int}}} \int_T (\operatorname{div} \boldsymbol{P}(\boldsymbol{u}_h) + \boldsymbol{f}_h) \operatorname{div}(D_{\boldsymbol{u}}(\boldsymbol{P})(\boldsymbol{u}_h)\boldsymbol{v}_h) \\
+ \gamma_u \int_{\Omega_h^{\Gamma_N}} (\boldsymbol{y}_h + \boldsymbol{P}(\boldsymbol{u}_h)) : (\boldsymbol{z}_h + D_{\boldsymbol{u}}(\boldsymbol{P})(\boldsymbol{u}_h)\boldsymbol{v}_h) \\
+ \frac{\gamma_p}{h^2} \int_{\Omega_h^{\Gamma_N}} (\boldsymbol{y}_h \nabla\varphi_h + \frac{1}{h}\boldsymbol{p}_{h,N}\varphi_h + \boldsymbol{g}|\nabla\varphi_h|) \cdot (\boldsymbol{z}_h \nabla\varphi_h + \frac{1}{h}\boldsymbol{q}_{h,N}\varphi_h) \\
+ \gamma_{div} \int_{\Omega_h^{\Gamma_N}} (\operatorname{div} \boldsymbol{y}_h + \boldsymbol{f}_h) \cdot \operatorname{div} \boldsymbol{z}_h + G_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = 0\,, \\
\forall \boldsymbol{v}_h \in \boldsymbol{V}_h^k,\ \boldsymbol{q}_{h,N} \in \boldsymbol{Q}_h^{(k-1)}(\Omega_h^{\Gamma_N}),\ \boldsymbol{z}_h \in \boldsymbol{Z}_h(\Omega_h^{\Gamma_N}), \boldsymbol{q}_{h,D} \in \boldsymbol{Q}_h^{(k)}(\Omega_h^{\Gamma_D})\,,
\end{aligned}$$

où

$$\begin{aligned}
G_h(\boldsymbol{u}, \boldsymbol{v}) := \sigma_D h \sum_{E \in \mathcal{F}_h^{\Gamma_D}} \int_E [\boldsymbol{P}(\boldsymbol{u})\boldsymbol{n}] \cdot [D_{\boldsymbol{u}}(\boldsymbol{P})(\boldsymbol{u})\boldsymbol{v}\boldsymbol{n}] \\
+ \sigma_N h \sum_{E \in \mathcal{F}_h^{N_S}} \int_E [\boldsymbol{P}(\boldsymbol{u})\boldsymbol{n}] \cdot [D_{\boldsymbol{u}}(\boldsymbol{P})(\boldsymbol{u})\boldsymbol{v}\boldsymbol{n}]\,,
\end{aligned}$$

avec $D_{\boldsymbol{u}}(\boldsymbol{P})(\boldsymbol{u})\boldsymbol{v}$ la dérivée de $\boldsymbol{P}$ évaluée en $\boldsymbol{u}$, dans la direction $\boldsymbol{v}$ et γ_p, γ_u, γ_{div}, σ_N des constantes positives.

2.5.2 Résultats numériques

Nous allons maintenant comparer ce schéma à une méthode éléments finis classique pour évaluer ses performances. Pour cela, nous étudierons 2 cas test numériques, notamment adaptés des situations vues précédemment dans le cas de l'élasticité linéaire.
Dans un premier temps, nous validerons le schéma sur le cas de l'anneau pour lequel la solution ne présente pas de singularité. Puis, nous considérerons un cas test modélisant la déformation d'une poutre avec des coins arrondis.

Cas test 1 : déformation d'un anneau. Pour ce premier cas, on se place dans le contexte d'une solution sans singularité, en considérant la géométrie du Cas test 1 de la Section 2.2.2 (représentée à la Figure 2.13, Gauche) avec le grand disque de rayon 0.4 et le petit de rayon 0.1. Le domaine est déformé par la gravité, et donc $\boldsymbol{f} = (0, -\rho g)$. On considère de plus le cas de conditions homogènes sur Γ_D et Γ_N et le matériau est modélisé par une loi Néo-Hookéenne. Comme pour le cas de l'élasticité linéaire, les éléments finis utilisés sont de degré 2. Les déformations obtenues sont représentées à la Figure 2.46.

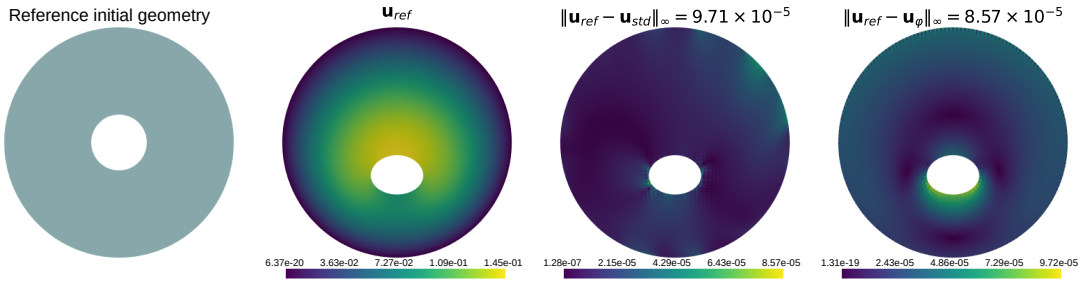

FIGURE 2.46 – **Cas test 1.** De gauche à droite : géométrie considérée ; solution de référence et géométrie déformée par cette solution ; géométrie déformée par la solution Standard-FEM ; géométrie déformée par la solution φ-FEM.

On calcule alors l'erreur relative L^2 pour φ-FEM et Standard-FEM, par rapport à une solution de référence. Les résultats obtenus sont représentés à la Figure 2.47, où l'on observe comme dans le cas de l'élasticité linéaire que les erreurs de φ-FEM convergent à l'ordre 3 lorsque Standard-FEM converge à l'ordre 2. Cependant, il est important de préciser que, le système non-linéaire généré par le schéma φ-FEM est plus lourd que celui de Standard-FEM et il est donc nécessaire de réaliser plus d'itérations lors de la résolution numérique, ce qui est plus coûteux en temps de calcul.

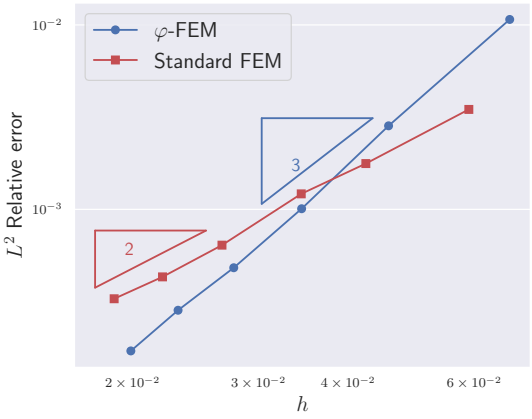

FIGURE 2.47 – **Cas test 1.** Erreurs relatives L^2 en fonction de la taille de cellule h.

Cas test 2 : déformation d'une poutre 2D. Pour le second cas test, nous allons maintenant considérer une situation plus complexe présentant des singularités de changement de conditions de bord. Pour cela, la géométrie représentera une poutre dont les 4 coins seront arrondis, caractérisée par une fonction level-set

$$\varphi(x, y) = \left(\frac{(x - 0.5)^4}{0.43^4} + \frac{(y - 0.5)^4}{0.17^4} \right)^{0.25} - 1 \,.$$

La partie gauche de la poutre sera fixée (conditions de Dirichlet $\boldsymbol{u} = 0$) et la partie droite sera libre (conditions de Neumann $\boldsymbol{P}(\boldsymbol{u}) \cdot n = 0$). Les frontières Dirichlet et Neumann seront caractérisées par la fonction level-set $\psi(x, y) = x - 0.3$, ce qui donne la situation représentée à la Figure 2.48. Enfin, le second membre de (2.54) sera la gravité.

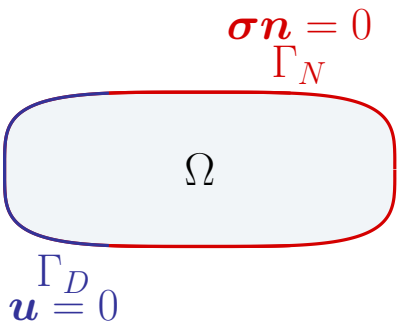

FIGURE 2.48 – **Cas test 2.** Représentation de la situation considérée pour la déformation d'une poutre.

On compare alors la méthode φ-FEM à la méthode Standard-FEM, en calculant l'erreur par rapport à une solution de référence Standard-FEM, obtenue sur un maillage fin.

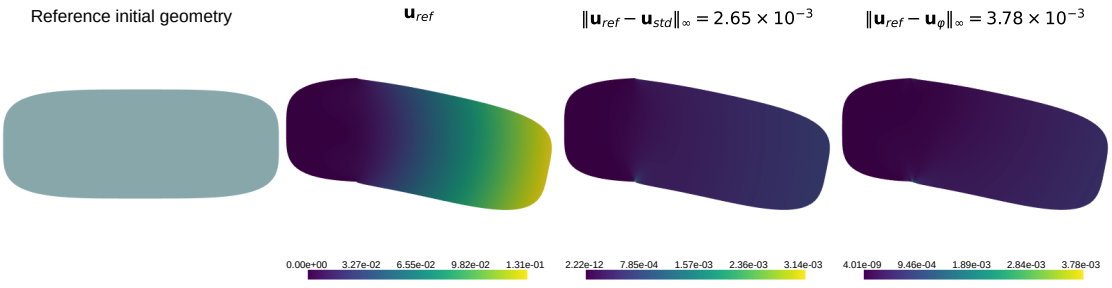

FIGURE 2.49 – **Cas test 2.** De gauche à droite : géométrie considérée ; solution de référence et géométrie déformée par cette solution ; géométrie déformée par la solution Standard-FEM ; géométrie déformée par la solution φ-FEM.

On représente un exemple de solutions obtenues par les deux méthodes à la Figure 2.49. Les résultats numériques présentés à la Figure 2.50 illustrent que les deux méthodes atteignent une convergence d'ordre 1 en norme L^2 relative, la méthode φ-FEM offrant des erreurs légèrement plus faibles que la méthode Standard-FEM.

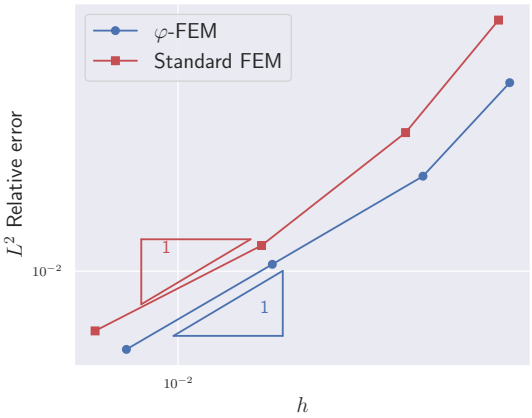

FIGURE 2.50 – **Cas test 2.** Erreurs relatives L^2 en fonction de la taille de cellule h.

2.6 Conclusion

Dans ce chapitre, nous avons présenté plusieurs schémas φ-FEM permettant de résoudre différents problèmes classiquement traités par les méthodes éléments finis. Nous avons dans un premier temps introduit une nouvelle version de la méthode φ-FEM pour résoudre le problème de Poisson avec conditions de Dirichlet, ayant l'avantage d'être compatible avec le schéma φ-FEM pour les conditions de Neumann, ce qui nous a alors permis de construire un schéma complet adapté au cas de conditions mixtes.

Nous avons par la suite traité le cas de l'équation de la chaleur pour laquelle la méthode φ-FEM s'est montrée très intéressante. En effet, nous avons démontré et illustré numériquement que la méthode converge de manière quasi-optimale.

Enfin, nous avons étendu notre étude numérique à différents problèmes d'élasticité, linéaire et non-linéaire. Dans tous les cas étudiés, la méthode φ-FEM s'est montrée au moins aussi performante que la méthode des éléments finis classique, avec des gains notables en précision et en coût de calcul.

3 φ-FD : φ-FEM adaptée aux différences finies

Résumé

Dans ce chapitre, nous présentons une nouvelle approche aux différences finies, inspirée de la méthode φ-FEM. Cette méthode, appelée φ-FD, utilise des grilles cartésiennes, offrant une simplicité d'implémentation. De plus, contrairement aux schémas de différences finies existants pour des domaines complexes, la matrice associée à la méthode est bien conditionnée.

L'utilisation d'une fonction *level-set* pour décrire la géométrie rend cette approche relativement flexible. Nous démontrons ici des taux de convergence quasi-optimaux ainsi que le bon conditionnement de la matrice. Des expériences numériques en 2D et 3D valideront les performances de la méthode φ-FD par rapport aux méthodes standard éléments finis et à l'approche de Shortley-Weller. Nous proposerons finalement une combinaison avec une technique multigrid pour accélérer davantage les calculs.

Chapitre 3 – φ-FD : φ-FEM adaptée aux différences finies

Dans ce chapitre, nous allons proposer une nouvelle approche aux différences finies pour résoudre l'équation de Poisson Dirichlet (1.1) sur un domaine Ω, de frontière $\Gamma = \partial\Omega$, que l'on rappelle :

$$\begin{cases} -\Delta u &= f\,, \quad \text{dans } \Omega\,, \\ u &= 0\,, \quad \text{sur } \Gamma\,. \end{cases}$$

Les résultats présentés dans ce chapitre ont été publiés dans l'article [25].

Nous avons jusqu'à présent considéré uniquement des méthodes éléments finis pour résoudre des EDP. Cependant, une autre méthode répendue dans ce domaine est la méthode des différences finies. Cette méthode est répandue notamment en raison de son efficacité numérique. En revanche, les approches différences finies classiques sont très limitées puisque nécessitant l'utilisation de grilles cartésiennes exclusivement. L'approche principale utilisée dans la littérature pour appliquer des méthodes aux différences finies à des géométries complexes est la méthode introduite par Shortley et Weller dans [82]. Dans [89, 9], les auteurs ont proposé des techniques d'étude de convergence pour cette méthode, utilisant des fonctions de Green discrètes ainsi que le principe du maximum pour obtenir des estimations précises des coefficients de la matrice inverse. Ces estimations génèrent parfois des phénomènes de « supraconvergence », où le schéma numérique converge à un ordre plus élevé que l'ordre espéré en théorie. Dans [20], les auteurs ont considéré des problèmes elliptiques avec des interfaces immergées. Un schéma de second ordre pour résoudre l'équation de Poisson avec conditions de Dirichlet sur des domaines irréguliers a été proposé dans [37]. La méthode d'interface immergée [55] est basée sur une grille cartésienne et associée à un schéma aux différences finies de second ordre, pour des équations elliptiques de second ordre générales ainsi que des équations paraboliques linéaires. La combinaison des différences finies et de méthodes non conformes est ainsi une idée naturelle. Cependant, l'inconvénient des méthodes proposées précédemment dans la littérature est généralement le mauvais conditionnement des matrices associées.

Dans ce chapitre, nous proposons un schéma aux différences finies sur grille cartésienne inspiré par φ-FEM. La géométrie sera décrite par une fonction level-set φ, utilisée pour appliquer les conditions de bord par pénalisation. Cette nouvelle méthode, appelée φ-FD allie convergence optimale, bon conditionnement de la matrice associée au problème et facilité d'implémentation (peu de lignes de code python, avec l'aide du package `scipy` [88], cf. Annexe A.1). La première section de ce chapitre sera dédiée à la présentation du schéma. Dans la deuxième section, nous ferons le lien entre cette méthode et la méthode φ-FEM, en particulier une version légèrement modifiée du schéma dual (2.2). Nous proposerons ensuite des preuves de résultats théoriques à la troisième section. Une deuxième version de schéma φ-FD sera introduite dans la quatrième section, sans résultat théorique. Enfin, la dernière section sera dédiée à la présentation des résultats numériques.

3.1 Présentation du schéma et des résultats principaux

On considère un domaine Ω défini par une fonction φ, telle que $\Omega = \{\varphi < 0\}$, et $\Gamma = \{\varphi = 0\}$.

On suppose que Ω est inclus dans $\mathcal{O} := \prod_{i=1}^{n} [a_i, b_i]$ avec $b_i - a_i = b_j - a_j$ pour $i \neq j$. Soit $N \in \mathbb{N}^*$, $h = (b_1 - a_1)/N$, on considère la grille cartésienne couvrant ce rectangle :

$$\mathcal{O}_h := \{x_\alpha : \alpha \in \{0, \cdots, N\}^n\}$$

avec $x_{\alpha_i} = a_i + \alpha_i h$ pour $\alpha = (\alpha_1, \cdots, \alpha_n)$. On note

$$D = \begin{cases} \{1\}, & \text{si } n = 1, \\ \{(1,0),(0,1)\}, & \text{si } n = 2, \\ \{(1,0,0),(0,1,0),(0,0,1)\}, & \text{si } n = 3\,, \end{cases}$$

et on définit les sous-grilles suivantes :

$$\Omega_h = \{x_\alpha \in \mathcal{O}_h : x_\alpha \in \Omega \text{ ou } x_{\alpha \pm d} \in \Omega,\ d \in D\}\,,$$

$$\Omega_h^{\text{int}} = \{x_\alpha \in \mathcal{O}_h : x_\alpha \in \Omega\}\,.$$

De plus, soit $\overline{\Omega}_h$, l'union des carrés de sommets $x_\alpha \in \mathcal{O}_h$ en intersection avec Ω et soit $\overline{\Omega}_h^{\text{int}}$ l'union des carrés de sommets $x_\alpha \in \mathcal{O}_h$ inclus dans Ω. Un exemple de situation est représenté à la Figure 3.1.

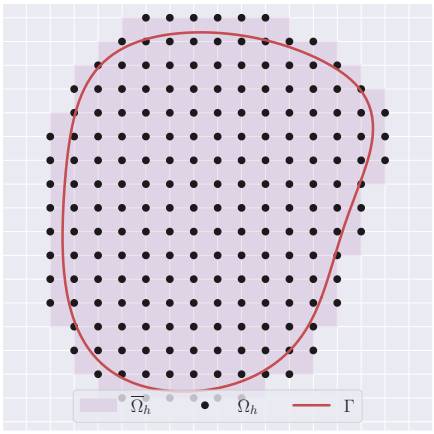

FIGURE 3.1 – Représentation de $\overline{\Omega}_h$, Ω_h et Γ.

Présentons maintenant notre schéma, qui sera introduit ici pour toute dimension. Le schéma sera également décrit en 2 dimensions, avec les indices explicités en Section 3.3. Le schéma est donné par : trouver une fonction discrète $u_h = (u_\alpha)_{\alpha : x_\alpha \in \Omega_h}$ définie sur Ω_h, telle que

$$a_h(u_h, v_h) = l_h(v_h), \tag{3.1}$$

pour toute fonction discrète $v_h = (v_\alpha)_{\alpha : x_\alpha \in \Omega_h}$ définie sur Ω_h, où

$$a_h(u_h, v_h) = (-\Delta_h u_h, v_h) + b_h(u_h, v_h) + j_h(u_h, v_h)\,,$$

et

$$l_h(v_h) = \sum_{\alpha : x_\alpha \in \Omega_h^{\text{int}}} \sum_{d \in D} f_\alpha v_\alpha\,,$$

avec $f_\alpha = (f(x_\alpha))_\alpha$, le Laplacien discret :

$$(-\Delta_h u_h, v_h) = \sum_{\alpha: x_\alpha \in \Omega_h^{\text{int}}} \sum_{d \in D} \frac{-u_{\alpha-d} + 2u_\alpha - u_{\alpha+d}}{h^2} v_\alpha,$$

la pénalisation pour imposer les conditions de bord

$$b_h(u_h, v_h) = \frac{\gamma}{2h^2} \sum_{(\alpha,d)\in B} \frac{1}{\varphi_\alpha^2 + \varphi_{\alpha+d}^2} (\varphi_{\alpha+d} u_\alpha - \varphi_\alpha u_{\alpha+d})(\varphi_{\alpha+d} v_\alpha - \varphi_\alpha v_{\alpha+d})$$

et un terme de stabilisation proche de la frontière

$$j_h(u_h, v_h) = \sigma \sum_{(\alpha,d)\in J} \frac{-u_{\alpha-d} + 2u_\alpha - u_{\alpha+d}}{h} \times \frac{-v_{\alpha-d} + 2v_\alpha - v_{\alpha+d}}{h}$$

avec $\gamma,\ \sigma > 0$ et

$$B = \{(\alpha, d) | \text{ le segment } x_\alpha - x_{\alpha+d} \text{ intersecte } \Gamma \text{ et n'est pas inclus dans } \Gamma\},$$

$$J = \{(\alpha, d) | x_\alpha \in \Omega \text{ et } [x_{\alpha-d} \notin \Omega \text{ ou } x_{\alpha+d} \notin \Omega]\}.$$

Les normes discrètes L^2, L^∞ et la semi-norme discrète H^1 sont définies pour tout $v_h = (v_\alpha)_{\alpha: x_\alpha \in \Omega_h^{\text{int}}}$ par

$$\|v_h\|_{h,0} = \left(h^n \sum_{\alpha: x_\alpha \in \Omega_h^{\text{int}}} v_\alpha^2 \right)^{1/2}, \quad \|v_h\|_{h,\infty} = \max_{\alpha: x_\alpha \in \Omega_h^{\text{int}}} |v_\alpha|$$

et

$$|v_h|_{h,1} = \left(\sum_{\substack{\alpha,d: x_\alpha \in \Omega^{\text{int}} \\ \text{et } x_{\alpha+d} \in \Omega^{\text{int}}}} h^n \left| \frac{v_{\alpha+d} - v_\alpha}{h} \right|^2 \right)^{1/2}.$$

Dans la suite de ce chapitre, dans les différentes inégalités, C sera utilisée pour des constantes indépendantes de h et de f.

Introduisons maintenant la notion de régularité qui sera nécessaire sur le domaine :

Définition 3.1. Un domaine Ω est dit *r-smooth*, si pour chaque point $x_0 \in \Gamma$ il existe un cône centré en x_0 d'angle strictement plus grand que $\pi/2$ et de rayon r, inclus dans $\overline{\Omega}$.

Énonçons maintenant le résultat principal de ce chapitre, le résultat de convergence :

Théorème 3.1 (Convergence, cf. [25, Théorème 1])**.** *Supposons que Ω est r-smooth, pour $r > 0$ et est défini par une fonction level-set $\varphi \in \mathcal{C}^2(\overline{\Omega}_h)$. Soit u la solution du problème continu* (1.1)*, telle que $u \in \mathcal{C}^4(\Omega)$. Pour σ, γ suffisamment grands et $h < \dfrac{2r}{\sqrt{10}}$, le système discret* (3.1) *admet une unique solution u_h. Dans ce cas, notant $U = (u(x_\alpha))_{\alpha: x_\alpha \in \Omega_h^{int}}$, alors*

$$\|U - u_h\|_{h,0} + \|U - u_h\|_{h,\infty} + |U - u_h|_{h,1} + \leqslant C h^{3/2} \|u\|_{\mathcal{C}^4(\Omega)}.$$

Remarque 3.1. Il est intéressant de remarquer que :

- L'ordre de convergence L^2 donné dans le Théorème 3.1 pourrait ne pas être optimal puisque numériquement, on observe une convergence d'ordre 2. De plus, les schémas différences finies classiques sont également d'ordre 2.
- Pour la norme H^1, l'ordre de convergence est plus élevé que pour les méthodes éléments finis. Ce phénomène est bien connu et est appelé supraconvergence (cf. [35] par exemple).

De plus, la matrice associée au système discret est bien conditionnée :

Théorème 3.2 (Conditionnement, cf. [25, Théorème 2])**.** *Sous les hypothèses du théorème 3.1, le conditionnement défini par* $\kappa(A) := \|A\|_2\|A^{-1}\|_2$ *de la matrice* A *associée à la forme bilinéaire* a_h *vérifie*

$$\kappa(A) \leq Ch^{-2}.$$

Ici, $\|\cdot\|_2$ *est la norme matricielle associée à la norme euclidienne.*

Ces deux théorèmes seront prouvés à la Section 3.3 dans le cas 2D pour des raisons de lisibilité. Cependant, les preuves peuvent être étendues de la même façon en ajoutant les indices correspondants en 3 dimensions.

Remarque 3.2. Dans le cas de conditions de Dirichlet non homogènes $u_h^D = (u_\alpha^D)_\alpha$, il est nécessaire d'ajouter le terme suivant dans le second membre :

$$b_h^{rhs}(v_h) = \frac{\gamma}{2h^2}\left(\sum_{(\alpha,d)\in B}\frac{1}{\varphi_\alpha^2+\varphi_{\alpha+d}^2}(\varphi_{\alpha+d}u_\alpha^D - \varphi_\alpha u_{\alpha+d}^D)(\varphi_{\alpha+d}v_\alpha - \varphi_\alpha v_{\alpha+d})\right).$$

3.2 Lien avec φ-FEM

On considère un maillage cartésien $\mathcal{T}_h^{\mathcal{O}}$ triangulaire (ou tétraédrique en 3D) de $\mathcal{O}$ avec des nœuds (x_α), $\mathcal{T}_h$ l'ensemble de cellules de $\mathcal{T}_h^{\mathcal{O}}$ en intersection avec Ω, Ω_h^{EF} le domaine couvert par le maillage $\mathcal{T}_h$ et $\partial\Omega_h^{EF}$ sa frontière. Soit $\mathcal{E}_h^\Gamma$ l'ensemble des faces de $\mathcal{T}_h$ coupées par Γ et $\mathcal{F}_h^\Gamma$ l'ensemble des faces internes des cellules de $\mathcal{T}_h$ coupées par Γ (cf. (1.8)). On définit

$$V_h = \{v_h \in C^0(\Omega_h) : v_{h|K} \in \mathbb{P}_1(K)\ \forall K \in \mathcal{T}_h\}$$

et

$$Q_h = \{p_h \in L^2(E_h^\Gamma) : p_{h|K} \in \mathbb{P}_0(E)\ \forall E \in \mathcal{E}_h^\Gamma\},$$

où $E_h^\Gamma = \cup_{E\in\mathcal{E}_h^\Gamma} E$.

On construit alors le schéma φ-FEM suivant pour (1.1) :
Trouver $(u_h, p_h) \in V_h \times Q_h$ tels que

$$\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h} \nabla u_h \cdot n v_h + \frac{\gamma}{h} \sum_{E \in \mathcal{E}_h^\Gamma} \int_E (u_h - \varphi_h p_h)(v_h - \varphi_h q_h)$$
$$+ \sigma h \sum_{F \in \mathcal{F}_h^\Gamma} \int_F [n \cdot \nabla u_h][n \cdot \nabla v_h] = \int_{\Omega_h} f v_h, \quad \forall (v_h, q_h) \in V_h \times Q_h. \quad (3.2)$$

Cette version du schéma φ-FEM est une variante de la version proposée à la Section 2.1, où l'on impose $u_h \sim \varphi_h p_h$ par pénalisation sur les faces $E \in \mathcal{E}_h^\Gamma$. La solution u_h est représentée par ses valeurs u_α aux nœuds x_α. Si x_α et tous ses voisins sont à l'intérieur de Ω, alors (3.2) donne la discrétisation

$$\sum_{d \in D} \frac{-u_{\alpha-d} + 2u_\alpha - u_{\alpha+d}}{h^2} = f_\alpha. \quad (3.3)$$

Ainsi, nous obtenons les équations aux nœuds intérieurs mais les inconnues sont également définies aux nœuds extérieurs à Ω, adjacents à un nœud interne.
Si v_h est une fonction de base associée à un tel nœud, alors la contribution $\int_{\Omega_h} \nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h} \nabla u_h \cdot n v_h$ et la partie correspondante dans le second membre $\int_{\Omega_h} f v_h$ sont ignorées. Sinon, on conserve l'équation venant de

$$\frac{\gamma}{h^3} \sum_{E \in \mathcal{E}_h^\Gamma} \int_E (u_h - \varphi_h p_h)(v_h - \varphi_h q_h), \quad (3.4)$$

qui a été divisée par h^2 par cohérence avec (3.3). Pour toute face $E \in \mathcal{E}_h^\Gamma$, p_h et q_h sur E valent p_E et q_E. En prenant $v_h = 0$ dans (3.4), cela donne

$$\int_E (u_h - \varphi_h p_E)\varphi_h = 0,$$

et donc

$$p_E = \frac{\int_E u_h \varphi_h}{\int_E \varphi_h^2}.$$

Finalement, en prenant $q_h = 0$ et en remplaçant p_h, (3.4) devient

$$\frac{\gamma}{h^3} \sum_{E \in \mathcal{E}_h^\Gamma} \int_E \left(u_h - \frac{\int_E u_h \varphi_h}{\int_E \varphi_h^2} \varphi_h \right) v_h. \quad (3.5)$$

Dans le cas où $E \in \mathcal{E}_h^\Gamma$ est une face de x_α à $x_{\alpha+d}$, avec x_α dans Ω et $x_{\alpha+d}$ en-dehors, et $d \in D$,

$$u_h - \frac{\int_E u_h \varphi_h}{\int_E \varphi_h^2} \varphi_h = \begin{cases} \frac{\varphi_{\alpha+d}}{\varphi_\alpha^2 + \varphi_{\alpha+d}^2} (\varphi_{\alpha+d} u_\alpha - \varphi_\alpha u_{\alpha+d}) \text{ en } x_\alpha, \\ \frac{\varphi_\alpha}{\varphi_\alpha^2 + \varphi_{\alpha+d}^2} (\varphi_\alpha u_{\alpha+d} - \varphi_{\alpha+d} u_\alpha) \text{ en } x_{\alpha+d} \end{cases},$$

de sorte que la contribution à (3.5) sur cette face E est donnée par

$$\frac{\gamma}{2h^2} \frac{1}{\varphi_\alpha^2 + \varphi_{\alpha+d}^2} (\varphi_{\alpha+d} u_\alpha - \varphi_\alpha u_{\alpha+d})(\varphi_{\alpha+d} v_\alpha - \varphi_\alpha v_{\alpha+d}),$$

qui est du même ordre que le terme de pénalisation b_h. Des formules similaires sont valables pour les autres configurations des faces $E \in \mathcal{E}_h^\Gamma$. On obtient finalement la matrice qui représente (3.4), qui doit être ajoutée à la matrice qui représente (3.3).

Finalement, la « ghost penalty »,

$$\sigma h \sum_{F \in \mathcal{F}_h^\Gamma} \int_F [n \cdot \nabla u_h][n \cdot \nabla v_h], \tag{3.6}$$

qui sera également divisée par h^2 peut être approchée à la manière des différences finies. Ainsi, en considérant un nœud x_α à l'intérieur de Ω tel que $x_{\alpha+d}$ soit à l'extérieur de Ω avec $d \in D$, les deux côtés $(x_{\alpha-d} - x_\alpha)$ et $(x_\alpha - x_{\alpha+d})$ adjacents à x_α sont dans $\mathcal{F}_h^\Gamma$ et les contributions sur ces côtés peuvent être approchées par

$$\sigma \frac{-u_{\alpha-d} + 2u_\alpha - u_{\alpha+d}}{h} \times \frac{-v_{\alpha-d} + 2v_\alpha - v_{\alpha+d}}{h}.$$

3.3 Preuves des théorèmes de convergence

La majorité de la littérature [59, 50] analyse les méthodes différences finies en utilisant le formalisme des méthodes éléments finis ou volumes finis [49] pour des problèmes elliptiques. Nous avons fait le choix ici de suivre le formalisme éléments finis.

Introduisons les normes discrètes L^2, L^∞ et semi-H^1 sur Ω_h, définies pour tout $v_h = (v_\alpha)_{\alpha : x_\alpha \in \Omega_h}$ par

$$\|v_h\|_{h,0,\Omega_h} = \left(h^2 \sum_{\alpha : x_\alpha \in \Omega_h} v_\alpha^2 \right)^{1/2}, \quad \|v_h\|_{h,\infty,\Omega_h} = \max_{\alpha : x_\alpha \in \Omega_h} |v_\alpha|$$

et

$$|v_h|_{h,1,\Omega_h} = \left(\sum_{\substack{\alpha, d : x_\alpha \in \Omega \\ \text{or } x_{\alpha+d} \in \Omega}} h^2 \left| \frac{v_{\alpha+d} - v_\alpha}{h} \right|^2 \right)^{1/2}.$$

Comme dit précédemment, nous allons ici nous concentrer sur le cas 2D, mais les situations en dimensions supérieures peuvent être traitées similairement.
Dans cette situation le problème peut être réécrit sous la forme : trouver une fonction discrète $u_h = (u_{ij})_{ij}$ définie sur Ω_h telle que

$$a_h(u_h, v_h) = l_h(v_h),$$

pour toute fonction discrète $v_h = (v_{ij})_{ij}$ définie sur Ω_h, où

$$a_h(u_h, v_h) = (-\Delta_h u_h, v_h) + b_h(u_h, v_h) + j_h(u_h, v_h)\,,$$

et

$$l_h(v_h) = \sum_{i,j} f_{ij} v_{ij}\,,$$

avec le Laplacien discret :

$$(-\Delta_h u_h, v_h) = \sum_{i,j|(x_i,y_j)\in\Omega} \frac{4u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} v_{ij},$$

une pénalisation pour les conditions de bord

$$b_h(u_h, v_h) = \frac{\gamma}{h^2}\left(\sum_{(i,j)\in B_x} \frac{1}{\varphi_{ij}^2 + \varphi_{i+1,j}^2}(\varphi_{i+1,j}u_{ij} - \varphi_{ij}u_{i+1,j})(\varphi_{i+1,j}v_{ij} - \varphi_{ij}v_{i+1,j}) \right.$$
$$\left. + \sum_{(i,j)\in B_y} \frac{1}{\varphi_{ij}^2 + \varphi_{i,j+1}^2}(\varphi_{i,j+1}u_{ij} - \varphi_{ij}u_{i,j+1})(\varphi_{i,j+1}v_{ij} - \varphi_{ij}v_{i,j+1}) \right)$$

et la stabilisation près du bord

$$j_h(u_h, v_h) = \sigma\Big(\sum_{(i,j)\in J_x} \frac{-u_{i-1,j} + 2u_{ij} - u_{i+1,j}}{h} \times \frac{-v_{i-1,j} + 2v_{ij} - v_{i+1,j}}{h}$$
$$+ \sum_{(i,j)\in J_y} \frac{-u_{i,j-1} + 2u_{ij} - u_{i,j+1}}{h} \times \frac{-v_{i,j-1} + 2v_{ij} - v_{i,j+1}}{h} \Big)$$

avec $\gamma,\ \sigma > 0$ et

$B_x = \{(i,j)|$ le segment $(x_i, y_j) - (x_{i+1}, y_j)$ intersecte Γ et est non inclus dans $\Gamma\}$,

$B_y = \{(i,j)|$ le segment $(x_i, y_j) - (x_i, y_{j+1})$ intersecte Γ et est non inclus dans $\Gamma\}$,

$$J_x = \{(i,j)|(x_i, y_j) \in \Omega \text{ et } [(x_{i-1}, y_j) \notin \Omega \text{ ou } (x_{i+1}, y_j) \notin \Omega]\},$$

et

$$J_y = \{(i,j)|(x_i, y_j) \in \Omega \text{ et } [(x_i, y_{j-1}) \notin \Omega \text{ ou } (x_i, y_{j+1}) \notin \Omega]\}.$$

Introduisons maintenant quelques résultats intermédiaires, nécessaires pour prouver les théorèmes 3.1 and 3.2. Le premier résultat est une adaptation du Lemme 3.3 de [28], qui sera central dans la preuve de convergence.

Lemme 3.1. *Il existe $\alpha_1 \in (0,1)$, $\alpha_2 \in (0, 1/2)$ et $\beta > 0$ tels que*

$$\left|\frac{u_1 - u_0}{h}\right|^2 \leqslant \alpha_1 \left|\frac{u_1 - u_0}{h}\right|^2 + \alpha_2 \left|\frac{u_2 - u_1}{h}\right|^2 + \beta \left|\frac{u_0 - 2u_1 + u_2}{h}\right|^2$$

pour tous $u_0, u_1, u_2 \in \mathbb{R}$.

Preuve. Pour tous $a,\ b \in \mathbb{R}$ et $\varepsilon,\ \delta > 0$,

$$\begin{aligned} a^2 &\leq |a|(|a-b| + |b|) \leq \frac{1}{2\varepsilon}a^2 + \frac{\varepsilon}{2}(|a-b| + |b|)^2 \\ &\leq \frac{1}{2\varepsilon}a^2 + \frac{\varepsilon}{2}b^2 + \varepsilon|a-b||b| + \frac{\varepsilon}{2}(a-b)^2 \\ &\leq \frac{1}{2\varepsilon}a^2 + \frac{\varepsilon}{2}(1+\delta)b^2 + \frac{\varepsilon}{2}(1 + \frac{1}{\delta})(a-b)^2. \end{aligned}$$

Pour $\varepsilon = \frac{3}{4}$ et $\delta = \frac{1}{6}$, on a

$$a^2 \leq \frac{2}{3}a^2 + \frac{7}{16}b^2 + \frac{\varepsilon}{2}(1 + \frac{1}{\delta})(a-b)^2,$$

ce qui entraîne la conclusion. □

Lemme 3.2. *Pour tout $\beta > 0$, il existe $\alpha \in (0,1)$ tel que pour tous $u_{ij} \in \mathbb{R}$*

$$\begin{aligned}\left|\frac{u_{10}-u_{00}}{h}\right|^2 + \left|\frac{u_{20}-u_{10}}{h}\right|^2 \leqslant \alpha &\left(\left|\frac{u_{10}-u_{00}}{h}\right|^2 + \left|\frac{u_{20}-u_{10}}{h}\right|^2\right) \\ &+ \beta\left(\left|\frac{u_{11}-u_{01}}{h}\right|^2 + \left|\frac{u_{11}-u_{10}}{h}\right|^2 + \left|\frac{u_{01}-u_{02}}{h}\right|^2\right. \\ &\left. + \left|\frac{u_{00}-2u_{10}+u_{20}}{h}\right|^2 + \left|\frac{u_{00}-2u_{01}+u_{02}}{h}\right|^2\right). \quad (3.7)\end{aligned}$$

Preuve. Il est seulement nécessaire de prouver que pour tout $\beta > 0$, il existe $\alpha \in (0,1)$ tel que pour tout $u_{ij} \in \mathbb{R}$

$$\begin{aligned}\left|\frac{u_{10}-u_{00}}{h}\right|^2 + \left|\frac{u_{20}-u_{10}}{h}\right|^2 \leqslant \alpha &\left(\left|\frac{u_{10}-u_{00}}{h}\right|^2 + \left|\frac{u_{20}-u_{10}}{h}\right|^2\right) \\ &+ \beta\left(\alpha\left|\frac{u_{11}-u_{01}}{h}\right|^2 + \alpha\left|\frac{u_{11}-u_{10}}{h}\right|^2 + \alpha\left|\frac{u_{01}-u_{02}}{h}\right|^2\right. \\ &\left. + \left|\frac{u_{00}-2u_{10}+u_{20}}{h}\right|^2 + \left|\frac{u_{00}-2u_{01}+u_{02}}{h}\right|^2\right). \quad (3.8)\end{aligned}$$

En effet, le second membre de (3.8) est plus petit que celui de (3.7). On considère

$$\alpha = \sup \frac{|u_{10}-u_{00}|^2 + |u_{20}-u_{10}|^2 - \beta\left(|u_{00}-2u_{10}+u_{20}|^2 + |u_{00}-2u_{01}+u_{02}|^2\right)}{|u_{10}-u_{00}|^2 + |u_{20}-u_{10}|^2 + \beta|u_{11}-u_{01}|^2 + \beta|u_{11}-u_{10}|^2 + \beta|u_{01}-u_{02}|^2}, \quad (3.9)$$

où

$$D := |u_{10}-u_{00}|^2 + |u_{20}-u_{10}|^2 + \beta|u_{11}-u_{01}|^2 + \beta|u_{11}-u_{10}|^2 + \beta|u_{01}-u_{02}|^2 \neq 0.$$

Sans perte de généralité, on peut supposer que

$$|u_{10}-u_{00}|^2 + |u_{20}-u_{10}|^2 + \beta|u_{11}-u_{01}|^2 + \beta|u_{11}-u_{10}|^2 + \beta|u_{01}-u_{02}|^2 = 1 \quad (3.10)$$

et

$$\sum_{i,j} u_{ij} = 0. \quad (3.11)$$

En effet, si $D \neq 1$, on peut diviser tous les u_{ij} par D. De plus, il est possible de soustraire $\sum_{i,j} u_{ij}$ aux u_{ij} dans (3.9) sans changer la définition de α, et donc les u_{ij} peuvent être choisis de sorte que (3.11) soit vérifiée. De plus, on a clairement $\alpha \leq 1$.

Montrons que l'ensemble de u_{ij} vérifiant (3.10) et (3.11) est uniformément borné. On note $\bar{u}_{ij} = u_{ij} - u_{00}$ pour tout $(i,j) \neq (0,0)$. Alors, (3.10) peut s'écrire

$$|\bar{u}_{10}|^2 + |\bar{u}_{20} - \bar{u}_{10}|^2 + \beta\,|\bar{u}_{11} - \bar{u}_{01}|^2 + \beta\,|\bar{u}_{11} - \bar{u}_{10}|^2 + \beta\,|\bar{u}_{01} - \bar{u}_{02}|^2 = 1.$$

On en déduit que

$$|\bar{u}_{10}| \leqslant 1,\ |\bar{u}_{20}| \leqslant 2,\ |\bar{u}_{11}| \leqslant 1 + 1/\beta,\ |\bar{u}_{01}| \leqslant 1 + 2/\beta,\ |\bar{u}_{02}| \leqslant 1 + 3/\beta. \tag{3.12}$$

En utilisant (3.11) et l'inégalité triangulaire,

$$|6u_{00}| = \left|\sum_{(i,j)} (u_{ij} - u_{00})\right| = \left|\sum_{(i,j)} \bar{u}_{ij}\right| \leqslant 6 + 6/\beta.$$

Ainsi, $|u_{00}| \leqslant 1 + 1/\beta$. De plus, en utilisant (3.12), $|\bar{u}_{ij}| \leqslant 2 + 3/\beta$. Alors,

$$|u_{ij}| \leqslant 3 + 4/\beta$$

pour tout (i,j).

Puisque l'ensemble de u_{ij} satisfaisant (3.10) et (3.11) est fermé, borné et de dimension finie, le supremum dans la définition de α est atteint. On suppose que $\alpha = 1$. Puisque $\beta \neq 0$, il existe u_{ij} tel que

$$|u_{11} - u_{01}|^2 + |u_{11} - u_{10}|^2 + |u_{01} - u_{02}|^2 + |u_{00} - 2u_{10} + u_{20}|^2 + |u_{00} - 2u_{01} + u_{02}|^2 = 0.$$

On en déduit que $u_{11} = u_{01} = u_{10} = u_{02}$, et donc

$$|u_{00} - 2u_{10} + u_{20}|^2 + |u_{00} - u_{10}|^2 = 0.$$

Finalement $u_{00} = u_{10} = u_{20}$ ce qui est en contradiction avec (3.10). □

Les lemmes 3.1 et 3.2 nous permettent de déduire la coercivité de la forme bilinéaire a_h :

Proposition 3.1 (Coercivité)**.** *Il existe $c > 0$ tel que, pour tout u_h,*

$$a_h(u_h, u_h) \geqslant c|||u_h|||_h^2,$$

où

$$|||u_h|||_h = \left(\frac{1}{h^2}|u_h|^2_{h,1,\Omega_h} + b_h(u_h, u_h) + j_h(u_h, u_h)\right)^{1/2}.$$

Dans la preuve suivante ainsi que dans le reste de ce chapitre, la notation suivante sera utilisée : pour tout i, j,

$$u^{\varphi}_{(i,j)-(i+1,j)} = \frac{\varphi_{i,j}u_{i+1,j} - \varphi_{i+1,j}u_{i,j}}{\varphi_{i,j} - \varphi_{i+1,j}}. \tag{3.13}$$

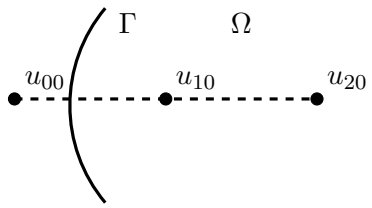

FIGURE 3.2 – Cas $N_j > 2$ dans la preuve de la Proposition 3.1.

Preuve de la Proposition 3.1. Fixons l'indice j et supposons que les nœuds (x_i, y_j) appartenant à Ω_h sont pour $i \in \{M_j, \dots, N_j\}$. Sans perte de généralité, on peut supposer que $M_j = 0$.

Cas $N_j > 2$: On se place d'abord dans la situation représentée à la Figure 3.2.
On remarque que

$$\sum_{i=1}^{N_j-1} (-u_{i-1,j} + 2u_{i,j} - u_{i+1,j})u_{i,j}$$
$$= -\underbrace{(u_{0,j} - u_{1,j})u_{0,j}}_{(I)} + \underbrace{(u_{N_j-1,j} - u_{N_j,j})u_{N_j,j}}_{(II)} + \sum_{i=0}^{N_j-1} |u_{i+1,j} - u_{ij}|^2.$$

Commençons par estimer le terme (I). En utilisant la notation (3.13), on remarque que

$$u_{0,j} = \frac{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}}{\varphi_{0,j} - \varphi_{1,j}} \frac{u_{0,j}\varphi_{0,j} - u_{0,j}\varphi_{1,j}}{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}}$$
$$= \frac{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}}{\varphi_{0,j} - \varphi_{1,j}} \left(u^{\varphi}_{(0,j)-(1,j)} + \frac{\varphi_{0,j}}{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}}(u_{0,j} - u_{1,j}) \right). \quad (3.14)$$

Puisque $\varphi_{0,j} \geq 0$ et $\varphi_{1,j} < 0$, on a

$$0 \leqslant \frac{\varphi_{0,j}}{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}} < 1 \text{ et } \frac{\sqrt{\varphi_{0,j}^2 + \varphi_{1,j}^2}}{\varphi_{0,j} - \varphi_{1,j}} \leq 1. \quad (3.15)$$

Alors,

$$(I) \leq |(u_{0,j} - u_{1,j})u^{\varphi}_{(0,j)-(1,j)}| + (u_{0,j} - u_{1,j})^2.$$

De plus, en utilisant l'inégalité de Young avec $\varepsilon > 0$ et le Lemme 3.1 avec $\alpha_1 \in (0,1)$, $\alpha_2 \in (0, 1/2)$ et $\beta > 0$, on observe que

$$(I) \leq \frac{1}{2\varepsilon}(u^{\varphi}_{(0,j)-(1,j)})^2 + \left(1 + \frac{\varepsilon}{2}\right)(u_{0,j} - u_{1,j})^2$$
$$\leqslant \frac{1}{2\varepsilon}(u^{\varphi}_{(0,j)-(1,j)})^2 + \left(1 + \frac{\varepsilon}{2}\right)(\alpha_1|u_{1,j} - u_{0,j}|^2 + \alpha_2|u_{2,j} - u_{1,j}|^2)$$
$$+ \left(1 + \frac{\varepsilon}{2}\right)\beta|u_2 - 2u_1 + u_0|^2.$$

De façon similaire,

$$(II) \leqslant \frac{1}{2\varepsilon}(u^{\varphi}_{(N_j-1,j)-(N_j,j)})^2 + \left(1+\frac{\varepsilon}{2}\right)\left(\alpha_1|u_{N_j-1,j}-u_{N_j,j}|^2+\alpha_2|u_{N_j-2,j}-u_{N_j-1,j}|^2\right) + \left(1+\frac{\varepsilon}{2}\right)\beta|u_{N_j-2}-2u_{N_j-1}+u_{N_j}|^2.$$

Puisque $N_j > 2$, en notant $\alpha = \max\{\alpha_1, 2\alpha_2\}$, on a

$$\sum_{i=1}^{N_j-1}\frac{(-u_{i-1,j}+2u_{i,j}-u_{i+1,j})u_{i,j}}{h^2} \geqslant \left(1-\alpha\left(1+\frac{\varepsilon}{2}\right)\right)\sum_{i=0}^{N_j-1}\left|\frac{u_{i+1,j}-u_{ij}}{h}\right|^2 - \frac{1}{2\varepsilon}\sum_{i=0}^{N_j-1}\frac{(u^{\varphi}_{(i,j)-(i+1,j)})^2}{h^2} - \left(1+\frac{\varepsilon}{2}\right)\beta\sum_{i=1}^{N_j-1}\left|\frac{-u_{i-1,j}+2u_{ij}-u_{i+1,j}}{h}\right|^2.$$

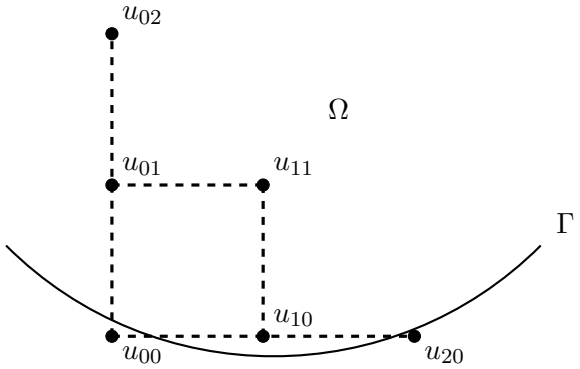

FIGURE 3.3 – Cas $N_j = 2$ dans la preuve de la Proposition 3.1.

Cas $N_j = 2$: On a :

$$\begin{aligned}&(-u_{0,j}+2u_{1,j}-u_{2,j})u_{1,j}\\ &\quad= -(u_{0,j}-u_{1,j})u_{0,j}+(u_{1,j}-u_{2,j})u_{2,j}+|u_{1,j}-u_{0j}|^2+|u_{2,j}-u_{1j}|^2\\ &\leq |(u_{0,j}-u_{1,j})u^{\varphi}_{(0,j)-(1,j)}|+(u_{0,j}-u_{1,j})^2+|(u_{2,j}-u_{1,j})u^{\varphi}_{(1,j)-(2,j)}|+(u_{2,j}-u_{1,j})^2.\end{aligned}$$

On a également $(x_0, y_j), (x_2, y_j) \notin \Omega$ et $(x_1, y_j) \in \Omega$. Le cercle contenant $(0,0)$, $(2h,0)$, $(0,2h)$ est de rayon $\frac{\sqrt{10}}{2}h$. Alors, puisque Ω est *r-smooth*, pour $h < \frac{2r}{\sqrt{10}}$, sans perte de généralité, on peut supposer que l'on se trouve dans la situation représentée à la Figure 3.3. Ainsi, d'après le Lemme 3.2, on obtient la même conclusion que dans le cas précédent.

Conclusion : En combinant les deux cas,

$$a_h(u_h,u_h) \geqslant \left(1-\alpha\left(1+\frac{\varepsilon}{2}\right)\right)\left(\sum_{i,j}\left|\frac{u_{i+1,j}-u_{ij}}{h}\right|^2+\sum_{j,i}\left|\frac{u_{i,j+1}-u_{ij}}{h}\right|^2\right) + \left(1-\frac{1}{2\varepsilon\gamma}\right)b_h(u_h,u_h)+\left(1-\left(1+\frac{\varepsilon}{2}\right)\frac{\beta}{\sigma}\right)j_h(u_h,u_h),$$

ce qui donne le résultat désiré, en prenant ε tel que $\alpha\left(1+\frac{\varepsilon}{2}\right)<1$ et γ, σ suffisamment grands. □

Remarque 3.3. Comme vu précédemment dans la preuve, l'hypothèse du Théorème 3.1 peut être remplacée par les deux hypothèses suivantes :

- Si (x_{i+1},y_j), $(x_{i-1},y_j)\notin\Omega$ et $(x_i,y_j)\in\Omega$ alors, il existe $k,l\in\{-1,1\}$ tel que (x_{i+k},y_{j+l}), (x_{i+k},y_{j+2l}), $(x_i,y_{j+l})\in\Omega$.
- Si (x_i,y_{j+1}), $(x_i,y_{j-1})\notin\Omega$ et $(x_i,y_j)\in\Omega$ alors, il existe $k,l\in\{-1,1\}$ tel que (x_{i+k},y_{j+l}), (x_{i+2k},y_{j+l}), $(x_{i+k},y_j)\in\Omega$.

Pour la preuve du Théorème 3.1, nous aurons également besoin de l'inégalité de Poincaré suivante :

Lemme 3.3. *Il existe $C_P>0$ tel que pour tout $v_h=(v_{ij})_{ij}$,*

$$\|v_h\|^2_{h,\infty,\Omega_h}+\|v_h\|^2_{h,0,\Omega_h}\leq C_P\left(|v_h|^2_{h,1,\Omega_h}+h^3b_h(v_h,v_h)\right).$$

Preuve. Fixons l'indice j et supposons que le premier et le dernier (x_i,y_j) appartenant à Ω_h sont pour $i\in\{M_j,\dots,N_j\}$. Sans perte de généralité, on peut supposer $M_j=0$. Pour tout i, on a

$$v_{ij}=v_{0j}+\sum_{k=0}^{i-1}(v_{k+1,j}-v_{kj}).$$

Alors,

$$v_{ij}^2\leq 2v_{0j}^2+2(i-1)\sum_{k=0}^{i-1}(v_{k+1,j}-v_{kj})^2.$$

En notant L le maximum des diamètres de Ω_h (i.e. la plus grande distance entre deux points de Ω_h), $N_j\leqslant CL/h$ ($C>0$), on en déduit que

$$\sum_{i=0}^{N_j}v_{ij}^2\leq 2C\frac{L}{h}v_{0j}^2+2C^2\frac{L^2}{h^2}\sum_{i=0}^{N_j-1}(v_{i+1,j}-v_{ij})^2.$$

En utilisant (3.14) et (3.15),

$$v_{0,j}^2\leq 2(u^{\varphi}_{(i,j)-(i+1,j)})^2+2(v_{0,j}-v_{1,j})^2,$$

ce qui donne la conclusion. □

Preuve du Théorème 3.1. Démontrons maintenant le Théorème 3.1. Premièrement, on remarque qu'il existe $C_0 > 0$ tel que pour tout $f \in C^2(\Omega)$ et tout $h < h_0$ avec $h_0 > 0$, il existe une extrapolation $\tilde{u} \in \mathcal{C}^4$ de la solution u de (1.1) telle que

$$\|\tilde{u}\|_{\mathcal{C}^4(\overline{\Omega}_h)} \leqslant C_0 \|u\|_{\mathcal{C}^4(\Omega)}.$$

Soit $\tilde{u}$ une telle extrapolation. On note $\tilde{f} = -\Delta\tilde{u}$ et $\tilde{U} = (\tilde{u}_{ij})_{ij} = (\tilde{u}(x_i, y_j))_{ij}$. Enfin, on note $e_{ij} = \tilde{u}_{ij} - u_{ij}$ et $e_h = (e_{ij})_{ij}$. D'après la Proposition 3.1,

$$|||e_h|||_h^2 \leq \frac{1}{c} a_h(e_h, e_h).$$

Puisque u_h est solution de (3.1),

$$a_h(u_h, e_h) = \sum_{ij} f_{ij} e_{ij}.$$

Alors,

$$a_h(e_h, e_h) = \underbrace{-\sum_{i,j|(x_i,y_j)\in\Omega} \left(-\frac{4\tilde{u}_{ij} - \tilde{u}_{i-1,j} - \tilde{u}_{i+1,j} - \tilde{u}_{i,j-1} - \tilde{u}_{i,j+1}}{h^2} - f_{ij} \right) e_{ij}}_{(I)} + \underbrace{b_h(\tilde{U}, e_h)}_{(II)} + \underbrace{j_h(\tilde{U}, e_h)}_{(III)}.$$

Estimons chacun des termes :

Terme (I) : En utilisant l'inégalité de Cauchy-Schwarz,

$$(I) \leqslant \sqrt{\sum_{i,j|(x_i,y_j)\in\Omega} \left(-\frac{4\tilde{u}_{ij} - \tilde{u}_{i-1,j} - \tilde{u}_{i+1,j} - \tilde{u}_{i,j-1} - \tilde{u}_{i,j+1}}{h^2} - f_{ij} \right)^2} \times \sqrt{\sum_{i,j|(x_i,y_j)\in\Omega} e_{ij}^2}.$$

Il existe $(\xi_i, \nu_j) \in [x_i - h, x_i + h] \times [y_j - h, y_j + h]$ tels que

$$-\frac{4\tilde{u}_{ij} - \tilde{u}_{i-1,j} - \tilde{u}_{i+1,j} - \tilde{u}_{i,j-1} - \tilde{u}_{i,j+1}}{h^2} = f_{ij} - \frac{h^2}{12} \left(\frac{\partial^4 \tilde{u}}{\partial x^4}(\xi_i, y_j) + \frac{\partial^4 \tilde{u}}{\partial y^4}(x_i, \nu_j) \right).$$

Puisque le nombre de nœuds dans Ω_h est d'ordre $1/h^2$, on déduit

$$\sqrt{\sum_{i,j|(x_i,y_j)\in\Omega} \left(-\frac{4\tilde{u}_{ij} - \tilde{u}_{i-1,j} - \tilde{u}_{i+1,j} - \tilde{u}_{i,j-1} - \tilde{u}_{i,j+1}}{h^2} - f_{ij} \right)^2} \leqslant \sqrt{\frac{1}{h^2} \times \frac{h^4}{12^2} \|u\|_{\mathcal{C}^4(\Omega)}^2} \leqslant Ch\|u\|_{\mathcal{C}^4(\Omega)}.$$

De plus, d'après le Lemme 3.3,

$$\sum_{i,j|(x_i,y_j)\in\Omega} e_{ij}^2 = \frac{1}{h^2}\|e_h\|_{h,0,\Omega_h}^2 \leq C_P \left(\frac{1}{h^2}\|e_h\|_{h,1,\Omega_h}^2 + h b_h(e_h, e_h) \right) \leq C|||e_h|||_h^2.$$

Alors,

$$(I) \leq Ch\|u\|_{\mathcal{C}^4(\Omega)}|||e_h|||_h.$$

Terme (II) : On considère $w := \tilde{u}/\varphi$. Soit $(x_i, y_j) \in \partial\Omega_h$ tel que $(x_{i+1}, y_j) \in \Omega$. En utilisant les inégalités de Sobolev et de Hardy (cf. [28]),

$$\|w\|_{\mathcal{C}^1([x_i,x_{i+1}])} \leqslant C\|w\|_{2,[x_i,x_{i+1}]} \leqslant C\|\tilde{u}\|_{3,[x_i,x_{i+1}]}.$$

Ainsi

$$\begin{aligned}\left|\frac{\varphi_{(i+1)j}\tilde{u}_{i,j} - \varphi_{ij}\tilde{u}_{i+1,j}}{\sqrt{\varphi^2_{(i+1)j} + \varphi^2_{ij}}}\right| &\leq \left|\frac{\varphi_{(i+1)j}\varphi_{ij}}{\min\{|\varphi_{(i+1)j}|, |\varphi_{ij}|\}}\right| |w(x_i, y_i) - w(x_{i+1}, y_j)| \\ &\leq \max\{|\varphi_{ij}|, |\varphi_{i+1,j}|\}|w(x_i, y_i) - w(x_{i+1}, y_j)| \\ &\leq Ch\|\varphi\|_{L^\infty([x_i,x_{i+1}])}\|w\|_{\mathcal{C}^1([x_i,x_{i+1}])} \leq Ch^2\|\tilde{u}\|_{3,[x_i,x_{i+1}]}.\end{aligned}$$

Puisque le nombre de faces où la ghost penalty est appliquée est d'ordre $\frac{CL}{h}$,

$$\begin{aligned}(II) &\leqslant b_h(\tilde{U}, \tilde{U})^{1/2} b_h(e_h, e_h)^{1/2} \\ &\leqslant \frac{C}{h}\left(\sqrt{\sum_{(i,j)\in B_x}\left|\frac{\varphi_{(i+1)j}\tilde{u}_{i,j} - \varphi_{ij}\tilde{u}_{i+1,j}}{\sqrt{\varphi^2_{(i+1)j} + \varphi^2_{ij}}}\right|^2} + \sqrt{\sum_{(i,j)\in B_y}\left|\frac{\varphi_{i(j+1)}\tilde{u}_{i,j} - \varphi_{ij}\tilde{u}_{i(j+1)}}{\sqrt{\varphi^2_{i(j+1)} + \varphi^2_{ij}}}\right|^2}\right)|||e_h|||_h \\ &\leqslant C\sqrt{h}\|\tilde{u}\|_{3,\overline{\Omega}_h}|||e_h|||_h.\end{aligned}$$

Terme (III) : Une nouvelle fois, puisque le nombre de faces où la ghost penalty est appliquée est d'ordre $\frac{CL}{h}$,

$$\begin{aligned}&\sum_{(i,j)\in J_x} \frac{-\tilde{u}_{i-1,j} + 2\tilde{u}_{ij} - \tilde{u}_{i+1,j}}{h} \times \frac{-e_{i-1,j} + 2e_{ij} - e_{i+1,j}}{h} \\ &\qquad \leqslant Ch\|\tilde{u}\|_{\mathcal{C}^2(\Omega_h)} \sum_{(i,j)\in J_x}\left|\frac{-e_{i-1,j} + 2e_{ij} - e_{i+1,j}}{h}\right| \\ &\qquad \leqslant Ch^{1/2}\|\tilde{u}\|_{\mathcal{C}^2(\Omega_h)}\left(\sum_{(i,j)\in J_x}\left|\frac{-e_{i-1,j} + 2e_{ij} - e_{i+1,j}}{h}\right|^2\right)^{1/2}.\end{aligned}$$

Ainsi,

$$(III) \leq Ch^{1/2}\|\tilde{u}\|_{\mathcal{C}^2(\overline{\Omega}_h)}|||e_h|||_h.$$

En combinant cela avec le Lemme 3.3,on obtient,

$$\|e_h\|_{h,1,\Omega} \leqslant h|||e_h|||_h \leqslant Ch^{3/2}\|u\|_{\mathcal{C}^4(\Omega)}.$$

Enfin, en utilisant une nouvelle fois le Lemme 3.3, on obtient les estimations L^∞ et L^2. □

Démontrons maintenant le Théorème 3.2.

Preuve du Théorème 3.2. D'après la Proposition 3.1 et le Lemme 3.3,

$$a_h(v_h, v_h) \geq C \sum_{(i,j):(x_i,y_j)\in\Omega_h} v_{ij}^2.$$

De plus, grâce à l'expression de a_h,

$$a_h(v_h, v_h) \leq \frac{C}{h^2} \sum_{(i,j):(x_i,y_j)\in\Omega_h} v_{ij}^2,$$

ce qui mène à la conclusion.

□

3.4 Schéma alternatif

Dans cette section, nous proposons une version alternative du schéma (3.1), plus complexe mais offrant un ordre de convergence numérique optimal pour la norme H^1.

En 2D, on considère le schéma suivant : trouver une fonction discrète $u_h = (u_{ij})_{ij}$ définie sur Ω_h telle que

$$\tilde{a}_h(u_h, v_h) = l_h(v_h), \tag{3.16}$$

pour toute fonction discrète $v_h = (v_{ij})_{ij}$ définie sur Ω_h, où

$$\tilde{a}_h(u_h, v_h) = (-\Delta_h u_h, v_h) + \tilde{b}_h(u_h, v_h) + \tilde{j}_h(u_h, v_h),$$

avec

$$\tilde{b}_h(u_h, v_h) = \frac{\gamma}{2h^2}\left(\sum_{ij} \frac{u^{\varphi}_{(i-1,j)-(i+1,j)} \times v^{\varphi}_{(i-1,j)-(i+1,j)}}{4\varphi_{i+1,j}^2\varphi_{i-1,j}^2 + \varphi_{ij}^2\varphi_{i-1,j}^2 + \varphi_{ij}^2\varphi_{i+1,j}^2} + \sum_{ij} \frac{u^{\varphi}_{(i,j-1)-(i,j+1)} \times v^{\varphi}_{(i,j-1)-(i,j+1)}}{4\varphi_{i,j+1}^2\varphi_{i,j-1}^2 + \varphi_{ij}^2\varphi_{i,j-1}^2 + \varphi_{ij}^2\varphi_{i,j+1}^2}\right)$$

et

$$u^{\varphi}_{(i-1,j)-(i+1,j)} := 2\varphi_{i+1}\varphi_{i-1}u_i - \varphi_i\varphi_{i-1}u_{i+1} - \varphi_i\varphi_{i+1}u_{i-1},$$

$u^{\varphi}_{(i,j-1)-(i,j+1)}$ et $v^{\varphi}_{(i,j-1)-(i,j+1)}$ sont définis de manière similaire, et le second terme de stabilisation est donné par

$$\tilde{j}_h(u_h, v_h) = \sigma\Big(\sum_{i,j} \frac{-u_{i-1,j} + 3u_{ij} - 3u_{i+1,j} + u_{i+2,j}}{h} \times \frac{-v_{i-1,j} + 3v_{ij} - 3v_{i+1,j} + v_{i+2,j}}{h} + \sum_{i,j} \frac{-u_{i,j-1} + 3u_{ij} - 3u_{i,j+1} + u_{i,j+2}}{h} \times \frac{-v_{i,j-1} + 3v_{ij} - 3v_{i,j+1} + v_{i,j+2}}{h}\Big). \tag{3.17}$$

Les indices (i, j) dans les sommes sont choisis de sorte que tous les nœuds $(i-1, j)$, (i, j), $(i+1, j)$ et $(i+2, j)$ appartiennent à Ω sauf 1. De même pour $(i, j-1)$, (i, j), $(i, j+1)$ et $(i, j+2)$.

Remarque 3.4. Ce schéma est donné en 2D pour des raisons de lisibilité mais est toujours valable en 3D, en ajoutant les termes correspondant au troisième indice. Des résultats numériques en 2D et en 3D pour ce schéma seront donnés en Section 3.5. Pour ce schéma nous ne proposons pas de résultat de convergence théorique, mais il pourrait être étudié dans une future contribution.

Il est important de préciser comment le terme de pénalisation $\tilde{b}_h$ est obtenu.
En supposant que $u = p\varphi$ avec $p = p_0 + p_1(x - x_i)$ et $u_{ij} = u(x_i, y_j)$, alors

$$\begin{cases} u_{i+1,j} = (p_0 + p_1 h)\varphi_{i+1,j}, \\ u_{ij} = p_0 \varphi_{ij}, \\ u_{i-1,j} = (p_0 - p_1 h)\varphi_{i-1,j}, \end{cases}$$

ce qui donne

$$u^{\varphi}_{(i-1,j)-(i+1,j)} = 0.$$

En ce qui concerne le terme de stabilisation (3.17), $\partial_x u(x_i, y_i)$ peut être approchée à l'ordre 2 par

$$\frac{u(x_{i+1}, y_i) - u(x_{i-1}, y_i)}{2h} \text{ et } \frac{-3u(x_i, y_i) + 4u(x_{i+1}, y_i) - u(x_{i+2}, y_i)}{2h},$$

ce qui donne le saut de $\partial_x u(x_i, y_i)$

$$\frac{-u(x_{i+1}, y_i) + 3u(x_i, y_i) - 3u(x_{i+1}, y_i) + u(x_{i+2}, y_i)}{2h}.$$

Ainsi, (3.17) est une approximation de (3.6).

3.5 Résultats numériques

Dans cette section, nous allons comparer les deux schémas aux différences finies (3.1) et (3.16) avec d'autres approches :

- φ-FEM : pour illustrer l'intérêt de notre nouvelle approche, il est important de la comparer numériquement à φ-FEM. Pour cela, nous utiliserons le schéma (1.10), afin d'illustrer les avantages et inconvénients des méthodes éléments finis par rapport aux approches différences finies ;
- une méthode éléments finis classique : nous comparerons aussi les résultats avec une méthode éléments finis standard conforme (cf. Section 1.1.1, (1.3)) ;
- l'approche Shortley-Weller : finalement, nous comparerons notre méthode à la littérature. Pour cela, nous avons implémenté la méthode Shortley-Weller (SW) [91, 9]. Cette méthode a le même objectif, qui est de traiter les géométries complexes avec des différences finies. Cependant, la matrice associée n'est ici pas bien conditionnée.

Les schémas présentés en Sections 3.1 et 3.4 seront respectivement notés φ-FD et φ-FD2. Les schémas φ-FEM utilisés sont implémentés en `FEniCS` (cf. [60]) et les schémas différences finies à l'aide des librairies `python` classiques : `scipy` [1] [88] et `numpy` [2] [45].

Les codes permettant de reproduire les différents résultats sont disponibles à :

`https://github.com/PhiFEM/PhiFD.git`

Puisque la solution de φ-FD est définie seulement aux nœuds $(x_i, y_j)_{ij}$ et les solutions calculées avec FEM ou SW le sont uniquement sur Ω, les solutions φ-FEM et Standard-FEM seront interpolées aux nœuds $(x_i, y_j)_{ij}$ appartenant à Ω. Les erreurs relatives seront calculées dans les normes $\|\cdot\|_{h,0}$, $\|\cdot\|_{h,\infty}$ et $\|\cdot\|_{h,1}$ définies en Section 3.1.

Il est important de préciser que cette manière de calculer les erreurs pour les méthodes éléments finis peut détériorer les résultats en comparaison à la manière habituelle de le faire. Cependant, cette méthode permet de comparer équitablement les différents schémas.

3.5.1 Premier cas test : un exemple 2D

On considère la solution explicite

$$u = \cos\left(\frac{\pi}{2}r\right)$$

sur le disque centré en $(0.5, 0.5)$ de rayon $R = 0.3 + 1e - 10$, avec

$$r = \frac{1}{R}\sqrt{(x-0.5)^2 + (y-0.5)^2}\,.$$

Ce choix de rayon permet de s'assurer que la frontière exacte coupe une face proche d'un nœud. Dans ce cas, l'approche SW sera mal conditionnée.

Pour le schéma φ-FD, l'ordre de convergence théorique de 3/2 est atteint pour la norme H^1 et on observe une convergence d'ordre 2 pour les normes L^2 et L^∞ (cf. Figures 3.4 et 3.5, gauche et Table 3.1). Le second schéma, φ-FD2, semble moins précis sur des grilles grossières mais légèrement meilleur pour des résolutions plus fines. De plus, l'ordre de convergence optimal quadratique est atteint, en particulier pour la norme H^1. Le conditionnement de la matrice est également optimal avec un ordre $1/h^2$ (cf. Figure 3.5, droite). Le code `python` fait moins de 100 lignes (cf. Annexe A.1) et n'utilise que les librairies `scipy` et `numpy`, ce qui permet un faible temps de calcul (cf. Figure 3.6).

Sur ces figures, il apparaît que φ-FEM et φ-FD ont toutes deux des intérêts dans la résolution d'EDP. En effet, alors que les erreurs en normes L^2 et L^∞ sont relativement proches pour les deux méthodes, l'erreur H^1, le conditionnement ou le temps de calcul sont très différents [3] : l'approche φ-FD est bien plus rapide que l'approche éléments finis, tandis qu'elle conduit à une erreur légèrement plus élevée sur les dérivées de la solution.

1. `https://scipy.org/`
2. `https://numpy.org/`
3. Il est important de prendre en compte ici que les résultats des méthodes éléments finis ont été obtenus avec FEniCS, qui comme nous l'avons déjà précisé précédemment est moins optimisé que la version FEniCSX.

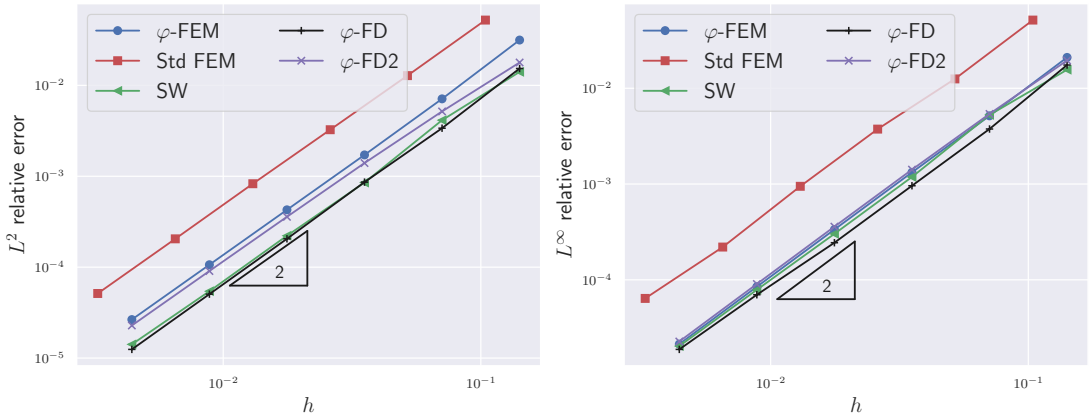

FIGURE 3.4 – **Premier cas test : un exemple 2D.** Erreurs relatives L^2 (gauche) et L^∞ (droite) en fonction de la taille de discrétisation pour φ-FEM, Standard FEM, Shortley-Weller, φ-FD et φ-FD2.

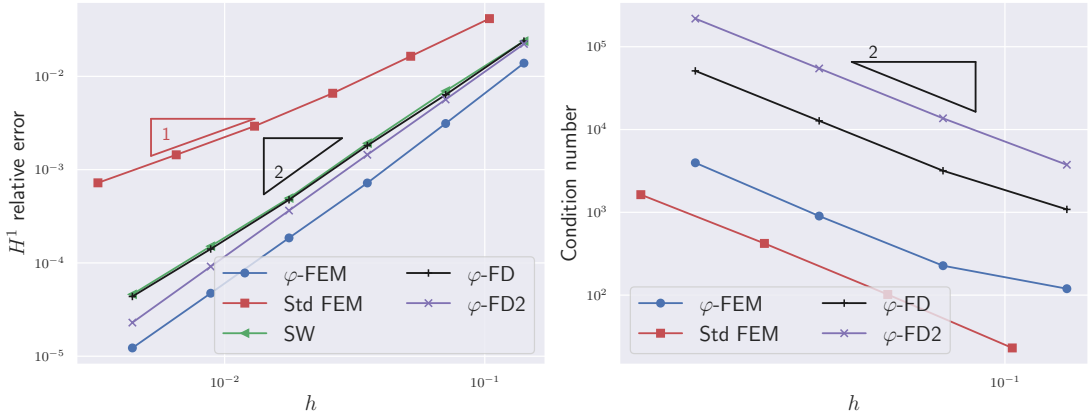

FIGURE 3.5 – **Premier cas test : un exemple 2D.** Erreur relative H^1 (gauche) et conditionnement (droite) en fonction de la taille de discrétisation pour φ-FEM, standard FEM, Shortley-Weller, φ-FD et φ-FD2.

	φ-FEM	Std FEM	SW	φ-FD	φ-FD2
Erreur L^2 relative	2.04	2.0	2.01	2.05	1.93
Erreur L^∞ relative	1.98	1.94	1.95	1.96	1.95
Erreur H^1 relative	2.02	1.17	1.82	1.83	1.98

TABLE 3.1 – **Premier cas test : un exemple 2D.** Ordres de convergence.

De plus, pour les deux schémas φ-FD, on observe le même phénomène de supraconvergence, comme pour la méthode Shortley-Weller. L'ordre de convergence pour la norme H^1 est plus élevé que pour les approches éléments finis : $\mathcal{O}(h^{3/2})$ contre $\mathcal{O}(h)$.

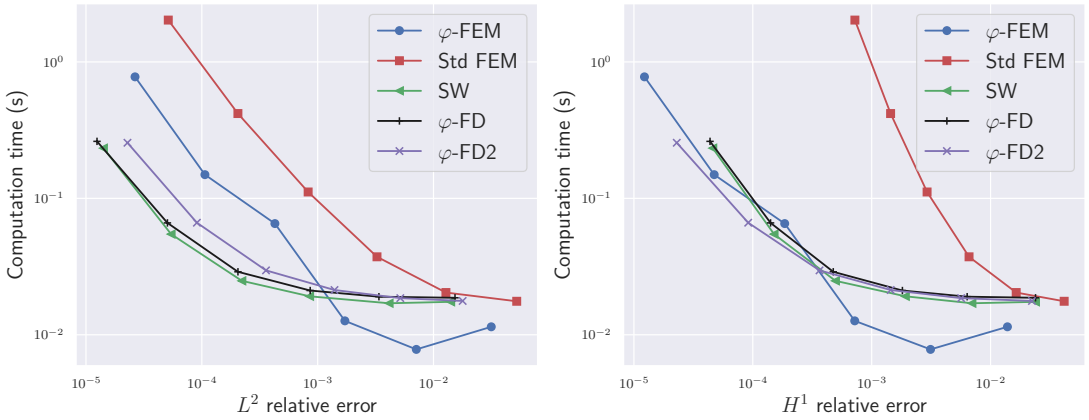

FIGURE 3.6 – **Premier cas test : un exemple 2D.** Temps de calcul en fonction de l'erreur relative L^2 (gauche) et de l'erreur relative H^1 (droite) pour φ-FEM, standard FEM, Shortley-Weller, φ-FD et φ-FD2.

Pour conclure ce cas test et notamment justifier notre choix des paramètres σ et γ, l'évolution de l'erreur relative L^2 et du conditionnement en fonction de ces paramètres est représentée à la Figure 3.7. Ces résultats entraînent le choix de $\sigma = 0.01$ pour les deux schémas et de $\gamma = 1$ pour le premier φ-FD et $\gamma = 10$ pour le second schéma.
On remarque à la Figure 3.7 que l'erreur relative L^2 du second schéma est plus stable aux variations de σ que celle du premier schéma φ-FD, ce qui s'explique par la présence du terme $\tilde{j}_h$ d'ordre 2.

3.5.2 Second cas test : un exemple 3D

On considère maintenant une extension 3D du cas test précédent, i.e. la même solution explicite, dans une sphère centrée en $(0.5, 0.5, 0.5)$, de rayon $R = 0.3$ et

$$r = \frac{1}{R}\sqrt{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2}\,.$$

Dans ce cas également, l'ordre de convergence optimal quadratique est observé en normes L^2 et H^1 (cf. Fig. 3.8). De plus, nos deux schémas différences finies ainsi que l'approche Shortley-Weller donnent de meilleurs résultats que les approches éléments finis. Il est intéressant de noter qu'ici aussi l'approche φ-FEM devient aussi précise en norme H^1 que l'approche différences finies lorsque la taille de discrétisation diminue.

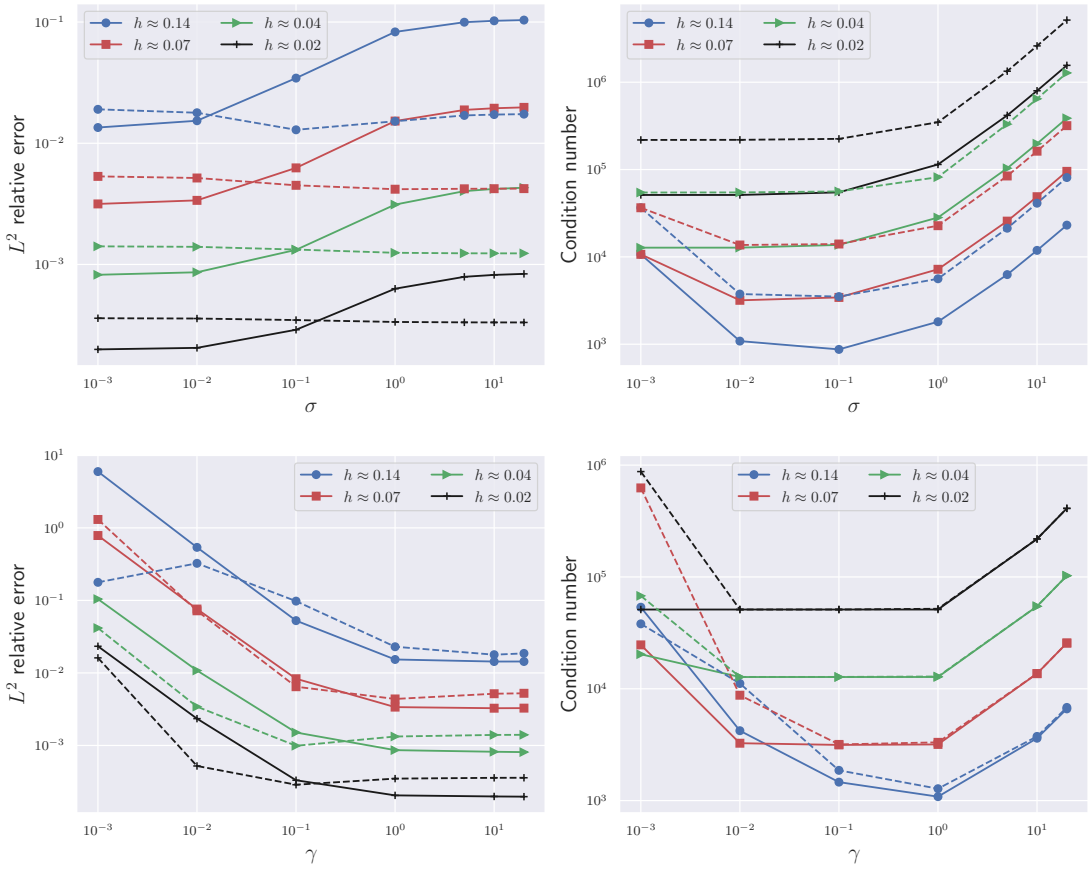

FIGURE 3.7 – **Premier cas test : un exemple 2D.** Haut : évolution de l'erreur relative L^2 (gauche) et du conditionnement (droite), en fonction de σ, avec $\gamma = 1$ pour φ-FD (***traits pleins***) et $\gamma = 10$ pour φ-FD2 (***pointillés***). Bas : évolution de l'erreur relative L^2 (gauche) et du conditionnement (droite), en fonction de γ, avec $\sigma = 0.01$ pour φ-FD (***traits pleins***) et pour φ-FD2 (***pointillés***).

3.5.3 Troisième cas test : combinaison avec une approche multigrid

Un autre avantage des grilles cartésiennes est leur compatibilité avec les solveurs multigrilles (multigrid, [1]) de sorte à améliorer la stabilité et le temps de calcul de la méthode. La méthode multigrid est basée sur la combinaison de schémas de relaxation et d'une hiérarchie particulière de grilles grossières.
Après avoir appliqué une méthode de relaxation sur la grille la plus fine, un terme de correction est obtenu en représentant les résidus interpolés sur la grille grossière suivante et en utilisant une méthode de relaxation. De manière récursive, une hiérarchie de grilles est obtenue et l'algorithme est arrêté lorsque le problème est sur une grille suffisamment grossière, permettant une résolution directe. Dans [34], une description de plusieurs méthodes itératives est proposée : la méthode de Seidel, de Richardson, de Young ou encore une méthode de relaxation ou de minimisation des résidus.

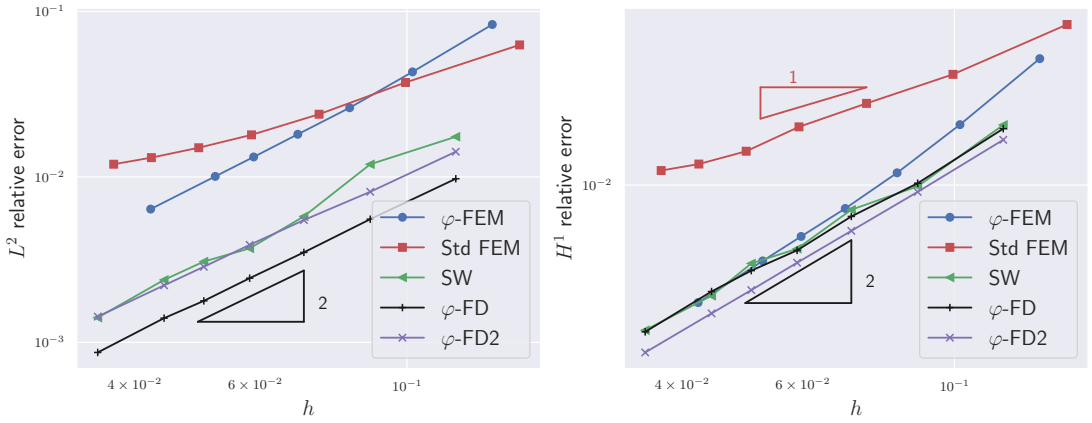

FIGURE 3.8 – **Second cas test : un exemple 3D.** Erreurs relatives L^2 (gauche) et H^1 (droite) en fonction de la taille de discrétisation pour φ-FEM, standard FEM, Shortley-Weller, φ-FD et φ-FD2.

On trouve également une description d'une méthode multigrid pour résoudre l'équation de Poisson sur des domaines généraux avec des exemples numériques dans [41]. Deux composantes importantes des méthodes multigrid sont les opérateurs de restriction et de prolongement qui permettent de transférer les informations entre les différentes grilles. Dans [78], une « sommation par parties » est utilisée, préservant les opérateurs d'interpolation, ce qui permet des approximations précises et stables sur les grilles grossières. Nous allons maintenant proposer une technique similaire à l'approche multigrid permettant un bon compromis entre temps de calcul et erreur.

Pour cela, nous proposons une combinaison de notre schéma φ-FD (3.1) avec une approche multigrid. L'idée est d'utiliser la solution φ-FD obtenue sur une grille grossière, avec un solveur direct, pour initialiser un solveur itératif à une résolution plus fine. L'algorithme sera décomposé en 3 étapes :

1. **Étape 1 : résolution directe sur grille grossière.** On résout une première fois le problème sur une grille grossière N_0^n, obtenant une solution φ-FD grossière u_0, avec un solveur direct.
2. **Étape 2 : interpolation sur une grille fine.** On considère u_1 l'interpolation par splines (d'ordre 2) de u_0 sur une grille fine donnée N_{obj}^n avec $N_{\text{obj}} >> N_0$.
3. **Étape 3 : résolution itérative sur une grille fine.** On calcule une solution φ-FD u_2 sur la grille fine avec un solveur itératif initialisé à u_1.

Dans le cas 2D, nous comparerons cet algorithme avec les deux méthodes suivantes :

- **Méthode directe :** résolution du problème avec un solveur direct sur des grilles de résolutions $N_0 \times N_0$, puis interpolation de la solution sur la grille fine N_{obj}^n. Le solveur utilisé ici est le solveur standard de `scipy`, i.e. un solveur LU.
- **Méthode itérative :** la même méthodologie est appliquée, cette fois avec un solveur itératif, le Gradient BiConjugué Stabilisé.

En 3D, notre méthode sera uniquement comparée à une méthode itérative. Les cas considérés seront les exemples 2D et 3D présentés dans les sous-sections précédentes. La résolution N_{obj} sera fixée à 2200 en 2D et 200 en 3D. Tous les solveurs itératifs ont la même tolérance pour les résidus intérieurs relatifs, fixée à 10^{-4}.

Tous les solveurs itératifs compatibles de la librairie python `scipy` ont été testés mais le Gradient BiConjugué Stabilisé [4] a toujours donné les meilleurs résultats.
Il est d'ailleurs important de noter que le simple gradient conjugué ne peut pas être utilisé ici puisque la matrice A n'est pas symétrique.

Remarque 3.5.

- Un point intéressant est qu'il est possible avec cette approche, d'ajouter une étape intermédiaire, avec une première résolution itérative sur une grille de résolution $N_0 < N_1 < N_{\text{obj}}$, afin de réduire le nombre d'itérations nécessaires lors de la résolution la plus fine. Cependant, sur les cas tests proposés dans cette section, cette approche n'a pas été nécessaire. De plus, ajouter une telle étape augmente le nombre de paramètres à déterminer : tolérance et nombre maximal d'itérations du solveur intermédiaire, taille de la grille intermédiaire, paramètres de l'interpolation intermédiaire.
- Si un schéma φ-FD est développé dans le futur pour résoudre des problèmes non-linéaires, cette approche pourra être appliquée aux itérations d'un algorithme de Newton.
- Dans la Section 5.2, nous proposerons une adaptation de cette idée à la méthode φ-FEM pour différents problèmes, notamment non-linéaires.

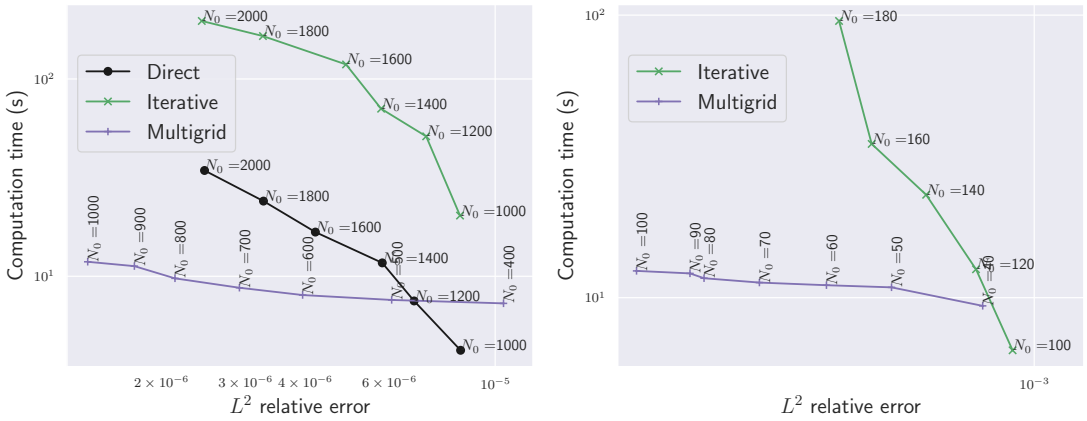

FIGURE 3.9 – **Troisième cas test : approche multigrid.** Temps de calcul en fonction de l'erreur relative L^2 pour les méthodes directe, itérative et multigrid, dans le cas 2D (gauche) et 3D (droite).

Les résultats de la Figure 3.9 (gauche) illustrent l'efficacité de notre approche par rapport aux deux autres méthodes de base : en effet, on atteint une meilleure précision (grâce au solveur itératif final) plus rapidement puisque seulement quelques itérations du

4. https ://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.linalg.bicgstab.html

solveur final sont nécessaires. Sur la Figure 3.9, pour les méthodes « baseline » (discrète et itérative), les valeurs correspondent à la discrétisation utilisée pour la résolution et pour la méthode multigrid, elles correspondent à la discrétisation utilisée pour la résolution grossière. Puisque nous avons choisi d'utiliser l'approche multigrid avec une interpolation de f et de φ de la résolution fine vers la résolution grossière, les temps de calcul ne comportent que les temps de résolution du système linéaire et le temps d'interpolation de u de la résolution N_0 à N_{obj} pour l'approche multigrid.

Comme dit précédemment, un des problèmes de φ-FD, et de toutes les méthodes différences finies, est la croissance de la taille du système linéaire à résoudre, en particulier en 3D : la matrice A contient $(N+1)^6$ valeurs pour une résolution N. Ainsi, il sera presque toujours nécessaire d'utiliser des solveurs itératifs pour résoudre de tels problèmes avec ces approches. Cependant, utiliser un solveur itératif sans solution initiale pour $N = 200$ revient à résoudre un problème avec une matrice A contenant plus de 10^{13} valeurs. Alors, même en utilisant le fait que la matrice est creuse, on obtient un énorme système, extrêmement long à résoudre. Comme illustré à la Figure 3.9 (droite), notre approche permet d'obtenir les résultats de tels problèmes beaucoup plus vite que l'approche naïve, la méthode itérative présentée précédemment.

3.6 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode aux différences finies inspirée par l'approche φ-FEM précédemment présentée, pour la résolution d'EDP elliptiques sur des géométries complexes. La méthode offre différents avantages : les matrices produites par la méthode sont bien conditionnées, ce qui assure une stabilité numérique de la méthode. De plus, le schéma principal proposé atteint des convergences quasi-optimales, comparables aux autres méthodes de la littérature. Enfin, cette nouvelle méthode a l'intérêt d'être compatible avec des approches de type multigrid, ce qui a l'avantage d'améliorer fortement les temps de calcul.

4 Les méthodes éléments finis combinées aux réseaux de neurones

Résumé

Dans ce chapitre, nous proposons une méthode pour résoudre des équations aux dérivées partielles (EDP) en combinant des techniques de Machine Learning et la méthode φ-FEM. Pour cela, nous utilisons le *Fourier Neural Operator* (FNO). L'objectif de ce chapitre est d'introduire cette combinaison et d'illustrer numériquement son intérêt. Nous nous concentrerons ici sur la résolution de deux équations : l'équation de Poisson-Dirichlet et les équations de l'élasticité non linéaire.

L'idée clé de notre méthode est de traiter le scénario complexe des domaines variables, où chaque problème est résolu sur une géométrie différente. Les domaines considérés sont définis par des fonctions *level-set* en raison de l'utilisation de l'approche φ-FEM. Nous présenterons dans un premier temps le FNO puis nous expliquerons notre approche. Nous proposerons ensuite deux autres méthodes : φ-FEM-UNet et Standard-FEM-FNO, combinant réseaux de neurones et méthodes éléments finis. Enfin, nous illustrerons l'efficacité de cette combinaison avec des résultats numériques sur trois cas tests.

Chapitre 4 – Les méthodes éléments finis combinées aux réseaux de neurones

Comme nous l'avons vu dans ce manuscrit, les méthodes éléments finis permettent de résoudre des EDP de manière précise. Cependant, dans certaines applications, il est nécessaire de résoudre ces équations en temps réel ce qui, comme nous l'avons vu n'est pas le cas pour les méthodes classiques. Pour obtenir des résultats en temps réel, de nombreuses méthodes de Machine-Learning ont été développées. Ces méthodes peuvent être séparées en deux groupes :

1. Les méthodes *physics-based* : une des possibilités pour approcher des solutions d'EDP est de minimiser les résidus de l'équation ou la fonctionnelle d'énergie associée à l'équation considérée. Ces méthodes ont alors l'avantage de ne pas nécessiter des approximations obtenues par exemple par des méthodes éléments finis. La méthode la plus populaire de cette catégorie est la méthode PINNs [76] mais on peut également trouver des méthodes telles que les méthodes Deep Ritz [29] ou Deep Galerkin [83]. Cependant, malgré la promesse initiale de ces méthodes, on trouve maintenant de nombreuses illustrations numériques indiquant que ces méthodes ne sont pas meilleures que les méthodes classiques tant en termes de temps de calcul que de précision, par exemple dans [40].
2. Les méthodes *data-based* : une seconde catégorie regroupe les méthodes utilisant des méthodes type éléments finis pour générer une base de données, permettant alors d'entraîner un réseau de neurones. Cette étape d'entraînement, bien que lourde en termes de calcul et de temps, peut être faite en amont des simulations, pendant une étape préparatoire. L'intérêt est alors de pouvoir obtenir la solution pour de nouvelles données de manière quasi-instantanée. De nombreux exemples tels que U-Net (voir par exemple [77]), les Graph Neural Operator [57], les DeepOnet [61] et les Fourier Neural Operator (FNO) [58, 56] ont démontré de très bonnes performances.

Dans ce chapitre, nous allons principalement nous concentrer sur le FNO qui s'est montré supérieur aux autres méthodes en rapport coût-précision (cf. [58]). L'inconvénient des FNO est la nécessité de grilles cartésiennes afin d'effectuer des FFT (Fast Fourier Transform), ce qui limite l'implémentation initiale à des problèmes posés sur des domaines rectangulaires. Plusieurs approches ont été proposées pour adapter la méthode à des géométries plus générales, par exemple l'approche Geo-FNO [56] où le domaine d'entrée est déformé en un maillage uniforme latent sur lequel les FFT peuvent être appliquées. Nous proposons ici une approche alternative : la géométrie sera encodée par une fonction level-set et associée aux autres données du problème. Cela nous permettra alors d'utiliser la méthode φ-FEM pour générer des données.
Nous allons détailler notre nouvelle méthode φ-FEM-FNO et la comparer notamment à deux autres approches : φ-FEM-UNet et Standard-FEM-FNO, ce qui permettra de justifier l'utilisation du FNO par rapport à un autre réseau ainsi que l'utilisation de φ-FEM par rapport à une méthode standard.

Nous nous concentrerons ici sur la résolution de deux équations, posées sur des géométries complexes : l'équation de Poisson avec conditions de Dirichlet

$$\begin{cases} -\Delta u &= f\,, \quad \text{dans } \Omega\,, \\ u &= g\,, \quad \text{sur } \Gamma\,, \end{cases} \tag{4.1}$$

et les équations de l'élasticité non-linéaire :

$$\begin{cases} -\operatorname{div} \boldsymbol{\Pi}(\boldsymbol{u}) &= \boldsymbol{f}\,, \quad \text{dans } \Omega\,, \\ \boldsymbol{u} &= \boldsymbol{u}_D\,, \quad \text{sur } \Gamma_D\,, \\ \boldsymbol{\Pi}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \boldsymbol{t}\,, \quad \text{sur } \Gamma_N\,, \end{cases} \tag{4.2}$$

avec Ω un domaine de $\mathbb{R}^d, d = 1, 2, 3$, de frontière $\Gamma = \Gamma_D \cup \Gamma_N$ avec $\Gamma_D \cap \Gamma_N = \emptyset$ pour (4.2).

4.1 La méthodologie φ-FEM-FNO

Dans cette section, nous avons choisi de nous concentrer sur le cas de l'équation de Poisson (4.1) afin de simplifier les écritures et notations utilisées. Les différences liées au passage à l'équation (4.2) seront présentées en préambule du cas test numérique associé à la résolution de cette équation.

4.1.1 Idée générale

Notre idée est de construire un réseau de neurones qui sera une approximation de l'opérateur qui associe les données f et g ainsi que la géométrie du problème à la solution u de (4.1). On souhaite que la solution obtenue soit précise mais, obtenue avec un coût de calcul limité, en particulier le plus rapidement possible. L'objectif est d'entraîner ce réseau à l'aide de données synthétiques générées par un solveur discret (par exemple une méthode éléments finis classique, ou φ-FEM). Dans cette section, nous utiliserons φ-FEM comme méthode de génération de données.

Pour cette approche, nous avons choisi d'utiliser le Fourier Neural Operator, introduit dans [58] et [52], reposant sur une architecture itérative proposée dans [57]. Ce choix a été motivé par plusieurs raisons : dans le cas de l'approximation de solutions d'EDP, les auteurs de [58] ont illustré que les performances du FNO étaient meilleures que de nombreuses autres approches. De plus, les FNOs pourront être utilisés pour différentes équations sans grande modification de l'architecture. Enfin, les FNO et φ-FEM sont compatibles puisque les deux méthodes utilisent des grilles cartésiennes.

4.1.2 L'opérateur "ground truth"

Dans la suite de ce chapitre, par analogie aux approximations éléments finis, sauf mention explicite du contraire, f_h, g_h, φ_h, u_h et w_h représenteront les matrices de $\mathbb{R}^{n_x \times n_y}$ associées aux approximations $\mathbb{P}^1$ des fonctions f, g, φ, u et w, composée pour chaque indice $i = 0, \ldots, n_x - 1$, $j = 0, \ldots, n_y - 1$, des valeurs des évaluations ou des extrapolations

dans $V_h^{\mathcal{O}}$ aux nœuds du maillage $\mathcal{T}_h^{\mathcal{O}}$ de coordonnées (x_i, y_j), avec $x_i := i/(n_x - 1)$, $y_j := j/(n_y - 1)$, où

$$V_h^{\mathcal{O}} := \{v_h \in H^1(\mathcal{O}) \;\; v_h|_T \in \mathbb{P}^k(T) \;\forall\; T \in \mathcal{T}_h^{\mathcal{O}}\}\,. \tag{4.3}$$

Dans la tradition de la littérature FNO (et réseaux de neurones en général), le FNO va approcher un opérateur appelé l'opérateur "ground truth", qui sera noté $\mathcal{G}^\dagger$. Dans notre cas, $\mathcal{G}^\dagger$ sera l'opérateur associant f_h, g_h, et la géométrie encodée par φ_h, à la solution φ-FEM w_h :

$$\begin{array}{rccc} \mathcal{G}^\dagger : & \mathbb{R}^{n_x \times n_y \times 3} & \to & \mathbb{R}^{n_x \times n_y \times 1} \\ & (f_h, \varphi_h, g_h) & \mapsto & w_h\,. \end{array} \tag{4.4}$$

Remarque 4.1. Il est important de noter que w_h est extrapolée par 0 en dehors de Ω_h, sans impact sur le FNO puisque ces valeurs ne seront pas vues par la fonctionnelle à minimiser que nous définirons par la suite. En pratique cette extrapolation sera faite par FEniCSX ([5, 80, 79, 3]).

4.1.3 Structure du FNO

Il est maintenant nécessaire de présenter quelques aspects essentiels à la compréhension de l'architecture d'un FNO. Plus de détails au sujet des Neural Operators en général ont été proposés dans [57], et en particulier au sujet du FNO dans [58, 52, 56].

Le principe sera de construire une application paramétrique

$$\begin{array}{rccc} \mathcal{G}_\theta : & \mathbb{R}^{n_x \times n_y \times 3} & \to & \mathbb{R}^{n_x \times n_y \times 1}\,, \\ & (f_h, \varphi_h, g_h) & \mapsto & w_\theta\,, \end{array}$$

qui approche l'opérateur $\mathcal{G}^\dagger$ défini par (4.4). On cherche ainsi à prédire une approximation w_θ de w_h qui nous permette de reconstruire $u_\theta = \varphi_h w_\theta + g_h$, une approximation de $u_h = \varphi_h w_h + g_h$ en suivant le paradigme φ-FEM, comme illustré à la Figure 4.1. La variable θ représente l'ensemble des paramètres que l'on devra obtenir par minimisation d'une fonctionnelle.

Remarque 4.2. Le choix de prédire w_h plutôt que directement la solution u_h provient du fait que multiplier la prédiction par φ_h permet d'imposer plus précisément les conditions de bord. En effet, prédire directement u_h introduira une erreur supplémentaire au bord. Cependant, l'utilisation de cette approche est restreinte aux situations où l'on considère le schéma direct φ-FEM. Ainsi, lors de l'utilisation du schéma dual (2.2) ou dans le cas de conditions mixtes, cette approche ne sera pas utilisable et il sera nécessaire de prédire la solution directement. Lors des simulations numériques à la Section 4.4, nous illustrerons pour le premier cas test la différence entre les deux approches : prédire u_h et prédire w_h.

L'application $\mathcal{G}_\theta$ est composée de plusieurs applications intermédiaires, appelées couches et est définie par

$$\mathcal{G}_\theta = N^{-1} \circ Q_\theta \circ \mathcal{H}_\theta^4 \circ \mathcal{H}_\theta^3 \circ \mathcal{H}_\theta^2 \circ \mathcal{H}_\theta^1 \circ P_\theta \circ N\,.$$

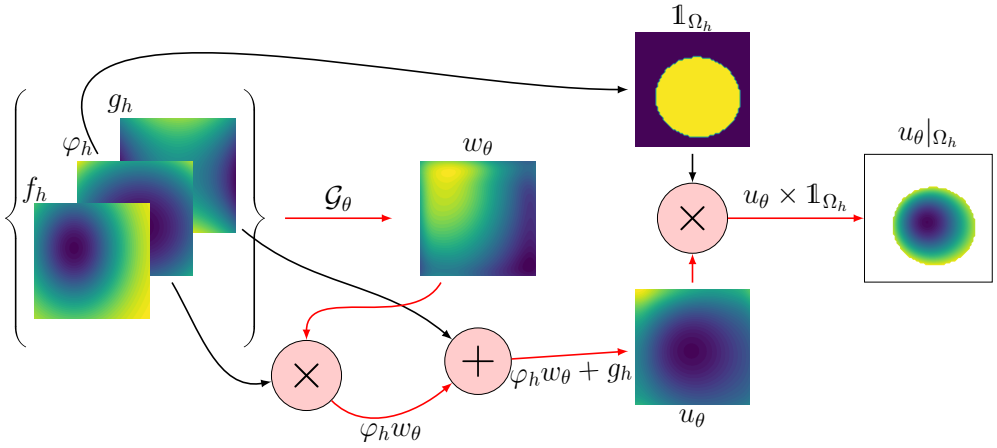

FIGURE 4.1 – Construction d'une prédiction de φ-FEM-FNO pour la résolution de (4.1).

Dans le cas du problème de Poisson 2D que nous considérons, chacune de ces couches agit sur des tenseurs 3D dont la troisième dimension (le nombre de *canaux*) varie entre les couches. Plus précisément, la structure est la suivante :

$$\mathcal{G}_\theta : \mathbb{R}^{n_x \times n_y \times 3} \xrightarrow{N} \mathbb{R}^{n_x \times n_y \times 3} \xrightarrow{P_\theta} \mathbb{R}^{n_x \times n_y \times n_d} \xrightarrow{\mathcal{H}^1_\theta} \mathbb{R}^{n_x \times n_y \times n_d} \xrightarrow{\mathcal{H}^2_\theta} \\ \dots \xrightarrow{\mathcal{H}^4_\theta} \mathbb{R}^{n_x \times n_y \times n_d} \xrightarrow{Q_\theta} \mathbb{R}^{n_x \times n_y \times 1} \xrightarrow{N^{-1}} \mathbb{R}^{n_x \times n_y \times 1},$$

où n_d est une dimension suffisamment élevée. Une représentation graphique (adaptée de [58]) de l'opérateur $\mathcal{G}_\theta$ est donnée à la Figure 4.2. Les transformations P_θ et Q_θ sont respectivement un *embedding* dans un espace de dimension élevée et une projection dans l'espace de dimension désirée, toutes deux effectuées par des couches denses (cf. [58]).

Normalisations N et N^{-1} Pour améliorer les performances des réseaux de neurones, il est bien connu que la normalisation des entrées et sorties du réseau est presque obligatoire (cf. [71] par exemple). Nous allons donc appliquer une normalisation canal par canal, notée N et une dé-normalisation N^{-1}. En effet, pour améliorer les performances de nos implémentations du FNO, puisque les valeurs des données peuvent être très différentes (notamment entre f et φ), nous avons décidé de normaliser les données et les sorties, comme dans [58].

L'opérateur de normalisation est appliqué indépendamment, canal par canal pour chacun des canaux de l'image X. Pour chaque canal C de X, en notant C^{train} l'ensemble des valeurs du même canal sur le sous-ensemble de données d'entraînement, la normalisation est donnée par

$$N_C(C) = \left(\frac{C - \text{mean}(C^{\text{train}})}{\text{std}(C^{\text{train}})} \right),$$

où la moyenne et l'écart-type sont calculés uniquement sur Ω_h. L'opérateur inverse est lui donné par :

$$N^{-1}(Y) = Y \times \text{std}(Y^{\text{train}}) + \text{mean}(Y^{\text{train}}),$$

où Y correspond à un canal de la sortie du FNO et Y^{train} est le vecteur composé des solutions de l'opérateur « ground truth » sur les données d'entraînement.

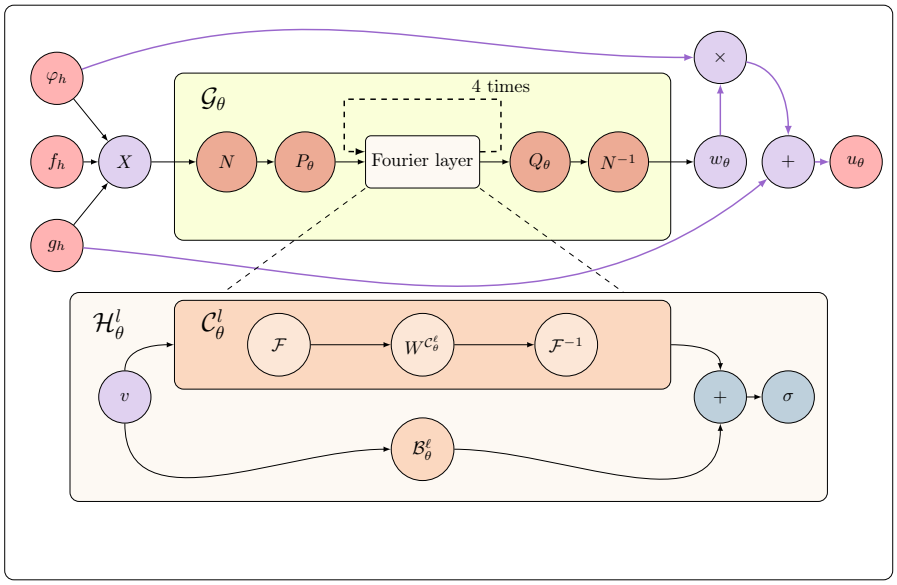

FIGURE 4.2 – Représentation graphique de la pipeline φ-FEM-FNO pour l'approximation de solutions de (4.1), adaptée de [58]. La partie supérieure représente la pipeline entière et la partie inférieure une représentation plus détaillée d'une couche de Fourier. Les cercles rouges correspondent aux entrées données au réseau et à la solution en sortie de φ-FEM-FNO. On représente les entrées et sorties vues par le FNO dans des cercles violets, avec en particulier $X = (f_h, \varphi_h, g_h)$. De plus, les flèches noires correspondent à des étapes internes du FNO et les flèches violettes à des étapes effectuées en dehors du FNO.

Structures des couches P_θ et Q_θ La transformation P_θ est composée d'une couche *fully connected* avec n_d neurones agissant sur chaque nœud, c'est-à-dire, pour tous $i \in \{1, ..., n_x\}$, $j \in \{1, ..., n_y\}$ et $k \in \{1, ..., n_d\}$,

$$P_\theta(X)_{ijk} = \sum_{k'=1}^{3} W^{P_\theta}_{kk'} X_{ijk'} + B^{P_\theta}_k \,,$$

avec $W^{P_\theta} \in \mathcal{M}_{n_d,3}(\mathbb{R})$, $B^{P_\theta} \in \mathbb{R}^{n_d}$ des paramètres à optimiser.

La transformation Q_θ est composée de deux couches *fully connected* de tailles n_Q et 1, agissant également sur chaque nœud. La première dimension n_Q est choisie plus élevée que n_d. La combinaison de ces deux couches permet finalement d'obtenir une solution plus lisse qu'avec une seule couche permettant de passer de la dimension n_d à la dimension finale souhaitée, ici 1.
Ainsi, $Q_\theta = (Q_{\theta,ijk})_{ijk}$ est définie pour tout $X = (X_{ijk})_{ijk}$ par, pour $i \in \{1, ..., n_x\}$, $j \in \{1, ..., n_y\}$,

$$Q_\theta(X)_{ij} = \left[\sum_{k=1}^{n_Q} W_{1k}^{Q_\theta,2}\sigma\left(\sum_{k'=1}^{n_d} W_{kk'}^{Q_\theta,1}X_{ijk'} + B_k^{Q_\theta,1}\right)\right] + B^{Q_\theta,2}\,,$$

avec $W^{Q_\theta,1} \in \mathcal{M}_{n_d,n_Q}(\mathbb{R})$, $B^{Q_\theta,1} \in \mathbb{R}^{n_Q}$, $W^{Q_\theta,2} \in \mathcal{M}_{n_Q,1}(\mathbb{R})$, $B^{Q_\theta,2} \in \mathbb{R}$ des paramètres à optimiser et σ une fonction d'activation appliquée terme à terme. Pour notre approche, nous avons choisi la fonction GELU (Gaussian Error Linear Unit) donnée par $f(x) = x\varphi(x)$ avec $\varphi(x) = P(X \leqslant x)$ où $X \sim \mathcal{N}(0,1)$, comme dans l'implémentation originelle du FNO [1] et de sa variante Geo-FNO [2].

Structure des couches de Fourier $\mathcal{H}_\theta^\ell$ Chaque couche $\mathcal{H}_\theta^\ell$ est constituée de deux applications (cf. [58]) :

$$\mathcal{H}_\theta^\ell(X) = \sigma(\mathcal{C}_\theta^\ell(X) + \mathcal{B}_\theta^\ell(X))\,,$$

où

- $\mathcal{C}_\theta^\ell$ est définie par

$$\mathcal{C}_\theta^\ell(X) = \mathcal{F}^{-1}\Big(W^{\mathcal{C}_\theta^\ell}\mathcal{F}(X)\Big) \in \mathbb{R}^{n_x\times n_y\times n_d\times n_d}\,,$$

  avec $W^{\mathcal{C}_\theta^\ell} \in \mathbb{C}^{n_x\times n_y\times n_d\times n_d}$ une matrice de paramètres à optimiser et $\mathcal{F}$, $\mathcal{F}^{-1}$ la FFT réelle et son inverse, définies par :
  Pour tous $i \in \{1,...,n_x\}$, $j \in \{1,...,n_y\}$ et $k \in \{1,...,n_d\}$,

$$\mathcal{F}(X)_{ijk} = \sum_{i'j'} X_{i'j'k}e^{2\sqrt{-1}\pi\left(\frac{ii'}{n_x}+\frac{jj'}{n_y}\right)}\,,$$

  et pour $Y \in \mathbb{C}^{n_x\times n_y\times n_d}$

$$\mathcal{F}^{-1}(Y)_{ijk} = \sum_{i'j'} Y_{i'j'k}e^{-2\sqrt{-1}\pi\left(\frac{ii'}{n_x}+\frac{jj'}{n_y}\right)}\,.$$

- $\mathcal{B}_\theta^\ell = (\mathcal{B}_{\theta,ijk}^\ell)_{ijk}$ est une couche de biais définie pour tout $X = (X_{ijk})_{ijk}$ par :
  Pour $i \in \{1,...,n_x\}$, $j \in \{1,...,n_y\}$ et $k \in \{1,...,n_d\}$,

$$\mathcal{B}_\theta^\ell(X)_{ijk} = \sum_{k'=1}^{n_d} W_{kk'}^{\mathcal{B}_\theta^\ell}X_{ijk'} + B_k^{\mathcal{B}_\theta^\ell}\,,$$

  avec $W^{\mathcal{B}_\theta^\ell} \in \mathcal{M}_{n_d}(\mathbb{R})$ et $B^{\mathcal{B}_\theta^\ell} \in \mathbb{R}^{n_d}$.

Les coefficients de $W^\cdot$ et $B^\cdot$ composent la quasi-totalité des paramètres à optimiser. Ces paramètres sont soumis à deux contraintes théoriques :

1. https://github.com/neuraloperator/neuraloperator
2. https://github.com/neuraloperator/Geo-FNO

- Pour que la matrice $\mathcal{C}_\theta^\ell(X)$ soit une matrice réelle, on doit imposer à $W^{\mathcal{C}_\theta^\ell}$ une contrainte de symétrie hermitienne, c'est-à-dire $W^{\mathcal{C}_\theta^\ell}_{n_x-i,n_y-j,k} = \overline{W}^{\mathcal{C}_\theta^\ell}_{i,j,k}$. En pratique, puisque l'on utilise une implémentation particulière de la FFT, la Real-FFT, les coefficients de Fourier sont stockés dans des matrices de taille $n_x \times (n_y/2+1)$ et sont automatiquement symétrisés lors de la FFT inverse. En pratique, il n'y a donc pas de précautions particulières à prendre lors de cette étape.
- Les solutions du problème (4.1) sont en général très lisses. Ainsi, lorsque l'on applique la RFFT, les hautes fréquences ne servant qu'à assurer la bijectivité, peuvent être négligées. On ne gardera ainsi que les $m \times m$ premiers coefficients de Fourier, correspondant aux basses fréquences.

Remarque 4.3. Un aspect intéressant du FNO est le nombre de paramètres à optimiser. En effet, puisque l'on tronque les hautes fréquences, pour chaque couche $\mathcal{C}_\theta^l$ le nombre de paramètres est moins élevé que $n_x \times n_y \times n_d \times n_d$. En particulier, le nombre de paramètres n_θ est indépendant de la résolution des données d'entrée et est donné par

$$n_\theta = \overbrace{4 \times n_d}^{P_\theta \,:\, 3\times n_d + n_d} + 4 \times \underbrace{(\overbrace{2 \times n_d^2 \times m^2}^{\mathcal{C}_\theta^l} + \overbrace{n_d^2 + n_d}^{\mathcal{B}_\theta^l})}_{\mathcal{H}_\theta^l} + \overbrace{(n_d+2) \times n_Q + 1}^{Q_\theta \,:\, n_d \times n_Q + n_Q \times 1 + 1}.$$

Par exemple, pour le premier cas test qui suivra, pour les paramètres choisis, cela représentera 324577 paramètres à optimiser.

Remarque 4.4. Une fois entraînés, les FNO peuvent être utilisés avec des nouvelles données pour des résolutions arbitraires n_x, n_y. Cette propriété de multi-résolution est due à la structure du FNO utilisant les FFT. Cependant, cette propriété n'est pas directement compatible avec l'approche φ-FEM de par la variation des domaines construits en fonction des résolutions considérées.

Remarque 4.5 (Phénomène de Gibbs et padding). Un problème usuel de la RFFT appliquée à des fonctions non périodiques est le phénomène de Gibbs : des oscillations apparaissent au bord. Pour effacer ces oscillations, on peut utiliser des techniques de *padding* : on étend les matrices en ajoutant des valeurs tout autour (i.e. on ajoute des couches de pixels aux images) avant d'effectuer les calculs. À la fin, on restreint les matrices à leurs dimensions originales. Il existe différentes méthodes de padding dans la littérature, mais nous n'utiliserons ici que la méthode de padding réflective (c.f. la documentation de PyTorch [3] pour un exemple). Il est intéressant de noter que puisque ces phénomènes n'apparaissent qu'au bord du domaine, dans les deux premiers cas test numériques que nous considérerons, le padding n'est pas nécessaire. Cependant, pour le troisième cas test, nous en aurons obligatoirement besoin.

4.1.4 Choix de la *loss function*

Nous allons maintenant présenter la fonctionnelle que nous avons choisi de minimiser pour l'approximation des solutions du problème (4.1).

3. `https://pytorch.org/docs/stable/generated/torch.nn.ReflectionPad2d.html`

Le choix de cette fonction, appelée *loss function* sera très important pour assurer une bonne précision et variera en fonction du problème considéré. Un choix de fonctionnelle adaptée à la résolution de (4.2) sera proposé en Section 4.4.

Par construction, une prédiction du FNO sera donnée sur la même grille cartésienne que les données d'entrée. Cependant, dans notre approche, les seules valeurs qui nous intéressent sont les valeurs de la solution sur Ω_h. Il est donc nécessaire de définir une fonction n'agissant que sur les pixels correspondants. Un exemple de données et de solution (restreintes à Ω_h) est représenté à la Figure 4.6.

Soit N_{data} la taille d'un échantillon de données. On note $U_{\text{true}} = (u^n_{\text{true}})_{n=0,\dots,N_{\text{data}}}$ où $u^n_{\text{true}} = \varphi^n_h w^n_h + g^n_h$, la solution *ground truth* et $U_\theta = (u^n_\theta)_{n=0,\dots,N_{\text{data}}}$ avec

$$u^n_\theta = \varphi^n_h \mathcal{G}_\theta(f^n_h, \varphi^n_h, g^n_h) + g^n_h = \varphi^n_h w^n_\theta + g^n_h$$

la solution φ-FEM-FNO.

La fonction à optimiser est une approximation de l'erreur moyenne H^1 sur les données considérées (cf. Figure 4.11 pour une justification numérique de ce choix), donnée par

$$\mathcal{L}\left(U_{\text{true}}; U_\theta\right) = \frac{1}{N_{\text{data}}} \sum_{n=0}^{N_{\text{data}}} \left(\mathcal{E}_0(u^n_{\text{true}}; u^n_\theta) + \mathcal{E}_1(u^n_{\text{true}}; u^n_\theta)\right) , \tag{4.5}$$

où

$$\mathcal{E}_0(u^n_{\text{true}}; u^n_\theta) = \|u^n_{\text{true}} - u^n_\theta\|^2_{0,\mathcal{S}^n_0} ,$$

et

$$\mathcal{E}_1(u^n_{\text{true}}; u^n_\theta) = \|\nabla^h_x u^n_{\text{true}} - \nabla^h_x u^n_\theta\|^2_{0,\mathcal{S}^n_1} + \|\nabla^h_y u^n_{\text{true}} - \nabla^h_y u^n_\theta\|^2_{0,\mathcal{S}^n_1} ,$$

avec ∇^h l'approximation du gradient par différences finies centrées et $\mathcal{S}_0$ est l'ensemble de pixels correspondant aux nœuds de $\mathcal{T}_h$. Enfin, $\mathcal{S}_1$ est l'ensemble des pixels de $\mathcal{S}_0$ privé d'une couche de pixels (construit en utilisant le 8-voisinage, cf. Figure 4.3 pour un exemple).

Remarque 4.6. Dans l'expression de la *loss function* (4.5), l'erreur est calculée par rapport à u^n_{true} et non w^n_{true}. Cependant, cela ne signifie pas pour autant que le FNO sera entraîné à prédire u^n_θ. Cela signifie seulement que l'opérateur sera entraîné à prédire une solution w_θ qui, multipliée par φ_h et ajoutée à g_h, sera proche de u^n_{true}. Nous illustrerons numériquement dans la Section 4.4 l'intérêt de prédire w_h par rapport à u_h quand cela est possible.

4.2 Trois autres approches

Nous avons pour l'instant considéré uniquement le cas de la méthode φ-FEM combinée à un FNO. Cependant, cette combinaison n'est évidemment pas la seule combinaison possible. Nous allons maintenant proposer deux autres variantes qui semblent également très intéressantes. Il est également intéressant de présenter la méthode Geo-FNO, proposée dans [56] proposant une autre solution permettant d'appliquer des FNO à des géométries complexes.

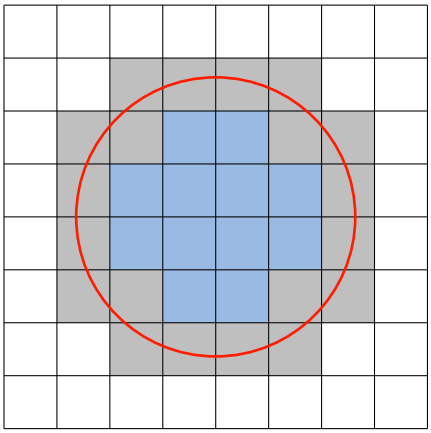

FIGURE 4.3 – En rouge, la frontière exacte d'un domaine circulaire. En couleurs (bleu et gris), $\mathcal{S}_0$. En gris uniquement, l'ensemble $\mathcal{S}_1$.

4.2.1 La méthode Geo-FNO

Pour adapter le FNO à des géométries complexes, une approche a été proposée dans [56]. Cette méthode, Geo-FNO, a notamment surpassé sur différents cas test numériques la méthode DeepONet [61]. Cependant, son architecture est plus complexe et lourde que l'architecture classique FNO. En effet, pour traiter les géométries complexes, tout en conservant la structure du FNO utilisant les FFT, les auteurs proposent de construire une transformation entre l'espace physique (la géométrie considérée, donnée par exemple sous la forme d'un ensemble de coordonnées de nœuds d'un maillage) et un espace latent, construit comme une grille cartésienne. Une fois cette transformation appliquée, il est alors possible d'appliquer un FNO classique pour déterminer une solution dans l'espace latent. Finalement, l'inverse de la première transformation est appliquée à la solution « latente », ce qui permet d'obtenir la solution dans l'espace physique. Cette transformation, dans l'idéal un difféomorphisme, peut être très complexe. Par exemple (cf. [56]), une telle transformation peut être construite à l'aide de polynômes de Tchebychev. Cependant, en pratique, la transformation sera souvent apprise par un réseau de neurones. Ainsi, cela ajoute de nombreux paramètres à optimiser et plusieurs couches au réseau, ce qui augmente la complexité et le coût (d'entraînement et d'inférence) de la méthode.

4.2.2 La combinaison φ-FEM-UNet

Il est naturel de s'interroger sur le choix du réseau à utiliser. En effet, le FNO est parfaitement compatible avec l'approche φ-FEM, mais d'autres méthodes bien connues telles que les UNet [77] le sont également. Nous avons donc adapté notre approche à un réseau de type UNet que nous allons présenter.

Le U-Net, introduit dans [77], est une architecture de réseau de neurones convolutifs conçue à l'origine pour la segmentation d'images. L'innovation principale du U-Net est sa structure en forme de "U", composée d'un chemin contractant (encodeur, la partie « descendante ») pour capturer le contexte et d'un chemin expansif (décodeur,

la partie « ascendante »). Contrairement aux architectures classiques de type encodeur-décodeur, le U-Net inclue des *skip connections* entre les couches correspondantes du chemin descendant et du chemin ascendant, permettant de préserver les informations spatiales fines, i.e. correspondant aux hautes fréquences tronquées dans le FNO. Le UNet est parfois aujourd'hui adapté et utilisé pour approcher des solutions d'EDP notamment dans [65] ou dans [30].

Architecture du U-Net La présentation du UNet que nous allons effectuer correspond comme nous l'avons fait pour le FNO à la version que nous avons implémentée numériquement.

On considère une image d'entrée $X \in \mathbb{R}^{n_x \times n_y \times n_d}$ avec $n_d = 3$, correspondant aux entrées (f_h, φ_h, g_h). L'objectif sera ici de construire un opérateur :

$$\mathcal{G}_\theta^{\text{UNet}}(f_h, \varphi_h, g_h) \to w_\theta \,. \tag{4.6}$$

Cet opérateur prendra donc en entrée une image X, comme pour φ-FEM-FNO et donnera une approximation w_θ de la solution w_h en sortie. L'architecture du réseau U-Net utilisé est représentée à la Figure 4.4. Ce réseau est construit comme une suite de couches de convolutions, où chaque étape « down » est une suite de convolutions, de *max pooling* (sauf pour la dernière étape) et de fonctions d'activation (ici ReLu), et chaque couche « Up » est également construite comme une combinaison de convolutions, en associant les étapes de *skip connection* représentées en pointillés sur la Figure 4.4. On indique également sur cette figure les dimensions des tenseurs en sortie de chacune des couches.

Finalement, pour notre approche φ-FEM-UNet, nous appliquons la même pipeline que celle représentée à la Figure 4.1, à la seule différence que l'opérateur $\mathcal{G}_\theta$ sera remplacé par $\mathcal{G}_\theta^{\text{UNet}}$. La fonctionnelle $\mathcal{L}$ à minimiser sera également définie par (4.5).

Remarque 4.7. L'un des avantages du réseau UNet est sa structure maintenant bien connue. En effet, de nombreuses évolutions ont été proposées dans la littérature pour améliorer les performances et pourraient donc être utilisées en combinaison avec φ-FEM. Nous avons ici choisi de nous concentrer sur la version la plus simple de ce réseau, puisque pour l'approche FNO, nous avons également considéré la version la plus simple. Il est important de noter que l'inconvénient de l'implémentation de UNet proposée par rapport au FNO est le nombre de paramètres à optimiser. En effet, en comparaison aux ≈ 325000 paramètres pour le FNO, dans le cas du UNet, 7753025 paramètres sont à optimiser.

4.2.3 La méthode Standard-FEM-FNO

Une autre approche envisageable est la combinaison d'un FNO avec une méthode éléments finis classique. En effet, comme nous l'avons dit précédemment cela peut introduire une erreur d'interpolation dans les résultats mais, ne rend pas la combinaison impossible. De plus, en construisant une méthodologie adaptée, cette erreur d'interpolation sera approximativement du même ordre que l'erreur de prédiction du réseau de neurones.

Pour cette approche, nous allons garder l'idée d'encoder la géométrie par une fonction level-set.

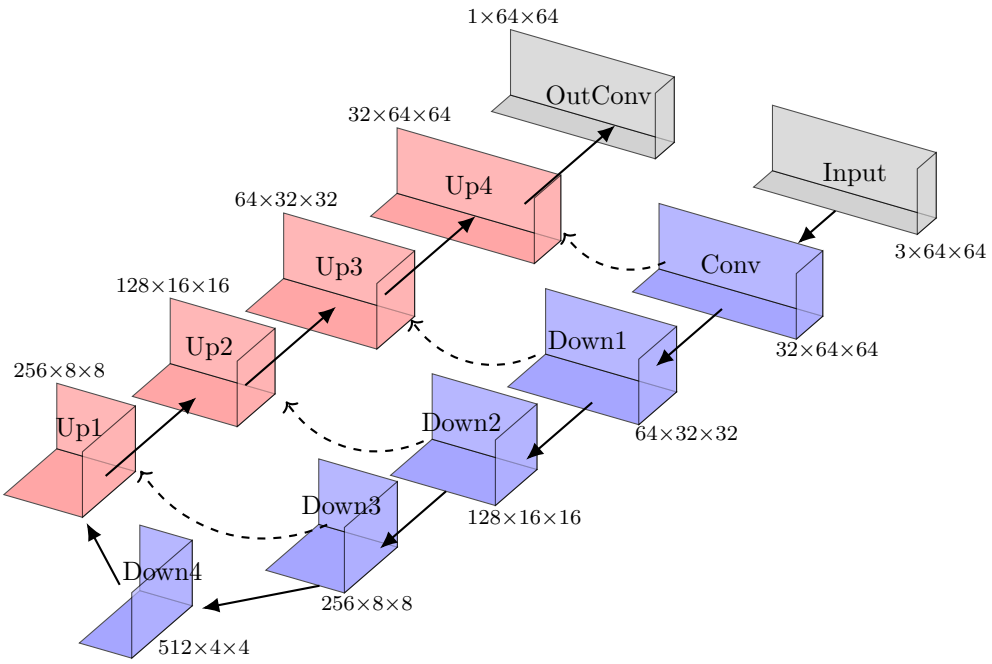

FIGURE 4.4 – Représentation de l'architecture UNet utilisée.

Ainsi, il sera nécessaire de générer des maillages conformes précis à partir de fonctions level-set pour construire la base de données d'entraînement, ce qui augmentera inévitablement le temps de génération de données par rapport aux approches basées sur φ-FEM. Pour cela, nous utiliserons l'approche proposée en Section 5.1.1. Les données seront alors générées par des simulations éléments finis classiques sur les maillages générés avant d'être extrapolées sur l'espace $V_h^{\mathcal{O}}$, défini par (4.3). Cependant, c'est notamment à cette étape qu'une première erreur d'interpolation sera introduite, puisqu'il sera nécessaire de passer des nœuds du maillage aux nœuds de la grille cartésienne. De plus, sur les nœuds de la grille cartésienne extérieurs au maillage conforme, proches de la frontière de ce dernier, la solution sera prolongée, de sorte à obtenir une solution sur tous les pixels du masque utilisé pour le FNO (l'ensemble $\mathcal{S}_0$ représenté à la Figure 4.3).

La structure du FNO sera la même que celle présentée précédemment, à la différence que l'on construira cette fois une application paramétrique

$$\begin{array}{rccc} \mathcal{G}_\theta^{std} : & \mathbb{R}^{n_x \times n_y \times 3} & \to & \mathbb{R}^{n_x \times n_y \times 1}, \\ & (f_h, \varphi_h, g_h) & \mapsto & u_\theta, \end{array} \tag{4.7}$$

qui approchera

$$\begin{array}{rccc} \mathcal{G}_{std}^{\dagger} : & \mathbb{R}^{n_x \times n_y \times 3} & \to & \mathbb{R}^{n_x \times n_y \times 1} \\ & (f_h, \varphi_h, g_h) & \mapsto & u_h, \end{array}$$

où u_h représente ici l'approximation de la solution éléments finis conformes, extrapolée sur $V_h^{\mathcal{O}}$.

Pour l'entraînement de cet opérateur, la *loss function* utilisée sera définie par (4.5), où u^n_{true} sera maintenant donnée par u^n_h et $u^n_\theta = \mathcal{G}^{std}_\theta(f^n_h, \varphi^n_h, g^n_h)$, sera la solution prédite par l'opérateur Standard-FEM-FNO (abrégé par la suite en « Std-FEM-FNO »).

4.3 Détails d'implémentation

Pour l'entraînement des différents modèles, nous avons toujours utilisé la même structure algorithmique et le même algorithme d'optimisation : une méthode ADAM, avec un *learning rate* initial $\alpha = 0.0005$, et des paramètres $\beta_1 = 0.9$, $\beta_2 = 0.999$ et $\varepsilon = 10^{-7}$ pour entraîner les FNO (cf. Algorithme 1). Pendant les entraînements, le *learning rate* est réduit lorsque la fonction $\mathcal{L}$ évaluée sur le jeu de données de validation ne diminue pas pendant plusieurs itérations. La boucle d'entraînement utilisée est détaillée à l'Algorithme 2, pour le cas de l'équation (4.1). Dans l'Algorithme 2, (F^i, φ^i, G^i) représente un batch de données. Les batches sont sélectionnés aléatoirement, tels que $F^i = (f^k_h)_{k\in K_i}$, $\varphi^i = (\varphi^k_h)_{k\in K_i}$, $G^i = (g^k_h)_{k\in K_i}$ avec K_i un ensemble d'indices aléatoires tels que $i \in \{1, \dots, \text{nombre de batches}\}$. Les ensembles K_i sont eux construits tels que $K_i \cap K_j = \emptyset$ pour $i \neq j$.

Algorithme 1 : Étape de l'algorithme ADAM.

Entrées : t, θ_{t-1}, β_1, β_2, ε, m_{t-1}, v_{t-1}.

1 Calculer le gradient : $g_t \leftarrow \nabla f(\theta_{t-1})$

2 Mise à jour du moment (« momentum update ») :

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t\,, \qquad v_t \leftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2)\cdot g_t \cdot \bar{g}_t$$

3 Correction :

$$\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}\,, \qquad \hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$$

4 Mise à jour des paramètres :

$$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t}+\varepsilon}\cdot \hat{m}_t - w_1\theta_{t-1}$$

Remarque 4.8 (Calibrage du *learning rate.*). Le *learning rate* est un paramètre critique à déterminer pour obtenir des résultats précis. Nous n'avons pas inclus de résultats illustrant notre choix de ce paramètre, mais une étude numérique a été réalisée pour déterminer un paramètre « optimal ». Une valeur trop élevée ou diminuant trop lentement entraîne généralement des grandes oscillations des valeurs de la fonctionnelle et donc une mauvaise convergence. À l'inverse, des valeurs trop faibles ou diminuant trop rapidement entraînent une convergence lente, parfois vers un minimum local très éloigné des résultats souhaités.

Pour éviter de telles situations, il a été nécessaire d'effectuer de multiples entraînements afin de déterminer la valeur la plus adaptée à notre situation. De plus, nous avons sélectionné un *learning rate scheduler* permettant de faire évoluer ce paramètre, offrant

également les meilleurs résultats. Pour cela, le learning rate évoluera ainsi tout au long de l'entraînement en fonction des valeurs de la fonctionnelle évaluée sur l'échantillon de validation.

Pour d'autres variantes d'évolution du learning rate, un travail annexe effectué lors d'une SEME et qui figure à l'annexe B pourrait être adapté à cette situation afin d'améliorer la convergence, en particulier diminuer le nombre d'itérations nécessaires à l'obtention de résultats satisfaisants.

Algorithme 2 : Algorithme d'entraînement utilisé.

Entrées : θ_0 : paramètres aléatoires initiaux, $X = (F, \varphi, G)$ et Y_{true} : les données d'entraînement, batch_size : la taille de batch, λ : paramètre de régularisation.

pour $t = 1, \ldots,$ *nombre d'epochs* **faire**

  pour $i = 1, \ldots,$ *nombre de batch* **faire**

    Sélectionner un batch $(F^i, \varphi^i, G^i) \subset X$ et $Y^i_{\text{true}} \subset Y_{\text{true}}$ de taille batch_size.

    Évaluer le modèle : $Y_\theta = \mathcal{G}_{\theta_{ti-1}}(F^i, \varphi^i, G^i)$.

    Calculer la loss :

$$\mathcal{L}(Y^i_{\text{true}}, Y_\theta) + \underbrace{\frac{\lambda}{2 \times \text{batch\_size}} \sum_j |w_j|^2}_{\text{régularisation } L^2} .$$

    Calculer le gradient de la loss, par rapport aux paramètres θ_{ti-1} :

$$\nabla_{\theta_{ti-1}} \mathcal{L} .$$

    Étape d'optimisation : application de l'Algorithme 1.

  } Entraînement

  Soient $(F_{\text{val}}, \varphi_{\text{val}}, G_{\text{val}})$ et Y_{val} la partie de validation du jeu de données.

  Évaluer le modèle sur l'échantillon de validation :

$$Y_\theta = \mathcal{G}_{\theta_{ti}}(F_{\text{val}}, \varphi_{\text{val}}, G_{\text{val}}) .$$

  Calculer la loss : $\mathcal{L}(Y_{\text{val}}, Y_\theta)$.

  Mise à jour du learning rate.

} Validation

4.4 Simulations numériques

Nous allons maintenant illustrer l'efficacité de notre méthode φ-FEM-FNO avec différents cas test numériques. Dans un premier temps, nous allons considérer l'équation de Poisson (4.1), sur des géométries simples données par des ellipses aléatoires, pour illustrer la précision et la rapidité de notre méthode, comparée à plusieurs autres techniques et

en particulier à φ-FEM-UNet et Std-FEM-FNO. Nous étendrons ensuite notre étude à des géométries plus complexes. Enfin, nous considérerons un problème d'élasticité non-linéaire, l'équation (4.2).

Dans les différents cas test, le paramètre n_d (nombre de neurones agissant sur chaque nœud) sera fixé à 20, le nombre de neurones pour la première couche de Q_θ, n_Q sera lui fixé à 128. Enfin, on conservera les $m = 10$ premiers coefficients de Fourier dans les approches utilisant un FNO.

Comme pour les précédentes simulations éléments finis réalisées dans ce manuscrit, les données ont été générées avec la librairie DOLFINx ([5, 80, 79, 3]). Les réseaux (FNO et UNet) ont été implémenté avec la librairie *Pytorch*[75] [4].

Métriques d'évaluation Pour évaluer les performances des différentes méthodes, on définit deux métriques différentes, correspondant à deux versions de l'erreur relative L^2 :

- La première métrique, utilisée pour calculer l'erreur entre 2 tenseurs, i.e. l'erreur entre une prédiction φ-FEM-FNO et une solution *ground truth*, est définie par :

$$E_1(u_{\text{true}}, u_\theta) := \sqrt{\frac{\mathcal{E}_0(u_{\text{true}}; u_\theta)}{\mathcal{N}_0(u_{\text{true}})}}\,, \tag{4.8}$$

  où $u_\theta = \varphi_h \mathcal{G}_\theta(\varphi_h, f_h, g_h) + g_h$, $u_{\text{true}} = \varphi_h w_h + g_h$ et $\mathcal{N}_0(u_{\text{true}}) = \|u_{\text{true}}\|^2_{0,\mathcal{S}_0}$. On notera également $\mathcal{L}_0(\cdot)$ la moyenne de cette métrique sur un ensemble de données.

- La seconde métrique, utilisée pour calculer les erreurs des différentes méthodes par rapport à une solution de référence éléments finis u_{ref} est donnée par :

$$E_2(u_{\text{ref}}, u_\theta) := \frac{\|\Pi_{\Omega_{\text{ref}}} u_\theta - u_{\text{ref}}\|_{0,\Omega_{\text{ref}}}}{\|u_{\text{ref}}\|_{0,\Omega_{\text{ref}}}} = \sqrt{\frac{\int_{\Omega_{\text{ref}}} (\Pi_{\Omega_{\text{ref}}} u_\theta - u_{\text{ref}})^2 \,\mathrm{d}x}{\int_{\Omega_{\text{ref}}} u_{\text{ref}}^2 \,\mathrm{d}x}}\,, \tag{4.9}$$

  où $\Pi_{\Omega_{\text{ref}}}$ est une approximation de la projection orthogonale L^2 sur le domaine de référence Ω_{ref} (domaine recouvrant le maillage $\mathcal{T}_h^{\text{ref}}$, maillage fin conforme sur Ω).

4.4.1 L'équation de Poisson-Dirichlet sur des ellipses aléatoires

Considérons premièrement le cas de l'équation (4.1) sur des domaines définis par les fonctions level-set

$$\varphi_{(x_0,y_0,l_x,l_y,\theta)}(x,y) = -1 + \frac{((x-x_0)\cos(\theta) + (y-y_0)\sin(\theta))^2}{l_x^2} + \frac{((x-x_0)\sin(\theta) - (y-y_0)\cos(\theta))^2}{l_y^2}\,, \tag{4.10}$$

avec

$$x_0,\ y_0 \sim \mathcal{U}([0.2, 0.8])\,, \quad l_x,\ l_y \sim \mathcal{U}([0.2, 0.45]) \quad \text{et } \theta \sim \mathcal{U}([0, \pi])\,.$$

4. Les codes et données correspondant à ces cas test sont disponibles à l'adresse `https://github.com/KVuillemot/PhiFEM_and_FNO`.

L'équation (4.10) permet de construire une ellipse de centre (x_0, y_0) de semi-grand axe l_x et semi-petit axe l_y, orientée d'un angle θ de centre (x_0, y_0). Pour ce cas test, les données sont générées en utilisant une méthode de rejet (dite *rejection sampling method*) sur les paramètres permettant de s'assurer que chaque domaine construit est bien totalement inscrit dans le carré unité. Les fonctions f et g de (4.1) sont données par

$$f_{(A,\mu_0,\mu_1,\sigma_x,\sigma_y)}(x,y) = A \exp\left(-\frac{(x-\mu_0)^2}{2\sigma_x^2} - \frac{(y-\mu_1)^2}{2\sigma_y^2}\right), \tag{4.11}$$

et

$$g_{(\alpha,\beta)}(x,y) = \alpha\left((x-0.5)^2 - (y-0.5)^2\right)\cos(\beta y \pi), \tag{4.12}$$

où $A \sim \mathcal{U}([-30,-20] \cup [20,30])$, $(\mu_0, \mu_1) \sim \mathcal{U}([0.2, 0.8]^2 \cap \{\varphi < -0.15\})$, σ_x, $\sigma_y \sim \mathcal{U}([0.15, 0.45])$ et α, $\beta \sim \mathcal{U}([-0.8, 0.8])$.

On génère un jeu de données de taille 2100, séparé en une partie pour l'entraînement, composée de 1500 données, une partie pour la validation de taille 300 et une partie de test de taille 300 également, toutes sur des grilles de résolution 64×64. Durant l'entraînement, le jeu de données d'entraînement est lui divisé en *batches* (sous-ensembles aléatoires) de taille 32 (correspondant à un ensemble de données considérées pour une évaluation de $\mathcal{L}$) à chacune des 2000 *epochs* (époques d'entraînement, c'est-à-dire le nombre total de boucles parcourant l'ensemble des *batches*), comme décrit à l'Algorithme 2.

Remarque 4.9 (Génération de données). Pour la génération des données, on utilise des éléments finis $\mathbb{P}^1$ et des interpolations $\mathbb{P}^2$ des fonctions f et φ, puisque l'on considère qu'à cette étape, on peut utiliser un maximum d'informations. Cependant, lors des comparaisons de méthodes qui vont suivre, nous utiliserons uniquement des interpolations $\mathbb{P}^1$ pour une comparaison honnête des méthodes puisque les approches basées sur les réseaux de neurones utilisent uniquement les valeurs aux nœuds.

Résultats sur les données de validation Dans un premier temps, on représente à la Figure 4.5 (gauche) l'évolution de la fonctionnelle à minimiser, $\mathcal{L}$ évaluée sur un sous-ensemble aléatoire (de taille 300) des données d'entraînement ainsi que sur les données de validation, ce qui illustre que la fonctionnelle décroit sur les deux ensembles de données. De plus, on représente à la Figure 4.5 (droite), l'évolution de $\mathcal{L}_0$ sur les mêmes ensembles de données. Cette représentation permet de remarquer deux choses : la *loss function* choisie semble adaptée au problème puisque la métrique d'intérêt (i.e. l'erreur relative L^2) décroit également.

À la fin des 2000 étapes d'entraînement, on sélectionne le modèle « optimal » correspondant à l'ensemble de paramètres minimisant $\mathcal{L}$ sur le jeu de données de validation. Ce modèle sera utilisé par la suite pour les comparaisons de méthodes.

Validation du modèle sur des données de test Il est maintenant nécessaire d'évaluer l'erreur (4.8) du FNO sur un jeu de données de test.
Cela permettra de vérifier que l'opérateur est bien entraîné et parvient à donner de bons résultats sur de nouvelles données, et donc se comporte de la même façon que sur les données de validation.

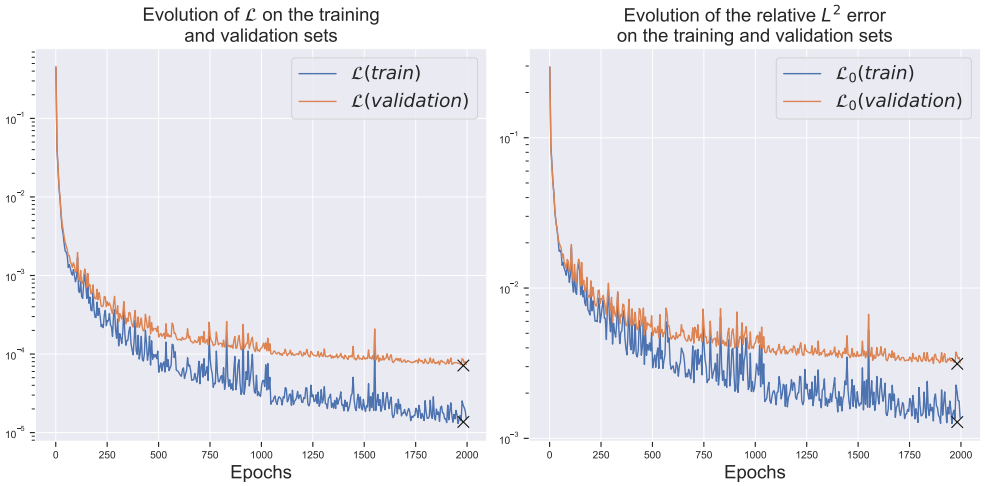

FIGURE 4.5 – **Cas test 1.** À gauche (resp. droite), on représente l'évolution de $\mathcal{L}$ (resp. la moyenne de l'erreur relative L^2, $\mathcal{L}_0$) sur un sous-ensemble de l'échantillon d'entraînement et sur les données de validation.

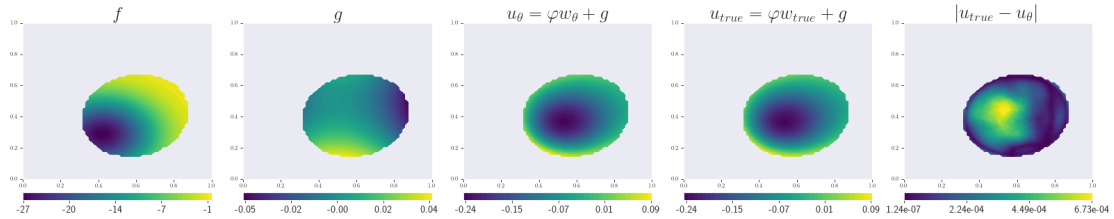

FIGURE 4.6 – **Cas test 1.** Exemple de données et de solution correspondante, restreintes à Ω_h, avec une erreur (4.8) de 2.5×10^{-3}, correspondant à l'erreur médiane sur les données de validation.

Cela permettra également de s'assurer que le modèle sélectionné précédemment est optimal parmi tous ceux testés. Pour cela, on génère un nouvel ensemble de 2500 données et on calcule l'erreur relative L^2 pour plusieurs modèles intermédiaires de l'entraînement, ainsi que pour le modèle optimal choisi. Les résultats présentés à la Figure 4.7, semblent bien confirmer que le modèle choisi est optimal parmi ceux considérés.

Comparaison de φ-FEM-FNO avec d'autres approches Nous allons maintenant conclure ce cas test par les résultats les plus importants pour confirmer l'intérêt de notre approche par rapport aux méthodes suivantes :

- φ-FEM : on applique l'opérateur « ground truth » $\mathcal{G}^\dagger$, sur des grilles de résolutions 64×64 (correspondant à une taille de cellule $h \approx 0.022$), avec $\sigma_D = 1$ et des éléments finis $\mathbb{P}^1$;
- Standard FEM : on utilise une méthode éléments finis classique, avec des éléments $\mathbb{P}^1$ sur des maillages avec $h \approx 0.022$;

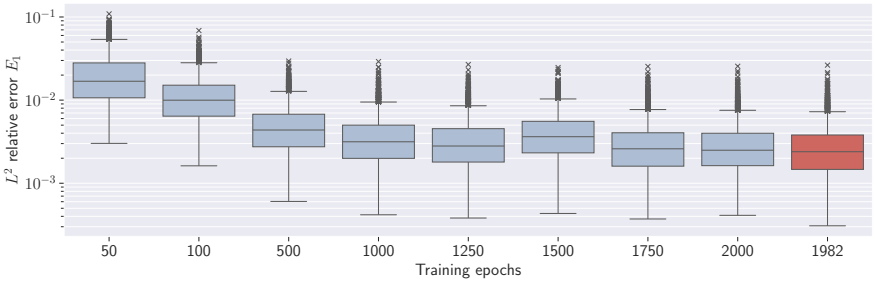

FIGURE 4.7 – **Cas test 1.** Erreurs (4.8) sur 2500 données de test. Le dernier modèle, en rouge, correspond au modèle optimal sélectionné.

- φ-FEM-FNO : le modèle optimal sélectionné précédemment (entraîné avec 1500 données sur des données de taille 64×64) ;
- φ-FEM-FNO 2 : on applique la méthodologie présentée pour φ-FEM-FNO mais en prédisant cette fois directement la solution u_θ plutôt que w_θ.
  On construit alors l'opérateur

$$\begin{aligned} \mathcal{G}_\theta : \quad \mathbb{R}^{n_x \times n_y \times 3} &\to \mathbb{R}^{n_x \times n_y \times 1}, \\ (f_h, \varphi_h, g_h) &\mapsto u_\theta, \end{aligned}$$

  que l'on entraîne en utilisant les mêmes données et la même fonctionnelle $\mathcal{L}$, définie par (4.5), avec u_θ^n la prédiction du réseau ;
- φ-FEM-UNet : on entraîne l'opérateur (4.6) avec une nouvelle fois la même fonctionnelle (4.5), pendant 2000 itérations.
- Standard-FEM-FNO : on entraîne l'opérateur (4.7) pendant 2000 itérations, avec des données $\mathbb{P}^1$ générées sur des maillages avec $h \approx 0.022$, à partir des mêmes paramètres.
- Geo-FNO : on entraîne un opérateur Geo-FNO (c.f. Section 4.2.1), en adaptant l'approche de [56] (c.f. l'implémentation originale sur GitHub [5]) à notre situation. Pour cela, on génère à partir des mêmes paramètres un jeu de données sur des maillages composés de 1053 nœuds (correspondant au nombre moyen de nœuds sur les maillages considérés pour l'approche Standard-FEM-FNO). Pour l'entraînement, la fonctionnelle donnant les meilleurs résultats dans ce cas est l'erreur relative L^2. C'est donc celle choisie pour entraîner l'opérateur utilisé ici.

Les différents opérateurs ont été entraînés avec des ensembles de données construits à partir des mêmes paramètres. De plus les différents opérateurs FNO ont été entraînés avec les mêmes hyper-paramètres. Enfin, pour générer les maillages conformes nécessaires pour les approches Standard-FEM, Standard-FEM-FNO, Geo-FNO ainsi que pour les maillages de référence utilisés pour déterminer les solutions de référence, nous avons utilisé la méthode de construction présentée en Section 5.1.1.

5. `https://github.com/neuraloperator/Geo-FNO/blob/main/elasticity/elas_geofno_v2.py`

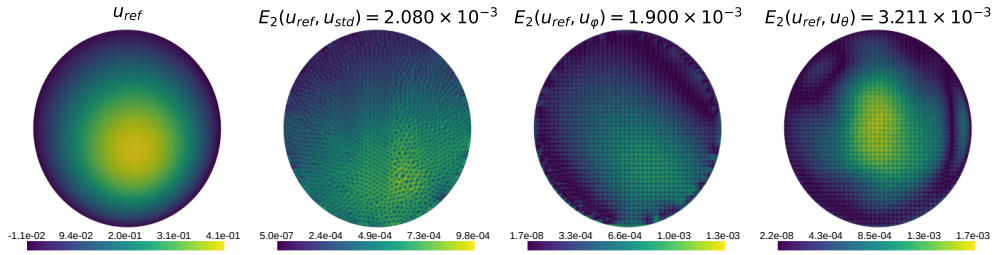

FIGURE 4.8 – **Cas test 1.** De gauche à droite : solution de référence, puis différences entre la solution de référence et la projection de la solution Standard-FEM (u_{std}), de la solution φ-FEM (u_φ), et de la prédiction (φ-FEM-FNO u_θ).
Le cas test présenté correspond au cas donnant l'erreur relative L^2 médiane.

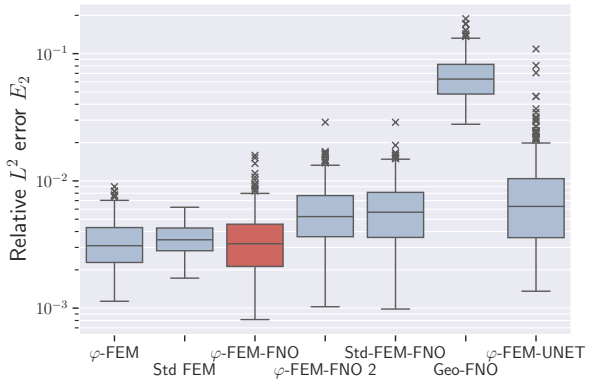

FIGURE 4.9 – **Cas test 1.** Erreurs relatives L^2 pour chaque méthode.

Pour comparer ces différentes approches, on considère un ensemble de 300 données de test. Les solutions déterminées (prédites par les opérateurs et calculées par les méthodes éléments finis) sont projetées sur des maillages de référence avec des tailles de cellules $h_{\text{ref}} \approx 0.005$, comme illustré à la Figure 4.8. On calcule ensuite les erreurs selon la norme (4.9), avec une solution éléments finis calculée sur le maillage fin pour solution de référence. Les résultats présentés à la Figure 4.9 permettent d'illustrer que l'opérateur φ-FEM-FNO parvient à déterminer des solutions avec une précision proche de celle des méthodes éléments finis. De plus, φ-FEM-FNO est près de 2 fois plus précise que Standard-FEM-FNO, et 10 fois plus que Geo-FNO. On remarque également que l'approche φ-FEM-FNO donne de meilleurs résultats que φ-FEM-UNet, illustrant l'intérêt du FNO par rapport à un classique UNet.
En effet, bien que les performances de φ-FEM-UNet soient relativement intéressantes, au regard des résultats présentés, on peut supposer que pour des performances équivalentes en termes de précision, il serait nécessaire d'utiliser plus de données et d'entraîner plus longtemps le UNet. Enfin, on remarque que les performances de φ-FEM-FNO-2, bien que légèrement inférieures à celles de φ-FEM-FNO, restent toutefois meilleures que celles de Standard-FEM-FNO et Geo-FNO.

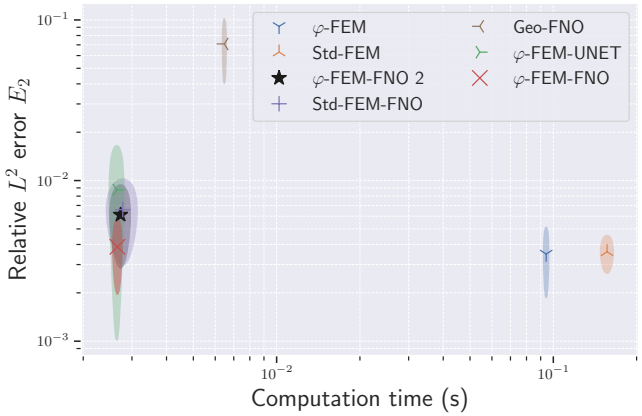

FIGURE 4.10 – **Cas test 1.** Erreurs relatives L^2 en fonction du temps de calcul (en secondes).

On choisit également de comparer un point important : le ratio erreur-temps de calcul. Pour cela, on mesure le temps de chacune des méthodes. Pour φ-FEM, le temps total comprend les temps de sélection et de construction des maillages $\mathcal{T}_h$ et $\mathcal{T}_h^\Gamma$ (en incluant le temps de génération du maillage cartésien), le temps d'interpolation des fonctions f, φ et g, l'assemblage de la matrice éléments finis et le temps de résolution du système linéaire. Pour Standard-FEM, le temps total comprend le temps de génération du maillage, les interpolations de f et g, l'assemblage de la matrice éléments finis et la résolution du système linéaire. Enfin, pour les autres méthodes, on mesure le temps d'inférence de chaque modèle. On représente alors les résultats à la Figure 4.10, où chaque marqueur a pour abscisse l'erreur moyenne et pour ordonnée le temps moyen (en secondes). Les régions de couleurs ont pour largeur l'écart type du temps de calcul et pour hauteur l'écart type de l'erreur, ce qui permet d'illustrer la variabilité de chaque quantité mesurée. Les résultats illustrent clairement le gain de temps apporté par l'utilisation de méthodes de Machine Learning, comparées aux méthodes éléments finis. En particulier, on remarque que les résultats de φ-FEM-FNO, qui sont comparables en termes de précision aux résultats FEMs, sont obtenus environ 100 fois plus vite.

Dans le Tableau 4.1, on compare les temps de calcul de chaque méthode. Pour les méthodes de machine learning, la première colonne contient les temps de génération des bases de données pour chaque méthode ; la deuxième colonne correspond au temps moyen d'une itération de l'entraînement de chaque méthode et la troisième colonne est le temps total des 2000 itérations de chaque entraînement. Enfin, pour l'ensemble des méthodes, la dernière colonne contient le temps moyen pour obtenir une solution, mesuré comme précédemment pour la Figure 4.10.

Méthode	Génération	Une *epoch*	Entraînement	Inférence
φ-FEM	\	\	\	0.095
Standard FEM	\	\	\	0.156
φ-FEM-FNO	214.2	2.2	4400	0.002
φ-FEM-FNO-2	219.8	2.2	4400	0.002
Std-FEM-FNO	687.3	2.2	4400	0.002
Geo-FNO	10619 [6]	4.5	13800	0.007
φ-FEM-UNet	214.2	3.1	6200	0.002

TABLE 4.1 – **Cas test 1.** Temps de calcul (en secondes) pour chaque méthode.

Choix de la fonctionnelle $\mathcal{L}$ Nous avons choisi d'utiliser la norme H^1 (approchée) comme fonctionnelle à minimiser, plutôt que seulement la norme L^2. Ce choix est motivé par le gain en terme d'erreur, illustré à la Figure 4.11 où nous avons comparé la fonctionnelle $\mathcal{L}$ choisie et la loss L^2 (notée $\mathcal{L}_0$). Ces résultats illustrent que l'utilisation du gradient dans la fonctionnelle n'est pas obligatoire pour obtenir de bons résultats, mais améliore tout de même la précision de la méthode.

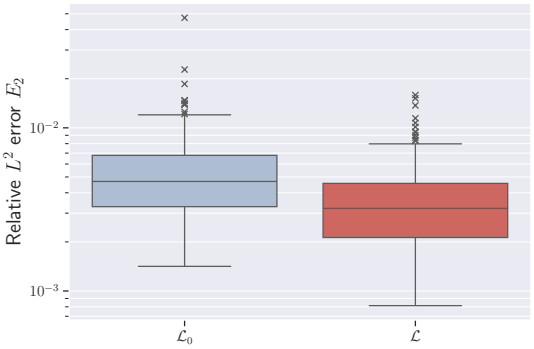

FIGURE 4.11 – Comparaison des deux fonctionnelles, appliquées au cas test 1.

4.4.2 Second cas test : problème de Poisson sur des géométries complexes aléatoires

Considérons une nouvelle fois le problème de Poisson (4.1) sur des géométries plus complexes, avec les fonctions f et g définies par (4.11) et (4.12), avec f restreinte à des valeurs positives uniquement. Pour ce cas test, on choisit de considérer des géométries construites à partir de fonctions level-set φ définies par des sommes de 3 fonctions

6. Il est important de préciser ici que l'implémentation de la génération de données n'est pas optimale. En effet, pour les données Geo-FNO il est nécessaire que toutes les données contiennent toujours le même nombre de points, ce qui rend la génération des maillages complexe dans notre cas. Ainsi, le temps de construction de tels maillages représente ici la majorité du temps de génération de données.

Gaussiennes, plus précisément :

$$\varphi(x,y) = -\psi(x,y) + 0.5 \max_{(x,y)\in[0,1]^2} \psi(x,y)\,, \tag{4.13}$$

avec

$$\psi(x,y) = \sum_{k=1}^{3} \exp\left(-\frac{(x-x_k)^2}{2\sigma_k} - \frac{(y-y_k)^2}{2\gamma_k}\right),$$

où les paramètres x_k, y_k, σ_k et γ_k ainsi que les paramètres des fonctions f et g sont générés à l'aide d'un Latin Hypercube [64]. Les hyper-paramètres d'entraînement sont les mêmes que pour le premier cas test, à l'exception de la taille de batch qui est fixée à 8, et on considère toujours des grilles cartésiennes de résolution 64×64.

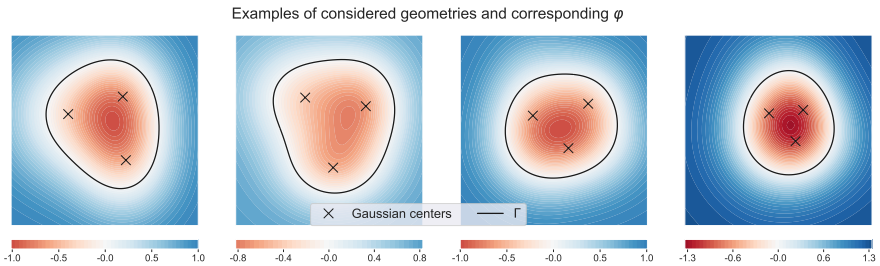

FIGURE 4.12 – **Cas test 2.** Exemples de fonctions level-set données par (4.13), avec les frontières Γ associées. Les centres des fonctions gaussiennes sont marqués par les croix noires.

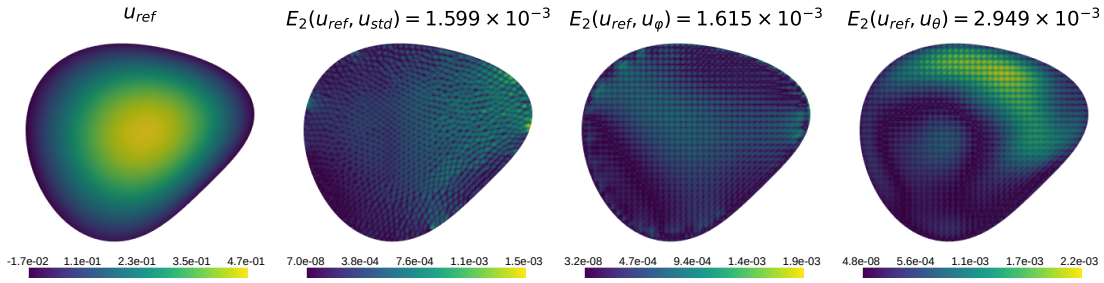

FIGURE 4.13 – **Cas test 2.** De gauche à droite : solution de référence, puis différences entre la solution de référence et la projection de la solution Standard-FEM (u_{std}), de la solution φ-FEM (u_φ), et de la prédiction (φ-FEM-FNO u_θ).
Le cas test présenté correspond au cas donnant l'erreur relative L^2 médiane.

Plusieurs exemples de fonctions φ sont représentés à la Figure 4.12. Comme pour le cas test précédent, l'opérateur est entraîné pendant 2000 epochs, mais cette fois seulement avec 500 données d'entraînement et toujours 300 de validation. De plus, l'erreur H^1 (4.5) est également à nouveau utilisée. On compare alors les performances de φ-FEM-FNO à φ-FEM, Standard-FEM et Standard-FEM-FNO, sur 300 nouvelles données test. Comme dans le cas test précédent, on utilisera une solution de référence FEM standard pour calculer l'erreur. Un exemple de solution de référence est représenté à la Figure 4.13.

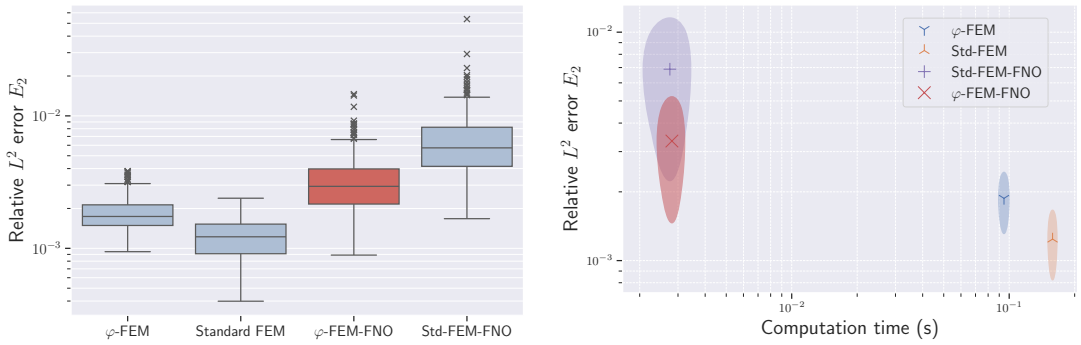

FIGURE 4.14 – **Cas test 2.** Gauche : erreurs des 4 méthodes sur 300 données de test. Droite : erreurs relatives L^2 en fonction du temps de calcul.

Les résultats présentés à la Figure 4.14 (gauche) illustrent une nouvelle fois que φ-FEM-FNO est capable d'atteindre une précision comparable à celle de φ-FEM et de Standard-FEM, tout en donnant également de meilleurs résultats que Standard-FEM-FNO. De plus, les résultats de φ-FEM-FNO et Standard-FEM-FNO sont obtenus significativement plus rapidement, comme cela est illustré à la Figure 4.14 (droite).

Enfin, la Figure 4.15 illustre la corrélation entre l'erreur du FNO et la distance de Hausdorff minimale entre une forme de test et les formes vues pendant l'entraînement.

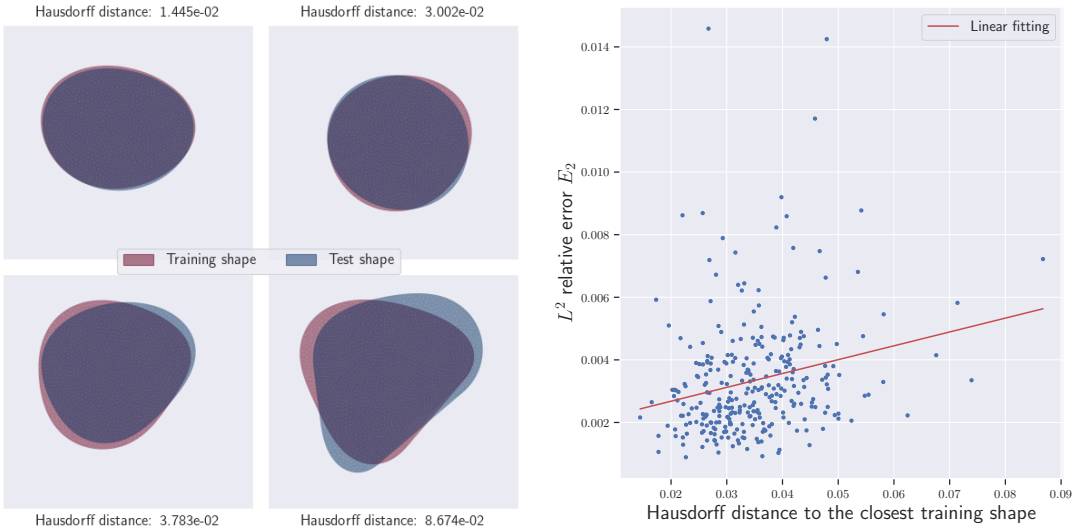

FIGURE 4.15 – **Cas test 2.** Gauche : Exemples de géométries de test. Les géométries d'entraînement représentées correspondent à chaque fois à la plus proche (au sens de la distance de Hausdorff) de la géométrie de test considérée.
Droite : erreurs relatives L^2 en fonction de la distance de Hausdorff minimale.

4.4.3 Déformation d'une plaque 2D trouée

Nous allons maintenant illustrer le potentiel de notre approche en considérant un cas test proche d'une application biomédicale [66] : l'équation d'élasticité non-linéaire (4.2).

Plus précisément, nous allons considérer une plaque rectangulaire avec 5 trous circulaires à l'intérieur. Ce domaine sera noté Ω dont un exemple est illustré à la Figure 4.16. Les différentes frontières de la plaque Ω sont données par :

- Γ_D^t et Γ_D^b sont le bord haut et le bord bas de la plaque, comme représenté à la Figure 4.16, où des conditions de Dirichlet sont imposées ;
- Γ_N est la frontière de Neumann, composée de :
  - Γ_N^l et Γ_N^r, respectivement le côté gauche et le côté droit de la plaque,
  - pour $i \in \{1, \dots, 5\}$, la frontière de chaque trou i notée Γ_N^i.

La plaque est fixée sur Γ_D^b, (i.e. $\boldsymbol{u} = 0$), et un déplacement constant $\boldsymbol{u}_D$ est appliqué sur Γ_D^t (i.e. des conditions de Dirichlet non homogènes sont imposées). Ces conditions ainsi que les conditions de Neumann sur Γ_N^l et Γ_N^r seront imposées de façon classique, tandis que les conditions de bord pour les différents trous seront imposées via φ-FEM.

Remarque 4.10. On partitionne la frontière Γ comme suit :

$$\Gamma = \overbrace{\Gamma_D^b \cup \Gamma_D^t \cup \Gamma_N^l \cup \Gamma_N^r}^{\text{imposition standard}} \cup \underbrace{\bigcup_{i=1}^{5} \Gamma_N^i}_{\text{imposition } \varphi\text{-FEM}} .$$

Le problème considéré peut être écrit sous la forme suivante (c.f. [48, eq. (8.28)]) : trouver le champ de déplacement $\boldsymbol{u} \in \mathbb{R}^2$ vérifiant

$$\begin{cases} -\operatorname{div} \boldsymbol{P}(F(\boldsymbol{u})) = 0\,, & \text{dans } \Omega\,, \\ \boldsymbol{u} = \boldsymbol{u}_D\,, & \text{sur } \Gamma_D^t\,, \\ \boldsymbol{u} = 0\,, & \text{sur } \Gamma_D^b\,, \\ \boldsymbol{P}(F(\boldsymbol{u})) \cdot \boldsymbol{n} = 0\,, & \text{sur } \Gamma_N\,. \end{cases}$$

On considère ici un matériau Néo-Hookéen compressible, comme à la Section 2.5. Le module de Young E est fixé à 0.97 Pa et le coefficient de Poisson ν à 0.3.

Le schéma φ-FEM

Comme nous l'avons dit précédemment, puisque l'on considère un domaine carré, une partie des conditions de bord peut être appliquée avec des méthodes standard. Il est en revanche nécessaire de construire un schéma φ-FEM le plus adapté à cette situation. Pour cela nous allons utiliser plusieurs level-set. Chacun des trous $\mathcal{C}_i$ de frontière $\Gamma_N^i = \{\varphi_i = 0\}$, $i = 1, \dots, 5$, est défini par

$$\mathcal{C}_i = \{\varphi_i < 0\}, \text{ avec } \varphi_i(x, y) = r_i^2 - (x - x_i)^2 - (y - y_i)^2\,,$$

où (x_i, y_i, r_i) sont les coordonnées du centre du trou et son rayon. Le domaine Ω peut alors être défini par

$$\Omega = \left\{ \underbrace{\prod_{i=1}^{5} \varphi_i}_{\varphi} < 0 \right\} \cap (0,1)^2.$$

Un exemple de configuration est représenté à la Figure 4.16.

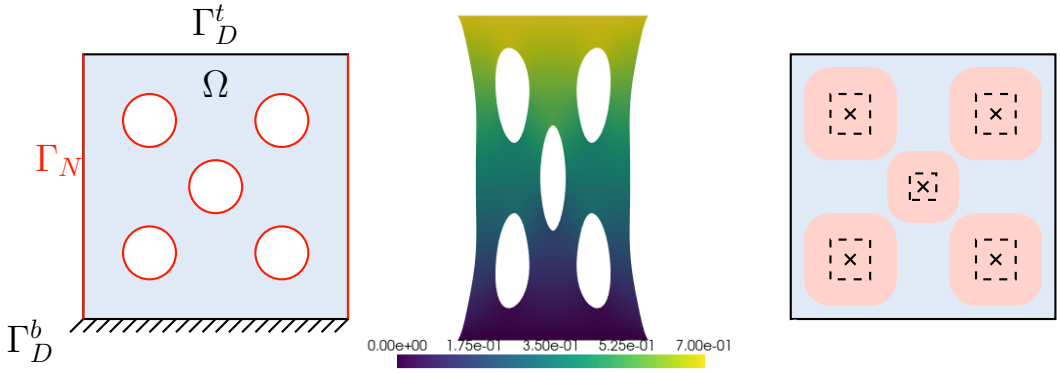

FIGURE 4.16 – **Cas test 3.** Gauche : représentation d'une configuration considérée. Centre : exemple de déformation. Droite : variations possibles de la géométrie. Les carrés en pointillés correspondent aux bornes des centres de chaque trou. Les sections rouges contiennent toutes les variations possibles des trous.

Pour construire le schéma φ-FEM, on introduit une nouvelle fois le maillage $\mathcal{T}_h$, construit à partir de l'interpolation φ_h de φ, qui couvre Ω et on note $\Omega_h := \cup_{T\in\mathcal{T}_h} T$.

On définit également le sous-maillage $\mathcal{T}_h^\Gamma$, contenant toutes les cellules de $\mathcal{T}_h$ en intersection avec l'un des trous :

$$\mathcal{T}_h^\Gamma := \{T \in \mathcal{T}_h : \exists i = 1, \dots, 5 \ \text{ t.q. } \ \varphi_i \geqslant 0 \ \text{ sur un nœud de } T\}$$

et on note $\Omega_h^\Gamma := \cup_{T\in\mathcal{T}_h^\Gamma} T$.

Les espaces éléments finis seront construits comme précédemment dans le cas de l'élasticité linéaire et non-linéaire. Plus précisément, pour $k \geqslant 2$, pour la solution $\boldsymbol{u}_h$, on considérera l'espace éléments finis $\boldsymbol{V}_h^{(k)}$ donné par (2.26) et l'espace homogène correspondant $V_h^{k,0}$. Pour imposer les conditions de Neumann sur les différents trous, on utilisera deux variables auxiliaires y et p. Pour cela, on introduit $\Omega_h^{\Gamma,i}$ le domaine recouvrant le maillage composé des cellules de $\mathcal{T}_h$ coupées par la frontière Γ_N^i :

$$\mathcal{T}_h^{\Gamma_N^i} = \{T \in \mathcal{T}_h : T \cap \Gamma_{i,h}^N \neq \emptyset\},$$

avec $\Gamma_{i,h}^N = \{\varphi_{i,h} = 0\}$, où $\varphi_{i,h}$ est l'interpolation $\mathbb{P}^k$ de φ_i sur $\mathcal{T}_h$.

Pour les variables auxiliaires, on considérera les espaces éléments finis $\boldsymbol{Z}_h(\Omega_h^\Gamma)$ (défini par (2.35)) et $\boldsymbol{Q}_h^{(k-1)}(\Omega_h^\Gamma)$ (défini par (2.27)). Pour chaque trou i, les conditions de Neumann seront imposées via les équations

$$\begin{aligned} \boldsymbol{y} + \boldsymbol{P}(F(\boldsymbol{u})) = 0, \quad \text{dans } \Omega_h^{\Gamma,i}\,, \\ \boldsymbol{y}\nabla\varphi_i + \boldsymbol{p}\varphi_i = 0, \quad \text{dans } \Omega_h^{\Gamma,i}\,. \end{aligned}$$

On obtient alors le schéma φ-FEM suivant : trouver $\boldsymbol{u}_h \in V_h^k$, $\boldsymbol{p}_h \in \boldsymbol{Q}_h^{(k-1)}(\Omega_h^\Gamma)$ et $\boldsymbol{y}_h \in \boldsymbol{Z}_h(\Omega_h^\Gamma)$ tels que

$$\begin{aligned} \int_{\Omega_h} \boldsymbol{P}(F(\boldsymbol{u}_h)) : \nabla \boldsymbol{v}_h + \sum_{i=1}^{5} \Big(\int_{\partial\Omega_h^{\Gamma,i}} \boldsymbol{y}_h \boldsymbol{n}\cdot\boldsymbol{v}_h + \gamma_u \int_{\Omega_h^{\Gamma,i}} (\boldsymbol{y}_h + \boldsymbol{P}(F(\boldsymbol{u}_h))) : (\boldsymbol{z}_h + D_{\boldsymbol{u}}(\boldsymbol{P}\circ F)(\boldsymbol{u}_h)\boldsymbol{v}_h) \\ + \frac{\gamma_p}{h^2} \int_{\Omega_h^{\Gamma,i}} (\boldsymbol{y}_h \nabla\varphi_{i,h} + \frac{1}{h}\boldsymbol{p}_h\varphi_{i,h}) \cdot (\boldsymbol{z}_h\nabla\varphi_{i,h} + \frac{1}{h}\boldsymbol{q}_h\varphi_{i,h}) \\ + \gamma_{div} \int_{\Omega_h^{\Gamma,i}} \operatorname{div}\boldsymbol{y}_h \cdot \operatorname{div}\boldsymbol{z}_h \Big) + G_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = 0\,, \\ \forall \boldsymbol{v}_h \in \boldsymbol{V}_h^{k,0},\ \boldsymbol{q}_h \in \boldsymbol{Q}_h^{(k-1)}(\Omega_h^\Gamma),\ \boldsymbol{z}_h \in \boldsymbol{Z}_h(\Omega_h^\Gamma)\,, \end{aligned}$$

où

$$G_h(\boldsymbol{u}, \boldsymbol{v}) := \sigma_N h \int_{\Gamma_h} [\boldsymbol{P}(F(\boldsymbol{u}))\boldsymbol{n}] \cdot [D_{\boldsymbol{u}}(\boldsymbol{P}\circ F)(\boldsymbol{u})\boldsymbol{v}\boldsymbol{n}]\ ,$$

avec $\Gamma_h := \partial\Omega_h^\Gamma \setminus \partial\Omega_h$, $D_{\boldsymbol{u}}(\boldsymbol{P}\circ F)(\boldsymbol{u})\boldsymbol{v}$ la dérivée de $\boldsymbol{P}$ évaluée en $\boldsymbol{u}$, dans la direction $\boldsymbol{v}$ et γ_p, γ_u, γ_{div}, σ_N des constantes positives.

Opérateur φ-FEM-FNO

Plusieurs aspects de ce cas test le rendent particulièrement différent des précédents. Dans un premier temps, on considère maintenant des conditions de Neumann et de Dirichlet. De plus, le bord du domaine étant le bord de la grille cartésienne, il sera nécessaire d'appliquer un padding au bord (c.f. Remarque 4.5) pour éviter le phénomène de Gibbs. De plus, contrairement aux situations précédentes, la solution obtenue par le schéma φ-FEM est directement la solution du problème, qui ici est vectorielle. Ainsi, le FNO prédira directement la solution, comme nous l'avons fait pour l'approche φ-FEM-FNO-2 dans le premier cas test. Enfin, puisque les données variables de ce cas test sont la géométrie et le déplacement $\boldsymbol{u}_D$ appliqué sur Γ_D^t, l'opérateur *ground-truth* à approcher est défini par

$$\begin{array}{rccc} \mathcal{G}^\dagger : & \mathbb{R}^{n_x\times n_y\times 2} & \to & \mathbb{R}^{n_x\times n_y\times 2} \\ & (\varphi_h, g_{h,y}) & \mapsto & \boldsymbol{u}_h = (u_{h,x}, u_{h,y})\,, \end{array} \tag{4.14}$$

où $u_{h,x}$ et $u_{h,y}$ sont les deux composantes du champ de déplacement $\boldsymbol{u}_h$, et $g_{h,y}$ est la composante verticale du déplacement $\boldsymbol{u}_D$ imposé au bord, constante sur l'ensemble du domaine (i.e. $g_{h,y} = g$ sur chaque pixel).

Dans ces situations de problèmes non-linéaires, les réseaux de neurones ont un grand avantage par rapport aux méthodes éléments finis classiques. En effet, comme nous avons pu le voir par exemple dans la Section 2.5, pour les méthodes classiques il est nécessaire d'utiliser des solveurs itératifs et souvent plusieurs incréments pour appliquer les forces. Ce nombre d'incréments peut fortement varier en fonctions des cas, ce qui le rend difficile

à déterminer de manière optimale. L'approche φ-FEM-FNO quant à elle permet de déterminer directement la solution sans passer par des itérations ou l'ajout de forces par incréments.

Génération de données Pour générer les données (entraînement, validation et test), on se place dans les configurations représentées à la Figure 4.16 (droite). Les trous sont placés suffisamment loin des bords de la plaque et les bornes des paramètres caractérisant les trous sont choisis de sorte à éviter des interpénétrations. La génération de paramètres est une nouvelle fois réalisée à l'aide d'un Latin Hypercube de dimension 16 : 15 dimensions pour les paramètres des différents trous et une dimension pour la condition de bord (qui appartient à l'intervalle $[0.3, 0.9]$). Les paramètres du schéma φ-FEM sont fixés à $\gamma_u = 0.001$, $\gamma_p = \gamma_{div} = \sigma_N = 0.01$. De plus, on réalise des simulations avec des éléments finis $\mathbb{P}^2$ sur des grilles de résolutions 64×64 et on ne conserve que les valeurs aux nœuds pour construire la base de données.

Modification de la fonctionnelle Pour l'entraînement de l'opérateur, on choisit de minimiser une approximation de la semi-norme H^1, définie par

$$\mathcal{L}\left(U_{\text{true}}; U_\theta\right) = \frac{1}{N_{\text{data}}} \sum_{n=0}^{N_{\text{data}}} \left(\mathcal{E}_1(u^n_{\text{true},x}; u^n_{\theta,x}) + \mathcal{E}_1(u^n_{\text{true},y}; u^n_{\theta,y}) \right) ,$$

où

$$\mathcal{E}_1(u^n_{\text{true},\cdot}; u^n_{\theta,\cdot}) = \|\nabla^h_x u^n_{\text{true},\cdot} - \nabla^h_x u^n_{\theta,\cdot}\|^2_{0,\mathcal{S}_1^n} + \|\nabla^h_y u^n_{\text{true},\cdot} - \nabla^h_y u^n_{\theta,\cdot}\|^2_{0,\mathcal{S}_1^n} ,$$

où $\boldsymbol{u}_{\text{true}} = (u_{\text{true},x}, u_{\text{true},y})$ est la solution de l'opérateur $\mathcal{G}^\dagger$ (4.14) et $\boldsymbol{u}_\theta = (u_{\theta,x}, u_{\theta,y})$ est la solution obtenue par l'approximation $\mathcal{G}_\theta$.

Remarque 4.11. Utiliser la semi-norme H^1 plutôt que la norme H^1 permet dans cette situation d'améliorer les performances de l'opérateur, en particulier aux bords. Cependant, une fois l'opérateur entraîné, cela rend l'étape d'inférence plus lourde numériquement que précédemment. En effet, il faut replacer la solution prédite dans le domaine de référence, ce qui est fait par soustraction de la valeur moyenne de la prédiction sur le bas de la grille, là où la solution doit être nulle. Cette méthode permet ainsi de simplifier l'optimisation, puisque la fonctionnelle est moins lourde qu'en utilisant la norme H^1 tout en améliorant également les performances en termes d'erreur.

Résultats numériques Nous allons maintenant comparer notre approche à une méthode éléments finis classique, à φ-FEM ainsi qu'à Standard-FEM-FNO et Geo-FNO. Pour les méthodes éléments finis, on utilisera des éléments $\mathbb{P}^2$, avec des maillages dont les tailles de cellules correspondent à des grilles cartésiennes de résolution 31×31. Les trois méthodes basées sur l'utilisation d'un FNO sont entraînées avec 200 données d'entraînement, divisées en batches de taille 8 et 300 données de validation, pendant 2000 epochs.

Pour évaluer les performances des différentes méthodes, on considérera l'erreur relative L^2 par rapport à une solution de référence $\boldsymbol{u}_{\text{ref}}$, que l'on notera $\bar{L_2}(\boldsymbol{u}_{\text{ref}}, \boldsymbol{u}_h)$.

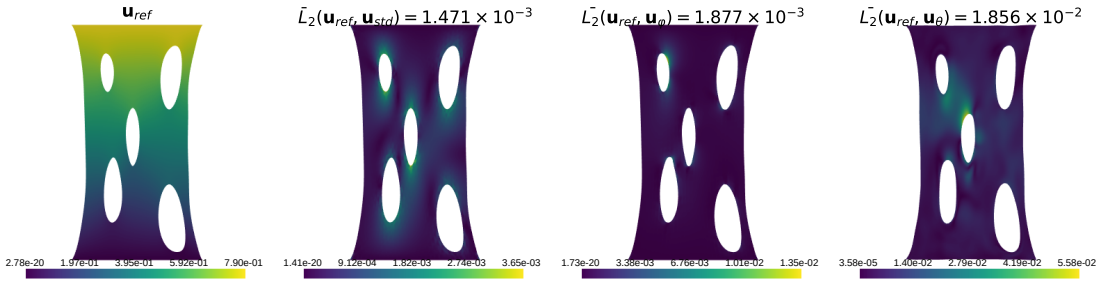

FIGURE 4.17 – **Cas test 3.** Exemple de déplacement obtenu, pour le cas correspondant à l'erreur médiane de φ-FEM-FNO.

Un exemple de déplacement obtenu est représenté à la Figure 4.17, où la géométrie de référence est déformée par la solution correspondante (solution de référence, solution Standard-FEM, solution φ-FEM, solution φ-FEM-FNO), interpolée sur le maillage de référence, avec en couleur l'erreur en chaque point par rapport à la solution de référence.

On compare les différentes méthodes sur le jeu de données de test (de taille 300). Les erreurs relatives L^2 représentées à la Figure 4.18 (gauche) indiquent que l'approche φ-FEM-FNO est la plus précise parmi les approches machine learning testées. Cependant, contrairement aux cas test précédents les résultats sont moins précis que les méthodes éléments finis à nombre de degrés de liberté équivalent. Finalement, on s'intéresse aux temps de calcul des différentes méthodes. Les résultats de la Figure 4.18 (droite) illustrent parfaitement l'intérêt de φ-FEM-FNO : en moyenne, une erreur relative de 2% (environ 10 fois plus que pour les méthodes éléments finis) est obtenue et cela 1000 fois plus rapidement que φ-FEM et Standard-FEM.

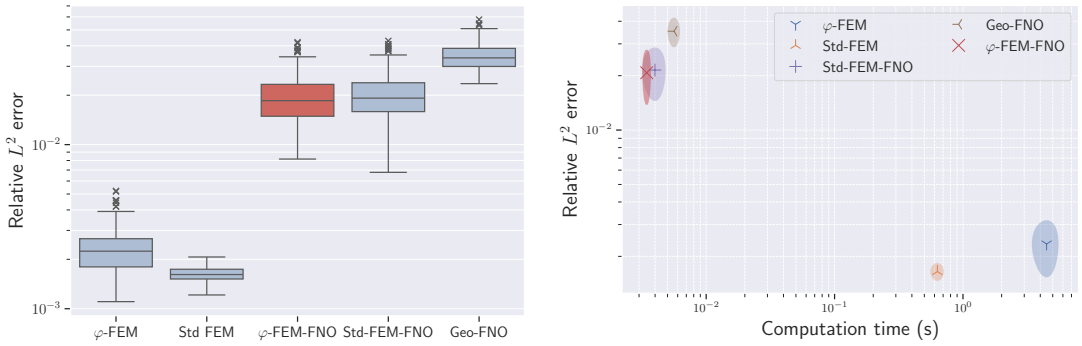

FIGURE 4.18 – **Cas test 3.** Gauche : erreurs relatives L^2. Droite : erreurs relatives L^2 en fonction du temps de calcul.

4.5 Conclusion

Nous avons présenté une nouvelle approche hybride entre méthode éléments finis et méthode de Machine Learning, appelée φ-FEM-FNO, permettant de traiter le cas de

géométries complexes. L'étude numérique de cette méthode sur trois cas test a illustré l'intérêt de notre approche. En effet, après entraînement de la méthode, φ-FEM-FNO permet d'obtenir systématiquement des résultats plus rapides que les méthodes φ-FEM ou Standard-FEM. L'approche s'est également montrée plus précise que plusieurs autres approches combinant φ-FEM avec un réseau UNet, Standard-FEM avec un FNO ou encore Geo-FNO. De plus, la méthode a permis d'obtenir ces résultats en utilisant peu de données d'entraînement, même dans le cas de problèmes complexes avec de grandes variations de géométries.

5 Quelques résultats en lien avec φ-FEM

Résumé

Dans ce dernier chapitre, nous présentons en détail deux outils utilisés dans les chapitres précédents, permettant d'utiliser en pratique des fonctions level-set. Dans un premier temps, nous décrirons une méthode de construction de maillages conformes à partir de fonctions level-set. Ensuite, nous proposerons deux techniques permettant de reconstruire des fonctions level-set dans des cas plus généraux à partir d'images binaires.

Dans une seconde partie, nous présenterons une méthode permettant de diminuer le temps de calcul de la méthode φ-FEM en combinant cette approche à une méthode multigrid. Nous présenterons alors cette approche, nommée φ-FEM-Multigrid et illustrerons numériquement son intérêt sur plusieurs cas test.

Enfin, nous proposerons une dernière méthode, basée sur l'approche précédente. Cette approche combinera alors les réseaux de neurones (FNO) avec l'approche φ-FEM-Multigrid.

Chapitre 5 – Quelques résultats en lien avec φ-FEM

Ce dernier chapitre sera consacré à la présentation de deux outils utilisés durant cette thèse, ainsi qu'à la combinaison de la méthode φ-FEM avec une approche de type *multigrid*. Dans une première section, nous présenterons la méthode qui a été utilisée à plusieurs reprises dans ce manuscrit afin de générer des maillages à partir de fonctions

level-set. Dans cette même section, nous proposerons ensuite une nouvelle méthode permettant de reconstruire des approximations de fonctions level-set lisses à partir d'images binaires. Nous illustrerons alors les intérêts et défauts de cette méthode et nous justifierons son intérêt dans notre situation. Ensuite, la deuxième section de ce chapitre sera consacrée à la présentation d'une méthode que nous avons appelée φ-FEM-M, pour φ-FEM-Multigrid. Nous présenterons alors l'algorithme ainsi que différents résultats numériques. Enfin, dans une troisième section, nous présenterons une méthode hybride combinant les avantages de la méthode φ-FEM-M et ceux de la méthode φ-FEM-FNO.

5.1 L'utilisation de fonctions *level-set* en pratique

Nous allons maintenant présenter deux méthodes qui ont eu un rôle essentiel pour les simulations numériques présentées tout au long de ce manuscrit. La première méthode a été utilisée à de nombreuses reprises pour générer des maillages de géométries complexes. La seconde méthode proposera une nouvelle technique permettant de construire des fonctions level-set utilisables notamment pour l'approche φ-FEM direct.

5.1.1 Construction d'un maillage conforme à partir d'une level-set

Dans un premier temps, nous proposons une approche qui a notamment été motivée par la nécessité de cas test numériques sur des géométries complexes et la limitation des mailleurs usuels à des formes classiques (cercles, carrés, ellipses, ...). Ainsi, puisqu'il était important de comparer la méthode φ-FEM à une méthode éléments finis classique, il était indispensable de considérer des situations où l'on disposait d'un maillage conforme pour une level-set donnée, notamment pour calculer des solutions de référence.

La librairie MMG [68], combinée à la librairie PyMedit [1], offre la possibilité de construire des maillages conformes à partir d'une level-set donnée. Un des principaux atouts de cette librairie est la qualité des maillages construits. Comme on peut le voir à la Figure 5.1 pour plusieurs résolutions, les maillages reconstruits sont très réguliers, ce qui est très intéressant numériquement.

Pour évaluer la précision de reconstruction de la frontière, on utilise l'expression analytique d'une level-set φ donnée par

$$\varphi(x, y) = r - R_0(1 + A\cos(n\theta))\,, \tag{5.1}$$

avec

$$\begin{cases} R_0 &= 0.3\,, \\ A &= 0.3\,, \\ n &= 5\,, \\ r &= \sqrt{(x-0.5)^2 + (y-0.5)^2}\,, \\ \theta &= \arctan 2(y-0.5, x-0.5)\,. \end{cases}$$

1. `https://gitlab.com/florian.feppon/pymedit`

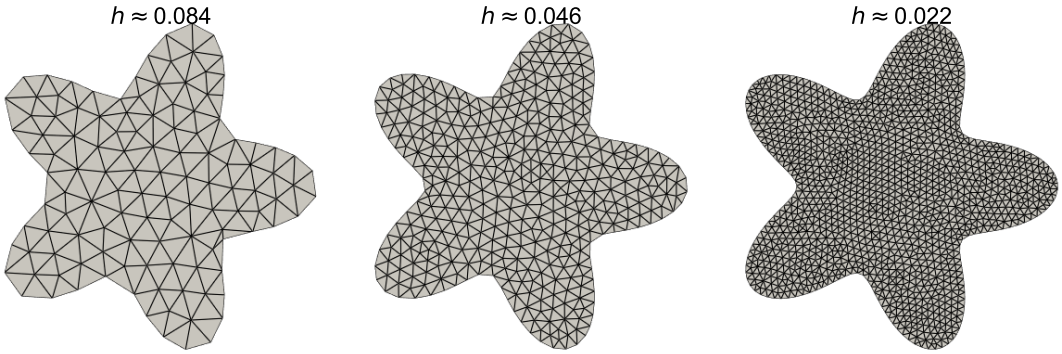

FIGURE 5.1 – Exemples de maillages reconstruits à partir de l'expression φ définie par (5.1)

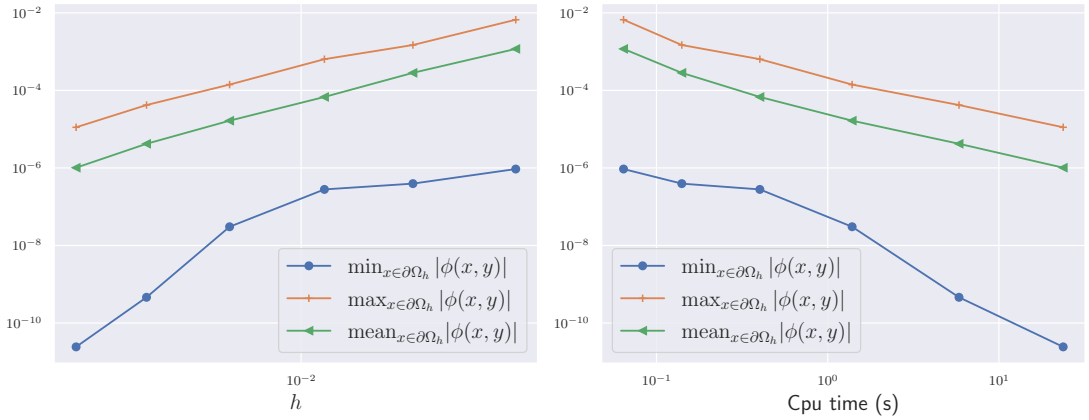

FIGURE 5.2 – Erreurs de reconstruction au bord en fonction de la taille de cellule (gauche) et du temps de calcul (droite).

On mesure l'erreur $|\varphi(x,y)|$ en chaque nœud de bord du maillage reconstruit et on s'intéresse à la moyenne, la valeur maximale et la valeur minimale. On représente les résultats obtenus en fonction de la taille de cellule maximale du maillage reconstruit, ainsi qu'en fonction du temps de construction du maillage à la Figure 5.2. Comme on peut le voir, pour obtenir une précision satisfaisante au bord du maillage, il est nécessaire de générer des maillages extrêmement fins. En particulier, il est très difficile d'atteindre une précision de l'ordre de la précision machine au bord.
L'idée étant d'utiliser cette méthode pour construire des solutions de référence, il est souhaitable de reconstruire le plus fidèlement possible le bord de la géométrie exacte.

Pour améliorer la précision au bord des maillages reconstruits, une méthode de recalage des nœuds de bord est utilisée. Pour cela, on construira dans un premier temps un maillage initial avec l'approche précédente, permettant d'avoir une initialisation relativement précise. En sélectionnant ensuite les nœuds de bord du maillage reconstruit,

il suffit alors d'appliquer l'algorithme suivant à chaque point $\boldsymbol{x}_i = (x_i, y_i)$ du bord,

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} - \varphi(\boldsymbol{x}_i^{(k)}) \frac{\nabla\varphi(\boldsymbol{x}_i^{(k)})}{\|\nabla \boldsymbol{x}_i^{(k)}\|}, \tag{5.2}$$

pour tout $k < N$ où N est un nombre d'itérations maximal, ou bien tant que $\varphi(\boldsymbol{x}_i^{(k)}) > \text{tol}$ où tol est une tolérance fixée, de l'ordre de la précision machine dans notre cas. Un exemple d'application de la méthode est représenté à la Figure 5.3, où la précision machine est obtenue après moins de 4 itérations pour chaque point.

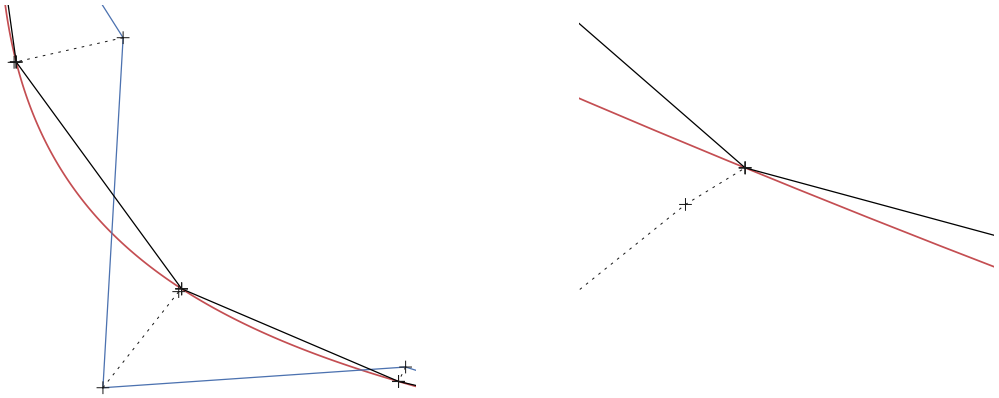

FIGURE 5.3 – Illustration de l'application de la méthode. Les croix noires sur les segments bleus correspondent aux nœuds de bord maillage initial. La frontière exacte est représentée en rouge et les différentes itérations de l'algorithme (5.2) pour chaque nœud considéré sont marquées avec des croix noires. La direction suivie à chaque itération est tracée en pointillés. Enfin, les segments noirs correspondent aux faces du bord optimal reconstruit. La figure de droite est un zoom des itérations correspondant au point du milieu sur la figure de gauche.

On représente à la Figure 5.4 les résultats obtenus pour la situation précédente. On voit alors que le coût de calcul supplémentaire est relativement faible, pour un gain de précision au bord très important puisque l'on obtient ainsi des résultats de l'ordre de la précision machine (10^{-14}).

Cette approche a également été utilisée pour des cas 3D, avec le même gain de précision comme sur l'exemple proposé à la Figure 5.5.

5.1.2 Approximation d'une level-set à partir d'une image binaire

Comme nous l'avons vu tout au long de ce manuscrit, la méthode φ-FEM repose sur l'utilisation d'une fonction level-set. Dans une majorité des cas tests présentés, nous nous sommes restreints à des géométries simples à décrire (cercles, ellipses, carrés, sphères, ...). Certains de nos cas tests impliquaient des géométries plus complexes à décrire, par exemple le second cas test de la Section 4.4.2. Cependant, dans l'ensemble de ces cas test, nous avons toujours considéré des fonctions level-set analytiques.

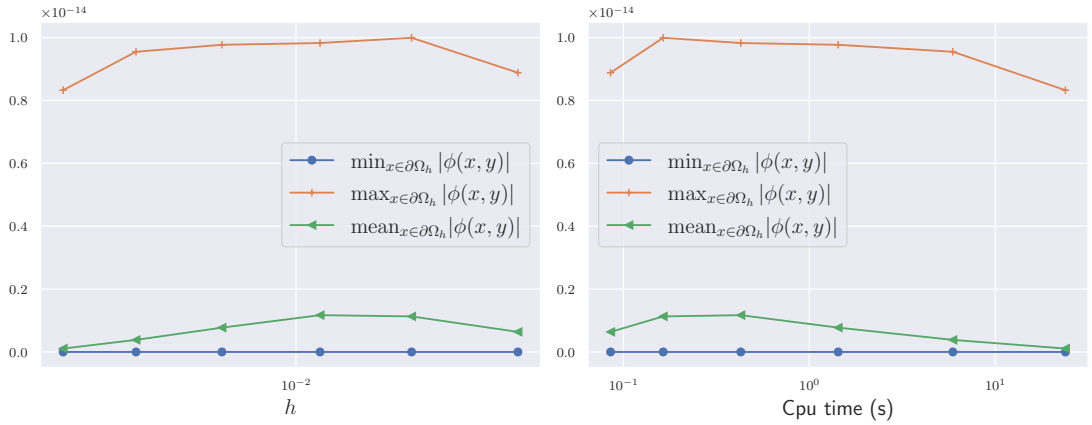

FIGURE 5.4 – Erreurs de reconstruction après recalage au bord en fonction de la taille de cellule (gauche) et du temps de calcul (droite).

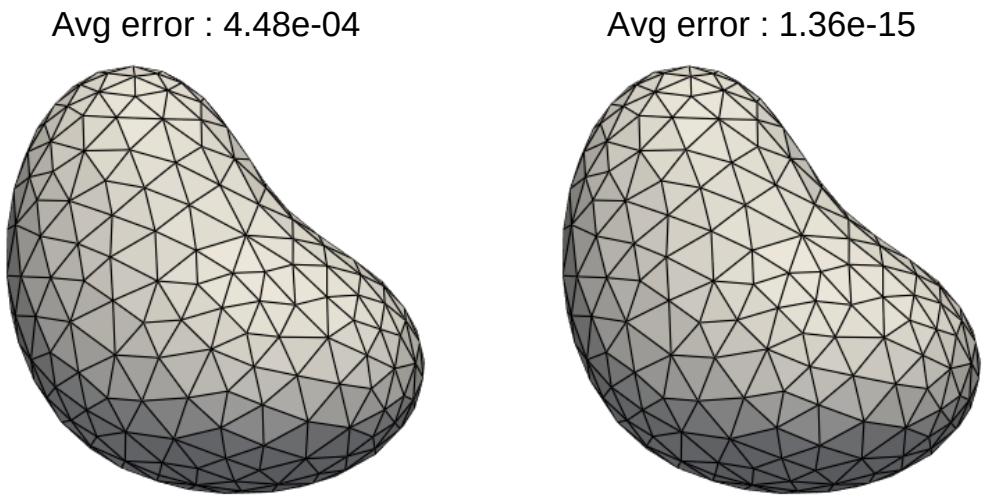

FIGURE 5.5 – Gauche : maillage 3D reconstruit à partir d'une level-set donnée. Droite : maillage adapté au bord.

En pratique, la construction de telles level-set peut être complexe en fonction des données d'entrée dont on dispose (maillage, nuage de points, images, etc). Ainsi pour compléter les résultats précédemment proposés, nous avons choisi de considérer une situation plus complexe, en considérant comme données d'entrée une image binaire en 2D. Cette situation génère plusieurs difficultés dont la plus importante est la localisation de la frontière du domaine représenté par cette image. Pour cela, on considérera par la suite que la frontière réelle se trouve sur des pixels associés à l'intérieur du domaine. En effet, dans le cas d'images binaires, il sera impossible de construire des coordonnées exactes de points de frontière uniquement à partir d'une image binaire. Nous allons donc proposer deux approches de reconstruction de level-set que nous combinerons aux schémas φ-FEM (direct et dual) sur un exemple de résolution du problème de Poisson (1.1). Nous comparerons alors ces approches à une méthode éléments finis classique en utilisant 2 méthodes de construction de maillage que nous détaillerons.

SDF-generator Une idée naturelle pour construire une level-set à partir d'un maillage ou d'une image serait de considérer la distance signée. Pour cela, il existe de nombreuses méthodes : des méthodes déterministes (par exemple la Fast-Marching-Method [81]) ou bien des méthodes basées sur des réseaux de neurones, par exemple [74]. On choisit ici d'utiliser une méthode déterministe, en utilisant la librairie Scipy [88].

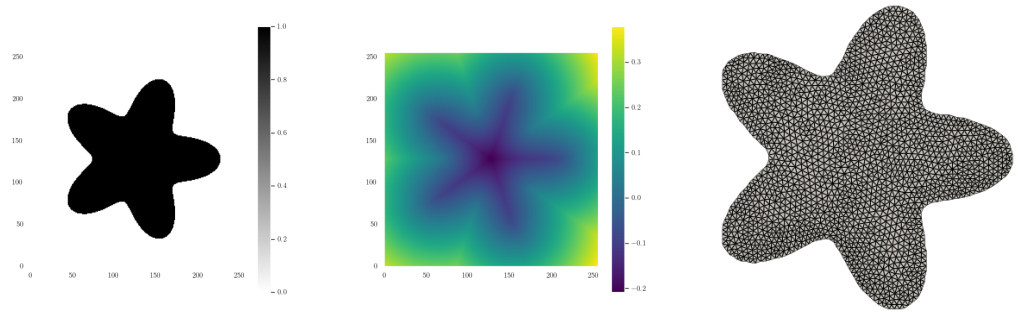

FIGURE 5.6 – Gauche : image binaire. Centre : distance signée reconstruite. Droite : maillage conforme reconstruit.

Pour ce cas test, on construit une image binaire (contenant la frontière du domaine réel) à partir de l'expression (5.1). L'image binaire générée est représentée à la Figure 5.6 (gauche). On construit alors la distance signée à la frontière de l'ensemble de pixels noirs, représentée à la Figure 5.6 (centre). Enfin, à partir de la méthode présentée à la Section 5.1.1, on utilise cette distance signée pour reconstruire un maillage conforme. Comme on peut le voir, la qualité au bord du maillage reconstruit est inférieure à la qualité des maillages représentés à la Figure 5.1, ce qui est évidemment dû à l'approximation de la frontière du domaine uniquement par des segments (que l'on pourrait voir comme les côtés de chaque pixel). Cela se retrouve également dans les résultats présentés Table 5.1. Ainsi, cette méthode génère une perte de précision au bord, mais offre tout de même des résultats relativement satisfaisants.

Cependant, comme nous avons pu le voir notamment avec la Figure 2.1, la version directe du schéma φ-FEM peut être très sensible à la level-set utilisée. En particulier, dans ce cas test, les résultats lors de l'utilisation de la distance signée étaient nettement dégradés par rapport à l'utilisation d'une expression plus lisse. Ainsi, plusieurs techniques de régularisation ont été testées, et nous avons finalement choisi d'appliquer la fonction tanh à la distance signée calculée avant d'effectuer une interpolation par splines cubiques.

Produit de gaussiennes Dans un second temps, nous avons choisi de proposer une nouvelle approche, afin de reconstruire des approximations de fonctions level-set caractérisant des frontières plus lisses qu'avec la distance signée.

Remarque 5.1. Il est important de préciser différents points. Dans un premier temps, il s'agit une nouvelle fois d'approximations, de par la nature mal posée du problème à résoudre. De plus, de par le choix de la forme de la level-set φ reconstruite, afin d'obtenir des résultats satisfaisants il sera nécessaire que les géométries considérées soient relativement lisses. Enfin, nous ne présenterons la méthode que dans le cas 2D, mais cette dernière pourra être étendue à des situations 3D. Cependant, le coût de la méthode pourra alors être relativement augmenté.

Pour cette méthode, l'idée est de construire une level-set sous la forme d'un produit de fonctions Gaussiennes, définie par

$$\varphi(x,y) = (-1)^n \prod_j^n \left(-1 + \exp\left(-\frac{x_j^2}{2l_{x,j}^2} - \frac{y_j^2}{2l_{y,j}^2} \right) \right), \tag{5.3}$$

où

$$x_j = \cos(\theta_j)(x - x_{0,j}) - \sin(\theta_j)(y - y_{0,j}) \text{ et } y_j = \sin(\theta_j)(x - x_{0,j}) + \cos(\theta_j)(y - y_{0,j}) .$$

Pour cela, on cherche à optimiser le choix des paramètres θ, x_0, y_0, l_x et l_y.
La méthode est séparée en plusieurs étapes :

1. À partir de l'image binaire, on construit deux polygones : le premier contiendra le domaine, en particulier sa frontière, et le second sera construit en retirant une couche de pixels au domaine (i.e. sera le plus grand domaine construit à partir de l'image, ne contenant pas la frontière) ;
2. On construit le squelette de l'image avec la librairie Python Scikit-Image [86] (c.f. Figure 5.7 gauche) ;
3. À partir du squelette, on détermine des points initiaux ainsi que le nombre de gaussiennes à utiliser. Pour construire les points initiaux, on utilisera les points de jonctions de plusieurs branches ainsi que les points de fin des branches. Un exemple de points initiaux est représenté à la Figure 5.7 (centre) ;
4. On minimise finalement une fonctionnelle afin de trouver les paramètres optimaux et d'obtenir φ comme représentée à la Figure 5.7 (droite). Le choix de la fonctionnelle a évidemment une grande importance dans la qualité des résultats. En effet, pour obtenir une solution φ satisfaisante, cette dernière devra capter les oscillations de la frontière, sans que ces oscillations ne fassent exploser les dérivées et dérivées

secondes de la solution. Pour cela, on construit une fonctionnelle qui sera évaluée en φ pour tout $\varphi_{(x_0,y_0,l_x,l_y,\theta)}(x,y)$, composée de plusieurs termes :

$$F(\varphi) = \alpha f_1(\varphi) + \beta f_2(\varphi) + \gamma f_3(\varphi) + \delta f_4(\varphi)\,,$$

où :

$$\begin{aligned}
f_1(\varphi) &= \frac{1}{n_x n_y} \sum_{(x,y)\in B} \left(\frac{\partial^2}{\partial x^2}\varphi(x,y)^2 + 2\frac{\partial^2}{\partial x \partial y}\varphi(x,y)^2 + \frac{\partial^2}{\partial y^2}\varphi(x,y)^2 \right)\,, \\
f_2(\varphi) &= \sum_{(x,y)\in B_i} \varphi(x,y)^2\,, \\
f_3(\varphi) &= \sum_{(x,y)\in B_e} \varphi(x,y)^2\,, \\
f_4(\varphi) &= \frac{1}{n_x n_y} \sum_{(x,y)\in B} \left(1 - \left(\frac{\partial}{\partial x}\varphi(x,y)^2 + \frac{\partial}{\partial y}\varphi(x,y)^2 \right) \right)^2\,,
\end{aligned}$$

où B_e et B_i sont les deux polygones construits à l'étape 1, et B est l'ensemble des coordonnées de la discrétisation de $[0,1]\times[0,1]$ correspondant à l'image considérée. Numériquement, toutes les dérivées seront approchées par des différences finies centrées du second ordre.

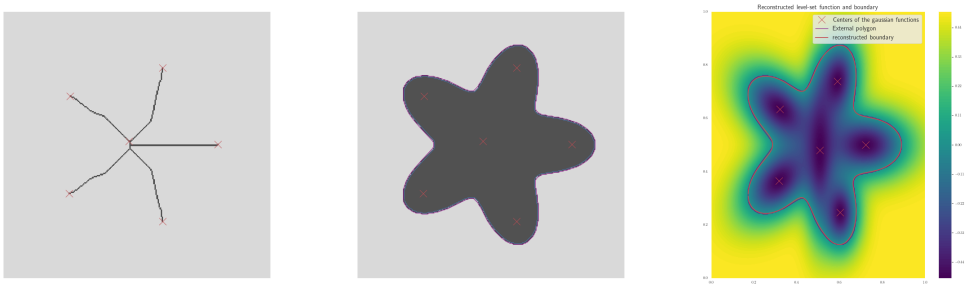

FIGURE 5.7 – Construction de level-set à partir d'images binaires.

Remarque 5.2. Bien que cette méthode ne fournisse qu'une approximation de la frontière à partir d'une image, elle permet néanmoins de générer des cas tests sur des géométries particulièrement complexes, comme le montre la Figure 5.8. Dans l'exemple présenté, on constate que l'approximation de la frontière n'est pas idéale avec les paramètres choisis. Toutefois, une géométrie satisfaisante est obtenue avec seulement quatre gaussiennes. Les paramètres estimés, bien qu'ils ne coïncident pas précisément avec ceux de la géométrie réelle, permettent néanmoins de reconstruire une forme suffisamment complexe pour être exploitée dans le cas test 3 de la Section 2.1.

Remarque 5.3. Une adaptation possible de cette méthode est l'utilisation d'une combinaison linéaire de gaussiennes plutôt que le produit (5.3). Ainsi, cela permettra par

exemple de mieux capter certaines oscillations de la frontière. Cependant, l'optimisation sera plus complexe notamment en raison des paramètres supplémentaires à optimiser et de la nécessité d'ajouter des contraintes d'optimisation.

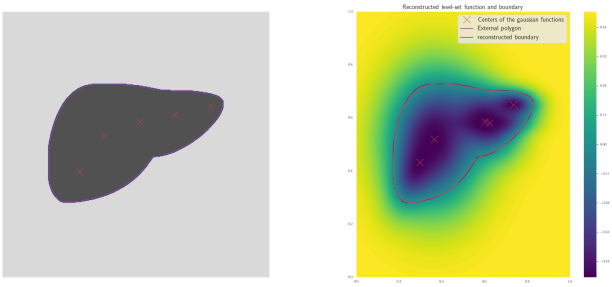

FIGURE 5.8 – Cas d'une géométrie plus complexe. Gauche : image initiale et initialisation des centres. Droite : solution φ obtenue.

Cas test Pour déterminer l'erreur lors de la reconstruction de maillages conformes, on utilise les deux approches précédentes combinées à la méthode présentée en Section 5.1.1. On compare également ces approches à une méthode permettant de reconstruire un maillage directement à partir d'une image, en utilisant la librairie Python *nanomesh*[2] [84].

Pour évaluer les reconstructions de maillage, on utilise une nouvelle fois l'expression (5.1) et on calcule l'erreur $|\varphi(x, y)|$ au bord. On obtient alors les résultats présentés dans le Tableau 5.1, pour des tailles de maillages comparables. On peut notamment remarquer que la méthode de reconstruction à partir de fonctions gaussiennes offre approximativement les mêmes résultats que l'approche à partir de la distance signée et que ces deux méthodes semblent légèrement plus précise que la version Nanomesh. On compare également l'approche de référence, pour laquelle le maillage est construit à partir de l'expression exacte de la level-set.

	Référence	Nanomesh	Distance signée	Gaussiennes
Erreur Mini.	0.0	2.3×10^{-5}	4.5×10^{-6}	2.0×10^{-6}
Erreur Moyenne	8.0×10^{-16}	2.9×10^{-3}	1.2×10^{-3}	1.2×10^{-3}
Erreur Maxi.	9.9×10^{-15}	7.7×10^{-3}	4.3×10^{-3}	4.1×10^{-3}

TABLE 5.1 – Erreurs de reconstruction à la frontière.

Dans un second temps, on s'intéresse à la résolution de l'équation de Poisson (2.12), avec uniquement des conditions de bord Dirichlet ($\Gamma = \Gamma_D$). Le second membre du problème sera donné par $f(x, y) = 10\cos(x - 0.5)\sin(\pi/3(y - 0.5))$ et les conditions de bord par $u_D(x, y) = x\cos(y)$.

2. `https://github.com/hpgem/nanomesh`

On considère une nouvelle fois l'expression (5.1) pour générer un maillage de référence sur lequel on calcule une solution de référence via une méthode standard.
On compare alors 9 approches sur la Figure 5.9 :

1. une méthode éléments finis classique avec (en rouge) :
   - un maillage généré à l'expression exacte de φ, en utilisant l'approche de la Section 5.1.1 (***traits pleins***) ;
   - un maillage généré à partir de la distance signée à une image binaire (avec l'approche de la Section 5.1.1) (***traits discontinus***) ;
   - un maillage généré à l'aide de Nanomesh (***pointillés***) ;
2. les schémas φ-FEM direct (en bleu) et dual (en vert), en utilisant :
   - l'expression exacte de φ (***traits pleins***) ;
   - la distance signée (régularisée avec la fonction tanh) (***traits discontinus***) ;
   - l'expression (5.3) (***pointillés***) ;

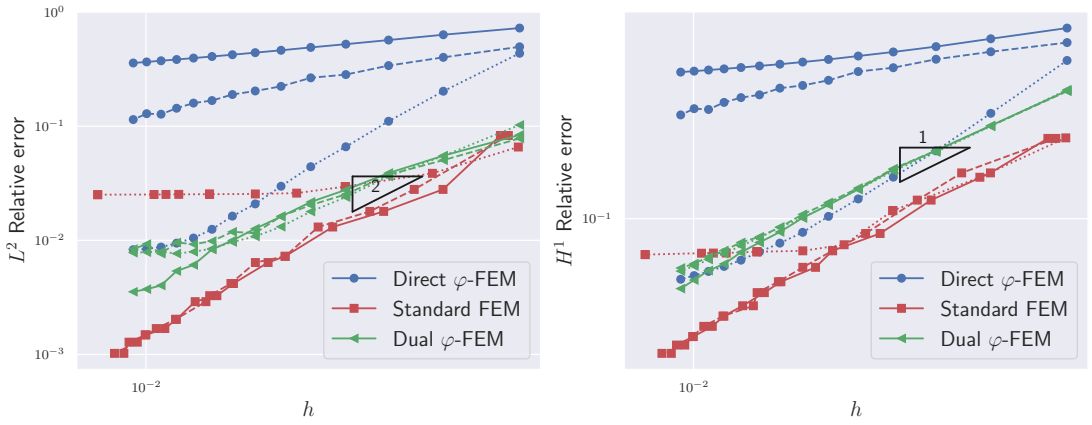

FIGURE 5.9 – Erreurs des différentes méthodes en norme relative L^2 (gauche) et en norme relative H^1 (droite).

On observe sur la Figure 5.9 différents points très intéressants. En ce qui concerne les approches Standard-FEM, sans surprise, les résultats obtenus avec un maillage reconstruit à partir de la distance signée sont relativement proches de ceux obtenus avec l'expression exacte. Cependant, les maillages reconstruits à partir de l'image initiale mènent à des erreurs qui stagnent très vite et donc des résultats peu précis. Concernant le schéma φ-FEM dual, comme on pouvait s'y attendre, l'expression exacte de la level-set φ donne les meilleurs résultats parmi les trois choix considérés. Cependant, il est très intéressant de remarquer que les deux méthodes de reconstruction entraînent des résultats très semblables pour la norme L^2 et la norme H^1. En particulier, il est intéressant de noter que la précision obtenue en norme L^2 atteint un plateau autour de 10^{-2}, liée à l'erreur de reconstruction de la fonction level-set. Enfin, concernant le schéma φ-FEM direct, les résultats sont très nettement améliorés lors de l'utilisation de la méthode basée sur les Gaussiennes en comparaison à l'utilisation de la distance signée ou à l'expression exacte

dont les gradients présentent des singularités, bien que la méthode reste particulièrement sensible à l'expression utilisée.

5.2 φ-FEM et l'approche « multigrid »

Dans la Section 3.5.3, nous avons proposé une approche « Multigrid » combinée au schéma φ-FD pour résoudre le problème de Poisson avec conditions de Dirichlet homogènes (1.1). Une extension naturelle à cette approche est une méthode φ-FEM combinée à la technique multigrid. Cette approche sera particulièrement intéressante à employer lors de la résolution de problèmes non-linéaires qui nécessitent l'utilisation de solveurs itératifs. Pour cela, nous allons présenter notre approche ainsi que des résultats numériques pour la résolution de deux problèmes. Dans un premier temps, l'équation de Poisson non-linéaire avec conditions de Dirichlet homogènes, de la forme

$$\begin{cases} -\nabla \cdot (q(u)\nabla u) &= f\,, \text{ dans } \Omega\,, \\ u &= 0\,, \quad \text{sur } \Gamma\,, \end{cases} \tag{5.4}$$

où $q(u)$ est une fonction rendant le problème non-linéaire, par exemple on considérera par la suite $q(u) = 1 + u^2 \exp(2u)$. Afin d'illustrer l'intérêt de notre méthode et son applicabilité dans un cas 3D, nous considérerons dans un second temps l'équation de Poisson-Dirichlet (1.1) sur une sphère.

5.2.1 Méthodologie

L'idée de départ est de construire une suite de raffinements $\mathcal{T}_h^{\mathcal{O},(i)}$ du maillage cartésien initial $\mathcal{T}_h^{\mathcal{O},(0)}$. Alors, pour chaque maillage intermédiaire $\mathcal{T}_h^{\mathcal{O},(i)}$, il reste à construire les domaines et maillages habituels φ-FEM : $\Omega_h^{(i)}$, $\mathcal{T}_h^{(i)}$, $\Omega_h^{\Gamma,(i)}$ et $\mathcal{T}_h^{\Gamma,(i)}$. On utilise ensuite un schéma φ-FEM pour résoudre chacun des problèmes intermédiaires : sur les maillages grossiers, avec un solveur direct et les maillages fins avec un solveur itératif.

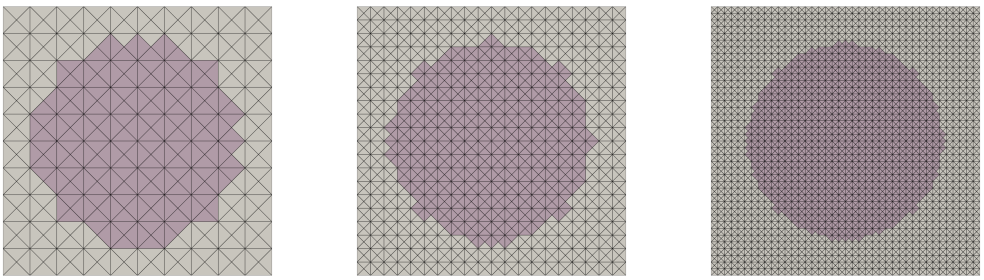

FIGURE 5.10 – Construction des maillages $\mathcal{T}_h^i$ correspondant à l'algorithme 3.

Différentes itérations des maillages $\mathcal{T}_h^i$ obtenus en raffinant $\mathcal{T}_h^{\mathcal{O}}$ sont représentées à la figure 5.10.

La méthode multigrid à appliquer est présentée dans l'algorithme simplifié 3, où l'on considère que l'on utilise seulement une étape initiale (résolution grossière) et une étape

finale (résolution fine). Cette solution a l'avantage de permettre d'initialiser le solveur itératif « fin » avec une solution proche de la solution recherchée, et donc de réaliser moins d'itérations du solveur itératif fin, ce qui représente un gain de temps non négligeable.

L'utilisation de la méthode φ-FEM offre ici un grand intérêt : en construisant correctement la grille cartésienne initiale, il sera possible de ne raffiner ensuite que les cellules de $\mathcal{T}_h$ et non plus de $\mathcal{T}_h^{\mathcal{O}}$. On pose ainsi $\mathcal{T}_h^0 = \mathcal{T}_h$. On construit alors une suite $(\mathcal{T}_h^{i+1})_i$, raffinements des maillages $\mathcal{T}_h^i$, où

$$\mathcal{T}_h^i := \left\{ T \in \mathcal{T}_h^{i,\mathcal{O}} : \ T \cap \{\varphi_h < 0\} \neq \emptyset \right\},$$

avec $\mathcal{T}_h^{i,\mathcal{O}}$ le raffinement de $\mathcal{T}_h^{i-1}$ pour $i > 1$ et $\mathcal{T}_h^{0,\mathcal{O}}$ la grille cartésienne initiale. En effet, comme représenté à la figure 5.10, chaque maillage $\mathcal{T}_h^{i+1}$ est contenu dans le maillage $\mathcal{T}_h^i$. Ainsi, le coût de construction des maillages $\mathcal{T}_h^i$ (pour $i > 0$) est bien moins élevé qu'en raffinant plusieurs fois $\mathcal{T}_h^{\mathcal{O}}$. On utilisera alors comme nouveau maillage initial, à chaque itération de raffinement, le maillage $\mathcal{T}_h$ précédent (i.e. le maillage le plus clair sur les figures de 5.11).

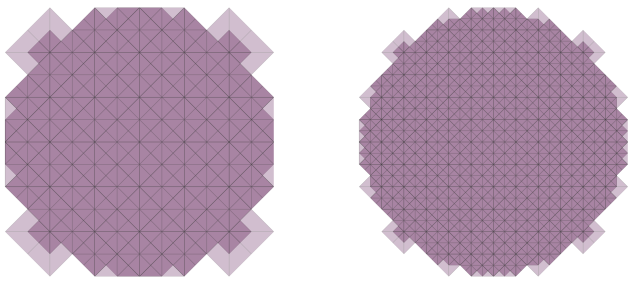

FIGURE 5.11 – Construction des maillages $\mathcal{T}_h^i$ correspondant à l'algorithme 4.

Algorithme 3 : φ-FEM-M « Brute force »

Entrées : N : nombre d'étapes de raffinement, N_s : nombre de résolutions φ-FEM ($N_s = 2$), n : nombre de cellules dans chaque direction

pour $i = 0$ *à* N_s **faire**
  si $i = 0$ **alors**
    Construire $\mathcal{T}_h^{\mathcal{O},(0)}$, avec $n \times n$ cellules
    $\mathcal{T}_h^{\mathcal{O}} = \mathcal{T}_h^{\mathcal{O},(0)}$
  sinon
    pour $j = 1$ *à* $N + 1$ **faire**
      $\mathcal{T}_h^{\mathcal{O},(j)} =$ Raffiner$(\mathcal{T}_h^{\mathcal{O},(j-1)})$
    $\mathcal{T}_h^{\mathcal{O}} = \mathcal{T}_h^{\mathcal{O},(N)}$
  Construire $\mathcal{T}_h$, $\mathcal{T}_h^{\Gamma}$, $\mathcal{F}_h^{\Gamma}$ et la formulation variationnelle
  si $i = 0$ **alors**
    Initialiser le solveur avec $u = 0$
  sinon
    Interpoler u (solution grossière) sur $\mathcal{T}_h$ (fin)
    Initialiser le solveur avec $u = I_h u_0$
  Résoudre $F(u, v) = 0$

Algorithme 4 : φ-FEM-M

Entrées : N : nombre d'étapes de raffinement, N_s : nombre de résolutions φ-FEM ($N_s = 2$), n : nombre de cellules dans chaque direction

pour $i = 0$ *à* N_s **faire**
  si $i = 0$ **alors**
    Construire $\mathcal{T}_h^{0,\mathcal{O}}$, avec $n \times n$ cellules
    $\mathcal{T}_h^{\mathcal{O}} = \mathcal{T}_h^{0,\mathcal{O}}$
  sinon
    $\mathcal{T}_h^{0,\mathcal{O}} = \mathcal{T}_h$
    pour $j = 1$ *à* $N + 1$ **faire**
      $\mathcal{T}_h^{j,\mathcal{O}} =$ Raffiner$(\mathcal{T}_h^{j-1,\mathcal{O}})$
    $\mathcal{T}_h^{\mathcal{O}} = \mathcal{T}_h^{N,\mathcal{O}}$
  Construire $\mathcal{T}_h$, $\mathcal{T}_h^{\Gamma}$, $\mathcal{F}_h^{\Gamma}$ et la formulation variationnelle
  si $i = 0$ **alors**
    Initialiser le solveur avec $u = 0$
  sinon
    Interpoler u (solution grossière) sur $\mathcal{T}_h$ (fin)
    Initialiser le solveur avec $u = I_h u_0$
  Résoudre $F(u, v) = 0$

Une représentation graphique de la pipeline appliquée dans l'Algorithme 4 est donnée à la Figure 5.12, dans le cas de conditions de Dirichlet homogènes (i.e. $u = 0$ sur Γ).

Remarque 5.4. La méthode de raffinement appliquée est la méthode classique implémentée dans le package DolfinX, dont plusieurs étapes sont représentées pour un cas simple à la Figure 5.13.

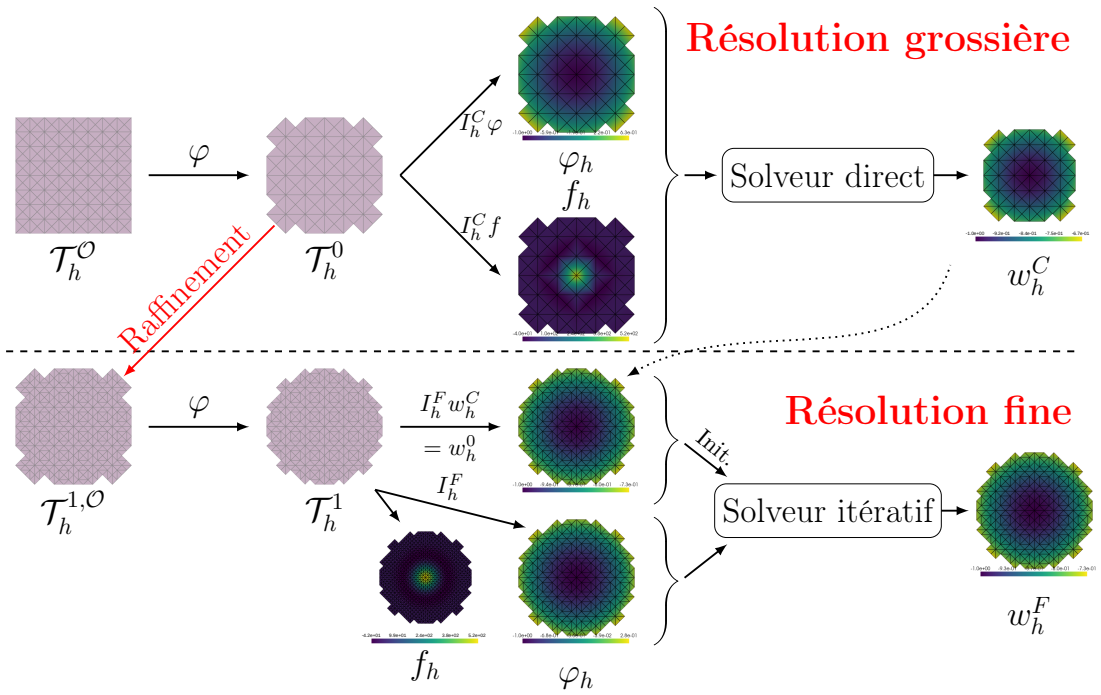

FIGURE 5.12 – Représentation graphique de la pipeline φ-FEM-M, dans le cas de conditions de Dirichlet homogènes.

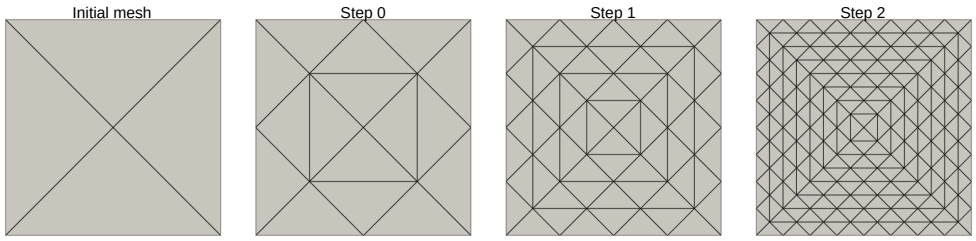

FIGURE 5.13 – Exemple de raffinements successifs du maillage initial.

5.2.2 Résultats numériques

Nous allons maintenant illustrer l'intérêt de cette approche sur plusieurs cas test numériques, en comparaison avec la méthode Standard-FEM et la méthode classique φ-FEM.

Cas test 1 : résolution de l'équation (5.4) **sur un disque** Dans un premier temps nous considérons l'équation (5.4) définie sur le disque de centre $(0.5, 0.5)$ et de rayon $\sqrt{2}/4$. Le calcul d'erreur sera fait à l'aide d'une solution manufacturée radiale satisfaisant $u = 0$ sur Γ, donnée par

$$u = \cos\left(\frac{\pi}{2} r\right)$$

avec $r = \frac{1}{R}\sqrt{(x-0.5)^2+(y-0.5)^2}$.
On choisit également $q(u) = 1 + u^3\exp(2.5u)$ et on calcule analytiquement f.

Le schéma φ-FEM utilisé sera le schéma direct (1.10), adapté à l'équation (5.4), écrit sous la forme : trouver $w_h \in V_h^{(k)}$ avec $u_h = \varphi_h w_h$ telle que

$$\int_{\Omega_h} q(u_h)\nabla u_h \cdot \nabla v_h - \int_{\partial\Omega_h} q(u_h)(\nabla u_h)n \cdot v_h + G_h(u_h, v_h) + J_h(u_h, v_h) - \int_{\Omega_h} f \cdot v_h = 0\,,\ \forall v_h \in V_h^k\,,$$

où

$$J_h(u,v) := \sigma_D h^2 \int_{\Omega_h^\Gamma} (\mathrm{div}(q(u)\nabla u) + f) \cdot \mathrm{div}(q(u)\nabla v)\,,$$

et

$$G_h(u,v) := \sigma h \sum_{E \in \mathcal{F}_h^\Gamma} \int_E [q(u)\nabla u \cdot n] \cdot [q(u)\nabla v \cdot n]\ .$$

Les résultats obtenus par la méthode φ-FEM, la méthode éléments finis standard et notre nouvelle approche φ-FEM-M sont représentés à la Figure 5.14, illustrant l'intérêt de notre approche. Sur ce premier cas test, nous avons fixé la tolérance des solveurs itératifs pour φ-FEM-M à 10^{-5} et avons choisi comme résolution grossière $1/4$ de la résolution fine : on choisit de faire une résolution grossière sur une grille $N \times N$ puis on raffine deux fois le maillage avant de résoudre le problème. Ainsi, notre méthode atteint ici presque les performances de φ-FEM en termes de précision, et les dépasse très clairement en temps de calcul. De plus, aussi bien en erreur qu'en temps de calcul, notre méthode donne de meilleurs résultats que la méthode standard : pour un seuil d'erreur fixé, φ-FEM-M est plus rapide que la méthode standard. On considère également une seconde version de φ-FEM-M, où la résolution grossière est constante à 20×20, essentielle pour l'applicabilité de la méthode que nous présenterons ensuite. Cette approche, notée φ-FEM Multigrid 2 sur la Figure 5.14 donne également de très bons résultats, aussi bien en termes d'erreurs que de temps de calcul.

Cas test 2 : Équation (1.1) **sur une sphère** Appliquons maintenant notre méthode à un cas 3D. Pour cela on considère l'équation (1.1), avec une solution de référence radiale, donnée par

$$u = \cos\left(\frac{\pi}{2}r\right)$$

avec $r = \frac{1}{R}\sqrt{(x-0.5)^2+(y-0.5)^2+(z-0.5)^2}$ sur la sphère centrée en $(0.5, 0.5, 0.5)$, de rayon $R = \sqrt{2}/4$.

Le schéma φ-FEM utilisé sera celui introduit en Section 1.2, à l'équation (1.10). Comme nous l'avons vu dans la Section 3.5.3 dédiée à φ-FD-multigrid, il est presque impossible de résoudre ce problème avec un solveur direct. Ainsi, nous allons comparer notre approche uniquement à des solveurs itératifs (à chaque fois le Gradient BiConjugué Stabilisé) pour φ-FEM et Standard-FEM. Les résultats de la Figure 5.15 illustrent une nouvelle fois l'intérêt de notre approche, puisqu'elle permet de diminuer l'erreur ainsi que le temps de calcul.

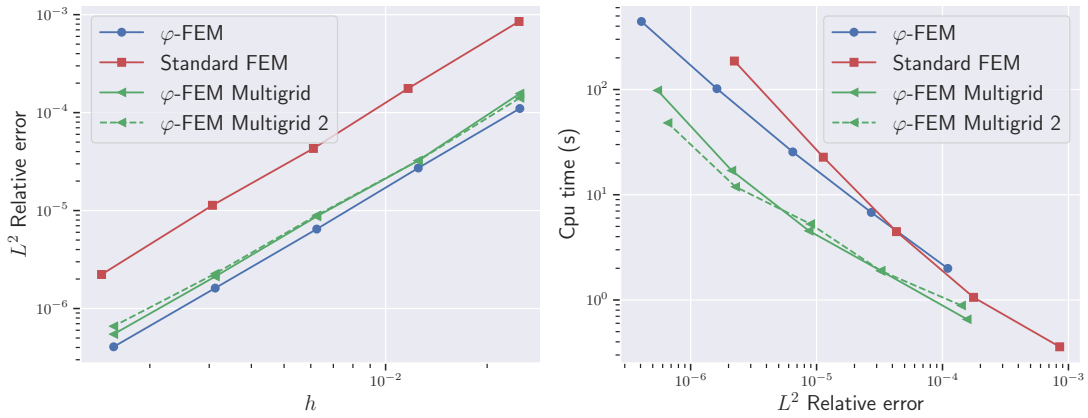

FIGURE 5.14 – **Cas test 1.** Gauche : erreurs relatives L^2 en fonction de la taille de cellule. Droite : temps de calcul en fonction de l'erreur relative L^2.

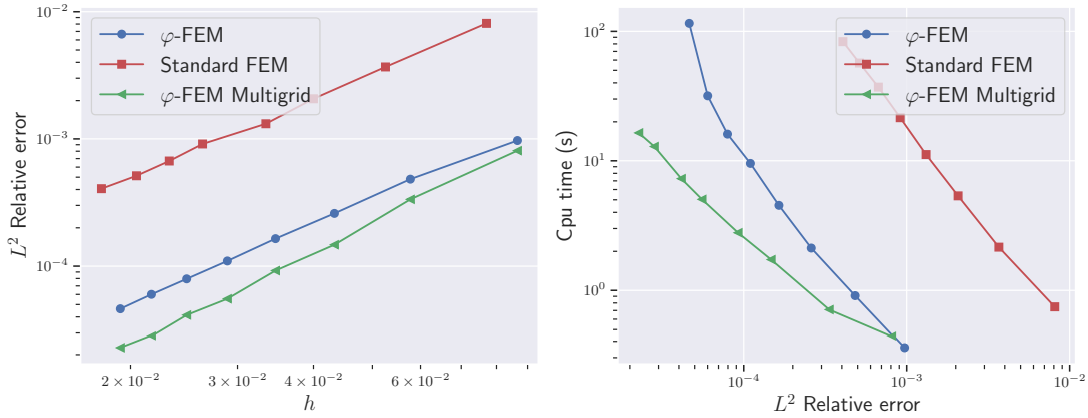

FIGURE 5.15 – **Cas test 2.** Gauche : erreurs relatives L^2 en fonction de h. Droite : temps de calcul en fonction de l'erreur relative L^2.

5.3 φ-FEM-M-FNO : une nouvelle méthode hybride

Finalement, une idée naturelle au regard de ce qui a été présenté précédemment, est une combinaison des méthodes φ-FEM-FNO et φ-FEM-M. En effet, entraîner un FNO à une résolution grossière fixée permettrait ainsi d'éviter la première résolution éléments finis. Cela a alors plusieurs intérêts en termes de coût de calcul :

- la génération de données est plus rapide que pour l'utilisation seule de φ-FEM-FNO puisqu'elle peut être effectuée sur des grilles relativement grossières ;
- le coût d'entraînement peut également être réduit puisque les tenseurs peuvent être de dimensions réduites (tant en résolution qu'en nombre de données) ;
- le coût de φ-FEM-M est réduit : la première résolution fine est évitée, tout comme les différentes interpolations sur le maillage initial.

Cependant, il est important de préciser que cette approche nécessite une étape coûteuse numériquement. En effet, lorsque l'on effectue une prédiction à l'aide de la méthode φ-FEM-FNO, on obtient une solution sous la forme d'une matrice (ou d'un tenseur) qu'il sera nécessaire de discrétiser sous la forme d'une fonction éléments finis. Pour cela, on utilisera les valeurs de la matrice (correspondant aux valeurs nodales) d'une fonction éléments finis associée. Cependant, cette étape n'est pas optimale sous FEniCSX et nécessite de construire un mapping entre l'ordre des degrés de liberté de l'espace éléments finis associé et l'ordre des valeurs dans le tenseur solution.

Remarque 5.5. Il serait bien évidemment possible de ne pas utiliser la méthode multigrid et de prédire uniquement une solution initiale pour les solveurs itératifs sur le maillage à la résolution désirée. On retrouve par exemple cette idée de prédiction utilisée pour initialiser un solveur de Newton dans [72]. Cependant, cela nécessite alors de générer des données à la résolution fine et d'entraîner le réseau pour la même résolution, ce qui augmente considérablement le coût de calcul total. De plus, il est alors nécessaire de construire le maillage $\mathcal{T}_h$ à partir de la grille cartésienne maillée finement, ce qui est également numériquement relativement lourd.

Une autre idée de combinaison entre méthode éléments finis et réseaux de neurones proposée par [6] démontre également de très bons résultats. Cette méthode, combinant PINNs et Standard-FEM, est construite dans l'idée de corriger une prédiction de réseau de neurones (effectuée sur un nombre élevé de points) avec une méthode éléments finis appliquée sur un maillage grossier.

5.3.1 Pipeline

Une représentation graphique de la pipeline de φ-FEM-M-FNO est donnée à la Figure 5.16, dans le cas de conditions de Dirichlet non homogènes (i.e. $u = g$ sur Γ).
L'approche consiste en trois étapes importantes :

- Résolution grossière : prédiction d'une solution grossière **et** construction du maillage $\mathcal{T}_h^0$;
- Raffinement : boucle de raffinement pour atteindre la résolution souhaitée ;
- Résolution fine : interpolation de la solution grossière sur le maillage fin et résolution φ-FEM classique avec un solveur itératif initialisé avec la solution précédemment déterminée.

5.3.2 Cas test numériques

Cas test 1 : le cas 2D

Pour ce premier cas test, nous allons considérer l'équation (5.4) avec des conditions de Dirichlet non homogènes $u = g$ sur Γ, où g est donnée par (4.12). Les géométries seront définies par des level-set φ de la forme (5.3). Pour l'entraînement du FNO, on choisit de générer 500 données d'entraînement et 300 de validation. On effectue 2000 itérations et on cherche à minimiser la fonctionnelle $\mathcal{L}$ définie par (4.5). On compare alors cette nouvelle approche à la méthode φ-FEM classique, à Standard-FEM ainsi qu'à la méthode φ-FEM-M présentée précédemment.

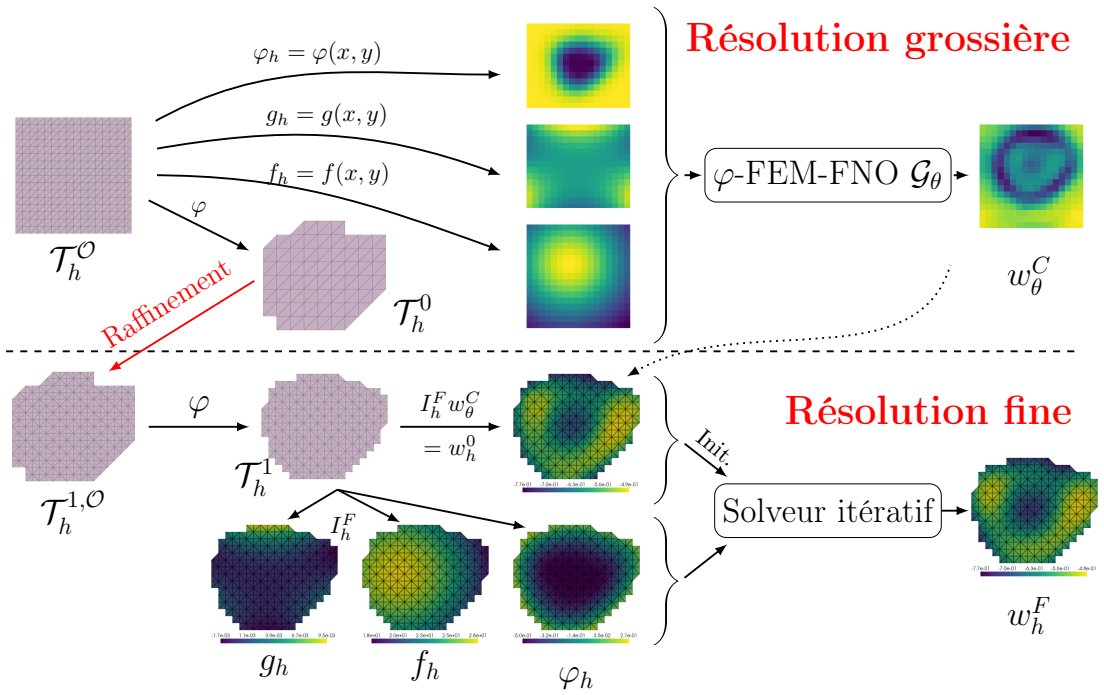

FIGURE 5.16 – Représentation graphique de la pipeline φ-FEM-M-FNO, dans une situation correspondant au cas test 1.

Un exemple de solution obtenue pour ce problème est représenté à la Figure 5.17. Dans les deux situations suivantes, on considère 5 données issues d'un jeu de données de test pour étudier numériquement les différentes méthodes.

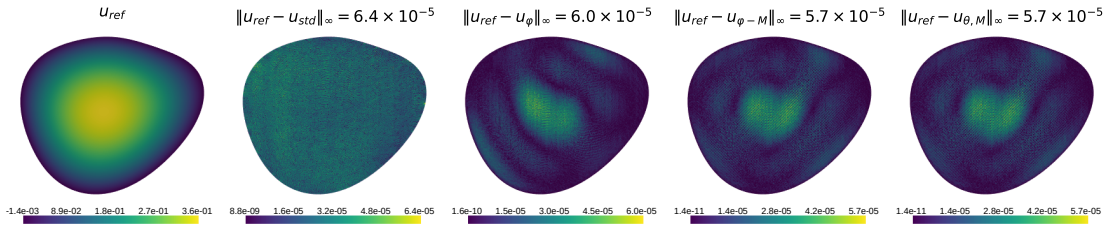

FIGURE 5.17 – **Cas test 1.** De gauche à droite : solution de référence, puis différences entre la solution de référence et la projection de la solution Standard-FEM (u_{std}), de la solution φ-FEM (u_φ), de la solution φ-FEM-M ($u_{\varphi-M}$), et de la solution φ-FEM-M-FNO ($u_{\theta,M}$).

φ-FEM-M-FNO : grilles 16×16 On considère dans un premier temps des données générées sur des grilles cartésiennes de taille 16×16. Pour la comparaison, la résolution grossière de la méthode φ-FEM-M, est réalisée sur une grille de même résolution.
On choisit alors de comparer les approches pour différentes tailles de grille fine : 32, 64, 128 et 256.

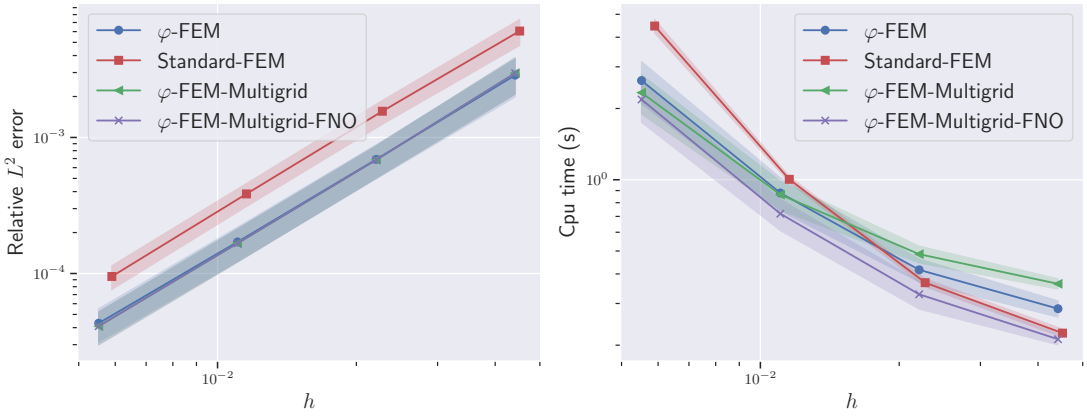

FIGURE 5.18 – **Cas test 1, données** 16×16**.** Gauche : erreur relative L^2 en fonction de h. Droite : temps de calcul des méthodes.

On voit alors sur les résultats présentés à la Figure 5.18 (gauche) que les trois méthodes basées sur φ-FEM donnent des erreurs comparables, moins élevées que les erreurs de Standard-FEM. Cependant, il est intéressant de remarquer (c.f. Figure 5.18 (droite)) que ces résultats sont systématiquement obtenus plus rapidement avec l'approche φ-FEM-M-FNO. Ainsi, malgré un entraînement réalisé avec des données très grossières, on obtient des résultats déjà très intéressants pour notre approche hybride. Cela se remarque notamment dans la Table 5.2 où la méthode donne toujours les meilleurs résultats.

Résolution	Méthode	Temps (grossier)	Temps (fin)	Temps (total)	Erreur relative
32 × 32	Standard-FEM	\	\	0.23	6.06×10^{-3}
	φ-FEM	\	\	0.28	2.87×10^{-3}
	φ-FEM-M	0.16	0.20	0.36	2.96×10^{-3}
	φ-FEM-M-FNO	0.004	0.21	0.21	2.96×10^{-3}
64 × 64	Standard-FEM	\	\	0.36	1.56×10^{-3}
	φ-FEM	\	\	0.41	6.90×10^{-4}
	φ-FEM-M	0.16	0.32	0.48	6.85×10^{-4}
	φ-FEM-M-FNO	0.004	0.32	0.32	6.85×10^{-4}
128 × 128	Standard-FEM	\	\	1	3.85×10^{-4}
	φ-FEM	\	\	0.88	1.71×10^{-4}
	φ-FEM-M	0.16	0.70	0.86	1.67×10^{-4}
	φ-FEM-M-FNO	0.004	0.70	0.70	1.67×10^{-4}
256 × 256	Standard-FEM	\	\	4.46	9.51×10^{-5}
	φ-FEM	\	\	2.62	4.31×10^{-5}
	φ-FEM-M	0.16	2.17	2.33	4.1×10^{-5}
	φ-FEM-M-FNO	0.004	2.17	2.18	4.1×10^{-5}

TABLE 5.2 – **Cas test 1, données** 16×16**.** Résultats des différentes méthodes. Les temps et erreurs correspondent aux valeurs moyennes sur 5 nouvelles données.

φ-FEM-M-FNO : grilles 32×32 Dans un second temps, il est intéressant d'étudier les résultats de la méthode lors d'un entraînement sur des données plus fines. Pour cela, on entraîne φ-FEM-FNO avec des données générées sur des grilles 32×32 et on utilise la même résolution grossière pour φ-FEM-M. On calcule alors l'erreur des différentes méthodes avec des résolutions fines sur des grilles 64×64, 128×128, 256×256 et 512×512. On remarque sur les résultats présentés à la Figure 5.19 l'intérêt de cette nouvelle approche. En effet, l'initialisation du solveur étant faite avec une prédiction plus précise et donc plus proche de la solution, le coût de calcul est plus faible que pour toutes les autres méthodes. De plus, à tolérance fixée comme critère d'arrêt pour les solveurs itératifs (10^{-9}), les deux approches multigrid offrent ici une meilleure précision que les deux autres méthodes classiques. On remarque en particulier à la Table 5.3 que l'approche φ-FEM-M-FNO est toujours la plus rapide en temps total, bien que le temps de résolution grossière soit plus élevé que dans le cas 16×16, notamment en raison de la conversion entre les tenseurs *numpy* et les vecteurs DolfinX.

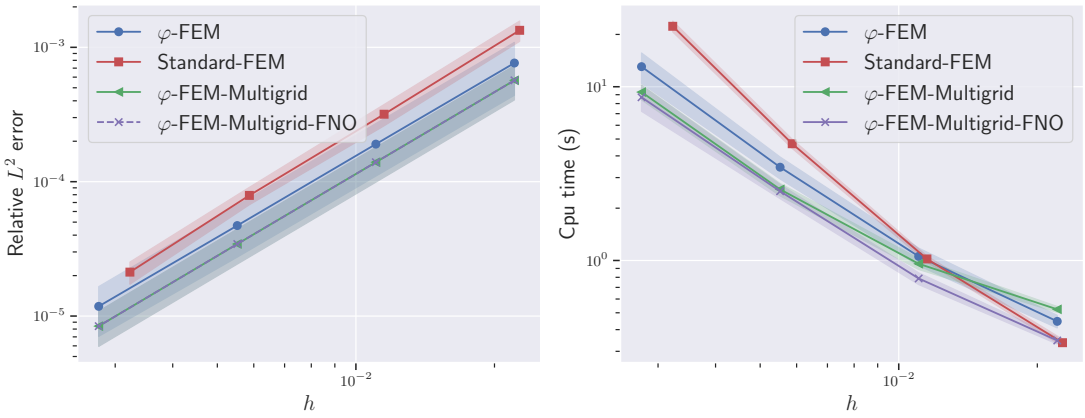

FIGURE 5.19 – **Cas test 1, données 32×32.** Gauche : erreur relative L^2 en fonction de h. Droite : temps de calcul des méthodes.

Cas test 2 : le cas 3D Pour le second cas test, nous allons comparer les trois méthodes basées sur l'approche φ-FEM. Pour cela, nous considérons le problème de Poisson avec conditions de Dirichlet non-homogènes, (4.1) sur des géométries complexes 3D définies à partir de fonctions gaussiennes, i.e. en adaptant l'équation (5.3) au cas 3D, et donc en utilisant des fonctions φ définies par

$$\varphi(x,y,z) = (-1)^n \prod_j^n \left(-1 + \exp\left(-\frac{x_j^2}{2l_{x,j}^2} - \frac{y_j^2}{2l_{y,j}^2} - \frac{z_j^2}{2l_{z,j}^2} \right) \right),$$

où

$$\begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = R_z(\theta_z)\, R_y(\theta_y)\, R_x(\theta_x) \begin{bmatrix} x - \mu_x \\ y - \mu_y \\ z - \mu_z \end{bmatrix},$$

Résolution	Méthode	Temps (grossier)	Temps (fin)	Temps (total)	Erreur relative
64 × 64	Standard-FEM	⁄	⁄	0.39	1.34×10^{-3}
	φ-FEM	⁄	⁄	0.50	7.64×10^{-4}
	φ-FEM-M	0.24	0.34	0.58	5.68×10^{-4}
	φ-FEM-M-FNO	0.011	0.36	0.37	5.68×10^{-4}
128 × 128	Standard-FEM	⁄	⁄	1.02	3.18×10^{-4}
	φ-FEM	⁄	⁄	1.09	1.90×10^{-4}
	φ-FEM-M	0.23	0.75	0.98	1.40×10^{-4}
	φ-FEM-M-FNO	0.011	0.88	0.90	1.40×10^{-4}
256 × 256	Standard-FEM	⁄	⁄	4.29	7.91×10^{-5}
	φ-FEM	⁄	⁄	3.46	4.71×10^{-5}
	φ-FEM-M	0.22	2.34	2.56	3.44×10^{-5}
	φ-FEM-M-FNO	0.011	2.49	2.50	3.44×10^{-5}
512 × 512	Standard-FEM	⁄	⁄	22.06	2.12×10^{-5}
	φ-FEM	⁄	⁄	13.16	1.18×10^{-5}
	φ-FEM-M	0.23	9.12	9.35	8.41×10^{-6}
	φ-FEM-M-FNO	0.011	8.62	8.64	8.41×10^{-6}

TABLE 5.3 – **Cas test 1, données** 32×32**.** Résultats des différentes méthodes. Les temps et erreurs correspondent aux valeurs moyennes sur 5 nouvelles données.

avec

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}, \quad R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix},$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

où l'ensemble des paramètres est choisi aléatoirement avec la seule contrainte que la géométrie construite soit connexe.

Le second membre f et les conditions de bord g sont eux adaptés de (4.11) et (4.12) et sont donnés par

$$f_{(A,\mu_0,\mu_1,\mu_2,\sigma_x,\sigma_y,\sigma_z)}(x,y,z) = A\exp\left(-\frac{(x-\mu_0)^2}{2\sigma_x^2} - \frac{(y-\mu_1)^2}{2\sigma_y^2} - \frac{(z-\mu_2)^2}{2\sigma_z^2}\right),$$

et

$$g_{(\alpha,\beta)}(x,y) = \alpha\left((x-0.5)^2 - (y-0.5)^2\right)\cos\left(\beta z\pi\right),$$

où les différents paramètres sont choisis aléatoirement selon des distributions uniformes.

Pour l'approche basée sur le FNO, on génère un ensemble de 250 données séparées en 200 données d'entraînement et 50 données de validation. Ces données sont générées sur

des grilles cartésiennes de résolution $20 \times 20 \times 20$ et on réalise un entraînement sur 200 epochs, en utilisant des batches de taille 8 et en fixant le nombre de modes conservés dans chaque direction à 8. De plus, la fonctionnelle à minimiser est l'adaptation 3D de la norme H^1, définie par (4.5).

Une fois l'opérateur φ-FEM-FNO entraîné, on considère un échantillon de 6 nouvelles données de test pour évaluer les performances des trois approches φ-FEM. Une représentation de 3 des 6 solutions de référence obtenues par Standard-FEM sur un maillage fin est donnée à la Figure 5.20 afin d'illustrer la variabilité des géométries considérées.

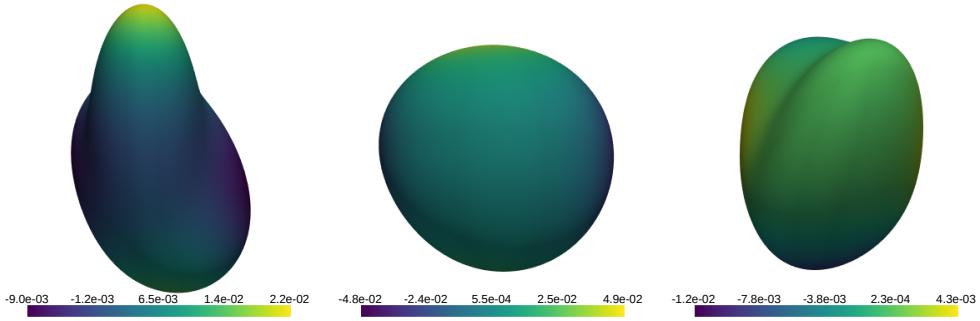

FIGURE 5.20 – **Cas test 2.** Représentation de 3 solutions de référence obtenues par Standard-FEM.

La résolution utilisée pour la génération des données étant fixée à $20 \times 20 \times 20$, afin de comparer les différentes méthodes, nous considérerons 3 nouvelles résolutions pour les tests : $40 \times 40 \times 40$, $80 \times 80 \times 80$ et $160 \times 160 \times 160$. Le solveur itératif pour toutes les méthodes sera une nouvelle fois le Gradient BiConjugué Stabilisé avec une tolérance de convergence fixée à 10^{-9}. Pour φ-FEM-M, la résolution grossière sera réalisée avec un solveur direct sur une grille de taille $N/2$. Enfin, pour l'approche φ-FEM le solveur itératif sera combiné à un pré-conditionneur LU.

Pour illustrer l'intérêt de notre approche φ-FEM-M-FNO, l'erreur relative L^2 et le temps de calcul sont mesurés. Les résultats présentés à la Figure 5.21 où l'erreur moyenne est représentée en fonction du temps de calcul moyen (avec les zones de couleurs indiquant les écarts-types) montrent que φ-FEM-M-FNO est systématiquement plus rapide que φ-FEM-M qui elle est plus rapide que φ-FEM. De plus les erreurs entre les deux approches multigrid sont comparables, et plus faibles que celles de la méthode φ-FEM classique.

5.4 Conclusion

Dans ce dernier chapitre, nous avons présenté deux méthodes permettant d'utiliser des fonctions level-set, bases de la méthode φ-FEM en pratique.
Dans un premier temps, nous avons présenté une méthode utilisée pour construire des maillages conformes à partir de fonctions level-set. Nous avons ensuite présenté plusieurs méthodes permettant d'appliquer les méthodes φ-FEM et Standard-FEM à des images binaires.

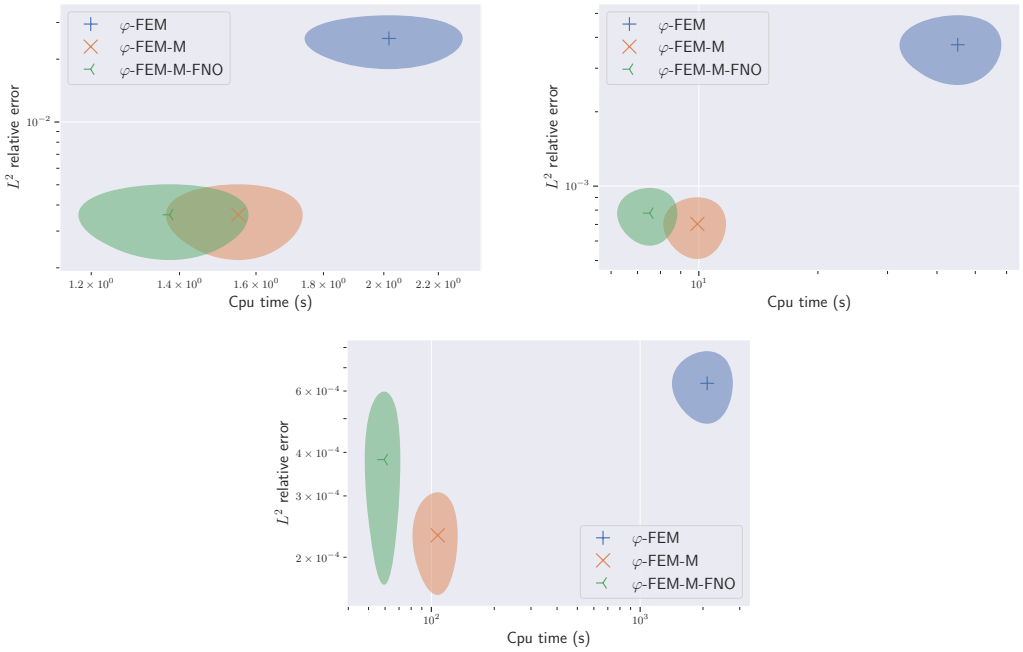

FIGURE 5.21 – **Cas test 2.** Résultats obtenus avec les différentes méthodes φ-FEM. En haut à gauche : résolution $40 \times 40 \times 40$. En haut à droite : résolution $80 \times 80 \times 80$. En bas : résolution $160 \times 160 \times 160$.

Par la suite, nous avons proposé une nouvelle architecture, basée sur l'approche multigrid, permettant de réduire le coût de calcul de la méthode φ-FEM. Une brève étude numérique a alors mis en évidence les gains offerts par l'utilisation de cette méthode par rapport à φ-FEM ainsi qu'à Standard-FEM.

Finalement, à partir de l'architecture φ-FEM-M et de la méthode φ-FEM-FNO présentées précédemment, nous avons construit une nouvelle méthode hybride qui a offert des résultats très intéressants, notamment dans le cas 3D puisqu'elle permet de réduire le coût de calcul de l'approche φ-FEM-M.

6 Conclusion

Dans ce travail, nous avons introduit et analysé différentes déclinaisons de la méthode φ-FEM, en montrant sa pertinence pour le traitement d'équations aux dérivées partielles sur des géométries complexes. L'étude s'est articulée autour de plusieurs axes complémentaires. Nous avons d'abord considéré l'adaptation de la méthode à divers types de conditions aux limites et d'équations modèles, telles que le problème de Poisson, l'équation de la chaleur et les problèmes d'élasticité, ce qui a permis de mettre en évidence sa robustesse et ses performances par rapport aux approches éléments finis classiques. Dans un second temps, nous avons proposé une méthode en différences finies, nommée φ-FD, directement inspirée de φ-FEM. Celle-ci conserve les atouts théoriques et numériques de convergence de la méthode initiale, tout en offrant une implémentation simplifiée.

Une autre direction majeure a été l'intégration de φ-FEM avec des approches issues de l'apprentissage automatique, en particulier les architectures de type Fourier Neural Operator. L'approche hybride φ-FEM-FNO a montré sa capacité à accélérer considérablement les calculs tout en conservant une précision satisfaisante, cela avec un volume de données d'entraînement limité, y compris dans des situations complexes.

Parallèlement, nous avons étudié la mise en œuvre pratique de la méthode φ-FEM et de la méthode Standard-FEM à partir de fonctions level-set et d'images binaires. Enfin, nous avons proposé une architecture multigrid dédiée à la réduction du coût de calcul de l'approche φ-FEM. La combinaison de cet outil avec l'approche φ-FEM-FNO a conduit à une méthode hybride particulièrement prometteuse, notamment pour les problèmes tridimensionnels, où les gains de performance se sont révélés significatifs.

Dans l'ensemble, ce travail souligne le potentiel de la méthode φ-FEM, à la fois comme alternative aux méthodes classiques et comme socle pour des développements hybrides intégrant des techniques modernes d'apprentissage et de calcul haute performance.

Ces résultats ouvrent néanmoins de nombreuses perspectives de recherche. Les schémas permettant de traiter les conditions mixtes, que ce soit pour le problème de Poisson ou pour les problèmes d'élasticité, ont été étudiés numériquement, mais leur l'aspect théorique reste encore à traiter. De plus, les problèmes d'élasticité considérés tout au long de ce manuscrit étaient limités au cadre statique, alors que l'extension aux problèmes dynamiques constitue un prolongement naturel et essentiel pour la modélisation de phénomènes concrets. Plus largement, l'adaptation de la méthode φ-FEM au cadre de

l'élasticité non linéaire représente un défi théorique et numérique majeur, mais également une étape incontournable vers des applications concrètes, notamment en biomécanique. La possibilité d'exploiter directement des données issues d'images confère à φ-FEM un avantage naturel pour la simulation du comportement mécanique d'organes, domaine dans lequel la prise en compte d'effets non linéaires est indispensable. L'association de cette approche avec des techniques d'apprentissage automatique, telles que φ-FEM-FNO, pourrait alors permettre de développer des outils capables de fournir des prédictions rapides et fiables, ouvrant la voie à des applications en chirurgie assistée par ordinateur ou en planification thérapeutique personnalisée.

Parallèlement, la méthode φ-FD offre, elle aussi, de nombreuses pistes de recherche. Son extension aux conditions de Neumann constitue une évolution naturelle du travail présenté ici. De plus, la compatibilité de sa structure régulière avec les architectures de type FNO ouvre la perspective d'une méthode φ-FD-FNO dont la comparaison avec φ-FEM-FNO serait particulièrement intéressante. Enfin, les méthodes hybrides développées dans ce manuscrit, telles que φ-FEM-FNO, φ-FEM-M et φ-FEM-M-FNO, restent largement à explorer. Leur application à des problèmes dynamiques, à des conditions mixtes ou à des configurations tridimensionnelles de grande taille constitue un champ de recherche riche.

En conclusion, les contributions présentées dans ce manuscrit témoignent de la richesse et de la flexibilité de la méthode φ-FEM et de ses variantes. Elles montrent que cette approche, bien au-delà de sa robustesse numérique, constitue un cadre de développement particulièrement adapté aux défis actuels du calcul scientifique, et qu'elle possède le potentiel de s'imposer comme un outil de référence pour la simulation de phénomènes complexes, à l'interface entre mathématiques appliquées, calcul haute performance et apprentissage automatique.

A Annexes du Chapitre 3

A.1 Exemple de code `python` pour φ-FD

```
import numpy as np
import scipy.sparse as sp
from scipy.sparse.linalg import spsolve

# Radius of the domain
R = 0.3 + 1e-10

# Parameter of penalization and stabilization
sigma, gamma = 0.01, 1.0

# Construction of the grid
Nx, Ny = 100, 100
x, y = np.linspace(0, 1, Nx + 1), np.linspace(0, 1, Ny + 1)
hx, hy = x[1] - x[0], y[1] - y[0]
X, Y = np.meshgrid(x, y)

# Computation of the exact solution, exact source term and the
    level-set
r = lambda x, y: np.sqrt((x - 0.5) * (x - 0.5) + (y - 0.5) * (y
    - 0.5) + 1e-12)
K = np.pi / 2 / R
ue = lambda x, y: np.cos(K * r(x, y))
f = lambda x, y: K * K * np.cos(K * r(x, y)) + K * np.sin(K * r
    (x, y)) / r(x, y)
phi = lambda x, y: (x - 0.5) * (x - 0.5) + (y - 0.5) * (y -
    0.5) - R * R
phiij = phi(X, Y)
ind = (phiij < 0) + 0
mask = sp.diags(diagonals=ind.ravel())
```

```
indOut = 1 - ind

# Laplacian matrix
D2x = (1 / hx / hx) * sp.diags(
    diagonals=[-1, 2, -1], offsets=[-1, 0, 1], shape=(Nx + 1,
   Nx + 1)
)
D2y = (1 / hy / hy) * sp.diags(
    diagonals=[-1, 2, -1], offsets=[-1, 0, 1], shape=(Ny + 1,
   Ny + 1)
)
D2x_2d = sp.kron(sp.eye(Ny + 1), D2x)
D2y_2d = sp.kron(D2y, sp.eye(Nx + 1))
A = mask @ (D2x_2d + D2y_2d)

# Boundary conditions
diag = np.zeros((Nx + 1) * (Ny + 1))
diagxp = np.zeros((Nx + 1) * (Ny + 1) - 1)
diagxm = np.zeros((Nx + 1) * (Ny + 1) - 1)
diagyp = np.zeros((Nx + 1) * Ny)
diagym = np.zeros((Nx + 1) * Ny)
actGx = np.zeros((Ny + 1, Nx + 1))
actGy = np.zeros((Ny + 1, Nx + 1))

indx = ind[:, 1 : Nx + 1] - ind[:, 0:Nx]
J, I = np.where((indx == 1) | (indx == -1))
for k in range(np.shape(I)[0]):
    if indx[J[k], I[k]] == 1:
        indOut[J[k], I[k]], actGx[J[k], I[k] + 1] = 0, 1
    else:
        indOut[J[k], I[k] + 1], actGx[J[k], I[k]] = 0, 1
phiS = np.square(phiij[J, I]) + np.square(phiij[J, I + 1])
diag[I + (Nx + 1) * J] = phiij[J, I + 1] * phiij[J, I + 1] /
   phiS
diagxp[I + (Nx + 1) * J] = -phiij[J, I] * phiij[J, I + 1] /
   phiS
diag[I + 1 + (Nx + 1) * J] = phiij[J, I] * phiij[J, I] / phiS
diagxm[I + (Nx + 1) * J] = -phiij[J, I] * phiij[J, I + 1] /
   phiS

indy = ind[1 : Ny + 1, :] - ind[0:Ny, :]
J, I = np.where((indy == 1) | (indy == -1))
for k in range(np.shape(I)[0]):
    if indy[J[k], I[k]] == 1:
        indOut[J[k], I[k]], actGy[J[k] + 1, I[k]] = 0, 1
```

```
    else:
        indOut[J[k] + 1, I[k]], actGy[J[k], I[k]] = 0, 1
phiS = np.square(phiij[J, I]) + np.square(phiij[J + 1, I])
diag[I + (Nx + 1) * J] += phiij[J + 1, I] * phiij[J + 1, I] /
    phiS
diagyp[I + (Nx + 1) * J] = -phiij[J, I] * phiij[J + 1, I] /
    phiS
diag[I + (Nx + 1) * (J + 1)] += phiij[J, I] * phiij[J, I] /
    phiS
diagym[I + (Nx + 1) * J] = -phiij[J, I] * phiij[J + 1, I] /
    phiS

B = (gamma / hx / hy) * sp.diags(
    diagonals=(diagym, diagxm, diag, diagxp, diagyp),
    offsets=(-Nx - 1, -1, 0, 1, Nx + 1),
)

# Stabilization
maskGx = sp.diags(diagonals=actGx.ravel())
maskGy = sp.diags(diagonals=actGy.ravel())
C = sigma * hx * hy * (D2x_2d.T @ maskGx @ D2x_2d + D2y_2d.T @
    maskGy @ D2y_2d)

# Penalization outside
D = sp.diags(diagonals=indOut.ravel())

# Linear system
A, b = (A + B + C + D).tocsr(), (ind * f(X, Y)).ravel()
u = spsolve(A, b).reshape(Ny + 1, Nx + 1)

# Computation of the errors
uref = ue(X, Y)
e = ind * (u - uref)
eL2 = np.linalg.norm(e) * np.sqrt(hx * hy)
emax = np.linalg.norm(e, np.inf)
print(eL2, emax)
```

Listing A.1 – Implementation Python de φ-FD.

B Adaptation du *learning rate* dans le contexte d'apprentissage en ligne

Dans cette Annexe, nous présentons un travail réalisé dans le cadre du « Treizième atelier de résolution de problèmes industriels de Montréal » qui s'est déroulé du 21 au 25 août 2023. Il a été réalisé en collaboration avec Jean-Bernard Hayet, Amey Kaloti, Samir Karam, Jean-Pierre Noot, Nassim Razaaly, Sébastien Tran Tien et Killian Verdure sur un sujet proposé par Brigitte Jaumard et Jean-Michel Sellier.

L'apprentissage automatique en ligne (*online learning*) désigne un ensemble de méthodes d'apprentissage supervisé où les données arrivent en continu et ne peuvent ainsi pas être stockées pour réaliser un unique entraînement ultérieur. Contrairement à l'apprentissage automatique traditionnel (hors ligne), où l'ensemble de données est fixe, dans l'apprentissage en ligne les données prennent la forme d'une série temporelle, seules les dernières valeurs étant disponibles à un moment donné.

Comme nous l'avons expliqué au Chapitre 4, lors d'un entraînement d'une méthode de machine learning, il est nécessaire de spécifier un taux d'apprentissage (*learning rate*) qui influe sur la convergence de la méthode d'optimisation. Cependant, pour les méthodes en ligne, ce paramètre détermine la réactivité du modèle face à de nouvelles données. Une partie de la difficulté de l'apprentissage en ligne réside ainsi dans le choix de ce paramètre : il doit permettre aux prédictions de prendre en compte les observations futures tout en restant cohérentes avec les observations passées.

Pour relever ce défi, nous proposons une méthode d'optimisation basée sur la descente d'hyper-gradient (hypergradient descent).

B.1 Définition du problème

Dans les problèmes de régression traditionnels hors ligne (*offline*), on dispose d'un jeu de données d'entraînement :

$$\mathbf{D}_0 = \{\mathbf{x}_i, \mathbf{y}_i\}_{1 \leq i \leq M}$$

comportant M observations, qui sert à entraîner (c'est-à-dire optimiser) un modèle $f_\theta(\mathbf{x})$ paramétré par θ.

Par exemple, dans les cas de prévision que nous verrons plus loin, $\mathbf{x} \in \mathbb{R}^p$ représente un ensemble de p données observées auparavant à un instant t :

$$(d_t, d_{t+1}, d_{t+2}, \ldots, d_{t+p-1}),$$

et $\mathbf{y} \in \mathbb{R}^f$ est l'ensemble des f valeurs futures à prédire :

$$(d_{t+p}, d_{t+p+1}, \ldots, d_{t+p+f-1}).$$

Pour déterminer les paramètres θ appropriés, on résout un problème d'optimisation de la forme :

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta),$$

où $\mathcal{L}$ est la fonction objectif à minimiser, qui dans ce cas de régression prend généralement la forme :

$$\mathcal{L}(\theta) = \frac{1}{M} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}_0} \|\mathbf{y}_i - f_\theta(\mathbf{x}_i)\|^2.$$

Pour résoudre ce problème d'optimisation, la plupart des approches utilisent des variantes de l'algorithme de descente de gradient (*gradient descent*, GD). Globalement, GD suit la direction de la plus forte descente. En partant d'une estimation initiale θ_0, on applique des itérations de mise à jour des paramètres selon :

$$\theta_k = \theta_{k-1} - \alpha \nabla \mathcal{L}(\theta_{k-1}),$$

où α est appelé *pas de gradient* ou *taux d'apprentissage* (*learning rate*). Dans la littérature, il est fréquent d'ajuster empiriquement la valeur de ce paramètre au cours d'expériences et de le conserver constant, mais nous verrons dans la section suivante qu'il existe de nombreuses manières de le faire évoluer pendant l'entraînement.

En apprentissage en ligne (*online learning*), on suppose encore qu'au départ, on dispose d'un jeu de données d'entraînement $\mathbf{D}_0$ de M observations, avec lequel on peut entraîner un modèle initial. La différence est que l'on continue à entraîner le modèle avec de nouvelles données $\mathbf{D}_\tau$, comprenant $N \ll M$ observations, arrivant à intervalles réguliers τ, et que l'on utilise pour mettre à jour les paramètres θ.

Ainsi, on produit régulièrement de nouvelles estimations « optimales » des paramètres θ à partir de ces nouvelles données, selon une règle :

$$\theta^{(\tau)*} = g(\theta^{(\tau-1)*}, \mathbf{D}_\tau).$$

Notons que cette règle de mise à jour g peut être choisie de différentes manières, avec une large gamme de comportements allant d'une dépendance exclusive aux dernières données $\mathbf{D}_\tau$ (et en oubliant la phase d'entraînement initiale) à une dépendance exclusive aux données les plus anciennes (et en ignorant les plus récentes). Dans ce projet, nous supposons que ce compromis est atteint en appliquant un nombre limité K d'itérations de descente de gradient sur les dernières données :

$$\theta_0^{(\tau)} = \theta^{(\tau-1)*} \tag{B.1}$$

$$\theta_k^{(\tau)} = \theta_{k-1}^{(\tau)} - \alpha \nabla \mathcal{L}^{(\tau)}(\theta_{k-1}^{(\tau)}) \tag{B.2}$$

$$\theta^{(\tau)*} = \theta_K^{(\tau)}. \tag{B.3}$$

Remarquons que ces étapes peuvent utiliser soit l'algorithme de descente de gradient stochastique (où l'on utilise une approximation Monte-Carlo du gradient), soit le vrai gradient (où l'on utilise toutes les données). Le choix entre les deux n'est pas particulièrement important dans notre contexte.

La fonctionnelle (*loss function*) utilisée à l'itération τ est donnée par

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}_\tau} \|\mathbf{y}_i - f_\theta(\mathbf{x}_i)\|^2.$$

La question que nous abordons ici est la suivante : dans le contexte en ligne, comment déterminer le « meilleur » pas de gradient à utiliser ? Les principales difficultés rencontrées sont :

- la possibilité d'un décalage de données (*data shift*) entre les $\mathbf{D}_\tau$, ce qui peut mener à un problème d'optimisation très différent entre τ et $\tau + 1$;
- la présence de fortes contraintes de temps à respecter, ce qui peut restreindre le nombre de méthodes utilisables.

B.2 Revue des méthodes existantes

B.2.1 Méthodes d'évolution de α dans le cas hors ligne

La stratégie la plus courante concernant le taux d'apprentissage est de le maintenir constant. Dans de nombreuses applications récentes, une exploration coûteuse de l'espace des hyperparamètres peut être effectuée afin de déterminer la meilleure valeur, par exemple via une recherche en grille (*grid search*). Cependant, ces méthodes sont beaucoup trop lentes pour une applicabilité à l'apprentissage en ligne.

Des techniques plus complexes peuvent être conçues pour faire évoluer ce taux d'apprentissage. Dans le cas des méthodes traditionnelles hors ligne, les principes guidant ces techniques sont généralement les suivants :

- lorsque l'erreur se dégrade ou a tendance à osciller, il est probablement préférable de réduire α afin d'effectuer des pas plus petits ;
- lorsque l'erreur diminue de façon constante, mais lente, il est alors préférable d'augmenter le taux d'apprentissage pour accélérer la convergence ;
- lorsque l'on approche de la fin de l'entraînement et que θ se rapproche de son optimum, on réduit généralement le taux d'apprentissage afin d'éviter une trop grande sensibilité aux variations entre *batches*.

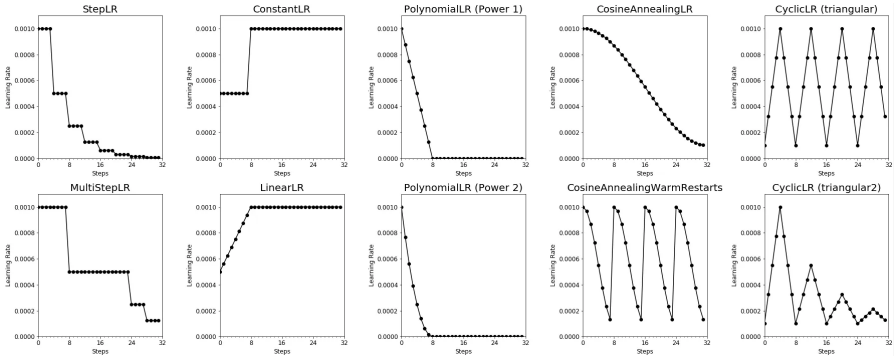

FIGURE B.1 – Exemples de *learning rate schedulers*. Extrait de [70].

Ces principes étant généraux, il est complexe de construire un algorithme systématique permettant de les appliquer en pratique. Une première famille courante de méthodes consiste à utiliser le nombre d'itérations pour faire varier le taux d'apprentissage, selon

$$\alpha_k \triangleq s(k),$$

où s est une fonction du nombre actuel d'itérations k. Quelques exemples de telles fonctions d'évolution (appelées *learning rate schedulers*) sont donnés à la Fig. B.1. En général, cette fonction est choisie comme étant décroissante (vers une valeur proche de zéro), mais son choix implique un réglage, par exemple en réalisant plusieurs entraînements, étape que nous ne pouvons pas nous permettre de réaliser dans le contexte de l'apprentissage en ligne.

B.2.2 Méthodes systématiques pour adapter le taux d'apprentissage

Une manière d'éviter le problème de la détermination de α consiste à ne pas utiliser de méthodes du premier ordre (utilisant le gradient) et à se tourner vers des méthodes du second ordre (utilisant la matrice hessienne). Cependant, nous écartons cette option ici, car elle pourrait être très coûteuse, notamment à cause de l'évaluation de la matrice hessienne de la fonction de coût par rapport aux paramètres du réseau.

Pour les méthodes du premier ordre, la recherche linéaire (*line search*) fournit un cadre général pour déterminer le meilleur pas dans l'itération de descente de gradient. La méthode décrite plus loin dans la Section B.2.2 peut être considérée comme un cas particulier de cette famille de méthodes. L'idée générale est que, étant donné un algorithme de descente de gradient, on optimise le pas α en annulant la dérivée de la fonction objectif par rapport à α. Il est facile de montrer que cela conduit à choisir des directions de descente consécutives qui sont orthogonales. Nous verrons que la solution proposée réalise implicitement la même chose.

MacLaurin *et al.* [62] proposent un algorithme très générique pour estimer les gradients de la fonction objectif d'un réseau de neurones par rapport à ses hyperparamètres. L'algorithme fournit le gradient exact et pourrait également être utilisé en considérant le taux d'apprentissage comme un hyperparamètre particulier ; cependant, il nécessite de multiples appels à l'auto-différentiation du réseau pour produire les dérivées souhaitées.

Van Erven et Koolen [87] se sont intéressés à la conception de méthodes adaptatives pouvant automatiquement obtenir des convergences rapides, sans réglage manuel. Les taux d'apprentissage ne décroissent pas de manière monotone dans le temps et ne sont pas ajustés sur la base d'une borne théorique, mais sont pondérés directement en proportion de leurs performances empiriques sur les données, en utilisant un algorithme de pondération exponentielle biaisée (*tilted exponential weights*).

La méthode que nous proposons ici s'inspire de Baydin *et al.* [7], et optimise également l'hyperparamètre α. Cependant, nous verrons qu'elle nécessite un nombre minimal d'opérations supplémentaires par rapport à la descente de gradient standard. D'une certaine manière, elle peut être considérée comme une méthode apprenant le taux d'apprentissage, et nous verrons qu'elle peut s'intégrer facilement dans un schéma en ligne.

B.3 Solution proposée

B.3.1 Méthode d'optimisation

Nous utilisons une variante de la descente de gradient connue sous le nom de descente hyper-gradient (*hypergradient descent*), proposée dans [7]. Comme mentionné ci-dessus, en apprentissage en ligne, nous gérons deux boucles :

- une boucle externe sur les *batches* de données des instants τ ;
- une boucle interne optimisant les paramètres sur le batch spécifique $\mathbf{D}_\tau$.

Nous mettons à jour α à chaque itération de la boucle interne. Nous utiliserons donc la notation $\alpha_k^{(\tau)}$ pour désigner le taux d'apprentissage utilisé pour le batch $\mathbf{D}_\tau$ à l'itération k. La règle de mise à jour pour $\alpha^{(\tau)}$ est donnée, par analogie avec la descente de gradient stochastique, par :

$$\alpha_k^{(\tau)} = \alpha_{k-1}^{(\tau)} - \beta \frac{\partial \mathcal{L}^{(\tau)}(\theta_{k-1})}{\partial \alpha},$$

où β est appelé taux d'apprentissage d'hyper-gradient (*hyper-gradient learning rate*).

En utilisant la règle de la dérivée en chaîne, on obtient :

$$\frac{\partial \mathcal{L}^{(\tau)}(\theta_{k-1}^{(\tau)})}{\partial \alpha} = \nabla \mathcal{L}(\theta_{k-1}^{(\tau)})^T \frac{\partial \theta_{k-1}^{(\tau)}}{\partial \alpha}. \tag{B.4}$$

En utilisant ensuite la règle de mise à jour de l'équation B.2 et en prenant la dérivée par rapport à α, on obtient :

$$\frac{\partial \theta_{k-1}^{(\tau)}}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left(\theta_{k-2}^{(\tau)} - \alpha \nabla \mathcal{L}^{(\tau)}(\theta_{k-2}^{(\tau)})\right) = -\nabla \mathcal{L}^{(\tau)}(\theta_{k-2}^{(\tau)}). \tag{B.5}$$

En combinant les équations B.4 et B.5, nous obtenons finalement la règle de mise à jour :

$$\alpha_k^{(\tau)} = \alpha_{k-1}^{(\tau)} + \beta \, \nabla \mathcal{L}^{(\tau)}(\theta_{k-1}^{(\tau)})^T \nabla \mathcal{L}^{(\tau)}(\theta_{k-2}^{(\tau)}). \tag{B.6}$$

Analysons les termes de cette équation : la mise à jour de l'équation B.6 est calculée à partir d'un produit scalaire entre deux gradients. Le premier est le gradient de la

fonctionnelle de coût par rapport aux paramètres du modèle que l'on utilise comme direction de descente pour la mise à jour des paramètres ; le second est exactement le même gradient évalué à l'itération précédente. Ainsi, après une première itération où l'on utilise éventuellement une valeur initiale $\alpha^{(\tau)}$, on poursuit dans la boucle interne en optimisant (partiellement) le modèle sur $\mathbf{D}_\tau$ tout en mettant à jour simultanément α avec l'équation B.6. Le coût supplémentaire est donc uniquement lié à la conservation en mémoire du gradient de l'itération précédente, ainsi qu'au calcul du produit scalaire entre deux gradients consécutifs.

Un autre point important à noter est l'introduction d'un nouvel hyperparamètre β ; conservé constant dans [7]. Nous verrons dans les expériences que sa valeur a un impact important sur la manière dont évolue α.

Notons également la connexion avec la recherche linéaire : lorsque le paramètre α est proche de son optimum, les incréments deviennent nuls, ce qui se traduit par des produits scalaires nuls entre deux directions de recherche consécutives, exactement comme le prédit la recherche linéaire.

Dans le même article [7], les auteurs étendent ce schéma d'hyper-gradient à d'autres algorithmes d'optimisation du premier ordre, en particulier ADAM [51], en utilisant la règle de mise à jour ci-dessus pour son taux d'apprentissage. Ces optimiseurs seront ceux utilisés dans nos expériences par la suite.

Algorithme 5 : Apprentissage en ligne avec learning rate adaptatif

$\mathbf{D}_0 \leftarrow$ AcquisitionDesObservationsInitiales()
$\theta_0^{(0)} \leftarrow \theta_{init}$
pour $k \in [1, K']$ **faire**
    $\theta_k^{(0)} \leftarrow \theta_{k-1}^{(0)} - \alpha \nabla \mathcal{L}^{(0)}(\theta_{k-1}^{(0)})$ **si** $k > 1$ **alors**
        $\alpha_k^{(0)} = \alpha_{k-1}^{(0)} + \beta \nabla \mathcal{L}^{(0)}(\theta_{k-1}^{(0)})^T \nabla \mathcal{L}^{(0)}(\theta_{k-2}^{(0)})$.
    sinon
        $\alpha_1^{(0)} = \alpha_0^{(0)}$.
$\tau \leftarrow 1$
tant que *True* **faire**
    $\mathbf{D}_\tau \leftarrow$ AcquisitionDesObservations()
    $\alpha_0^{(\tau)} \leftarrow \alpha_{K'}^{(\tau-1)}$
    pour $k \in [1, K']$ **faire**
        $\theta_k^{(\tau)} \leftarrow \theta_{k-1}^{(\tau)} - \alpha \nabla \mathcal{L}^{(\tau)}(\theta_{k-1}^{(\tau)})$ **si** $k > 1$ **alors**
            $\alpha_k^{(\tau)} = \alpha_{k-1}^{(\tau)} + \beta \nabla \mathcal{L}^{(\tau)}(\theta_{k-1}^{(\tau)})^T \nabla \mathcal{L}^{(\tau)}(\theta_{k-2}^{(\tau)})$.
        sinon
            $\alpha_1^{(\tau)} = \alpha_0^{(\tau)}$.
    $\tau \leftarrow \tau + 1$

B.3.2 Schéma d'apprentissage en ligne

Dans l'algorithme 5, nous décrivons le schéma d'apprentissage en ligne proposé, incluant les mises à jour par hyper-gradient, conformément aux descriptions données ci-dessus. Il comporte deux phases :

1. une première phase hors ligne d'entraînement sur $\mathbf{D}_0$ pour initialiser le modèle ;
2. puis un cycle d'observation optimisé sur les batches successifs $\mathbf{D}_\tau$.

La valeur initiale de α à l'instant τ est prise comme la dernière valeur obtenue à l'instant $\tau - 1$.

Nous concevons nos expériences pour des problèmes de prévision, c'est-à-dire la prédiction de f valeurs futures dans une série temporelle à partir de l'observation de p valeurs passées.

B.3.3 Modèles et fonctionnelles de coût

Nous utilisons un réseau à « Mémoire à Long Court Terme » (*Long Short Term Memory*, LSTM) [47], une forme de réseaux de neurones récurrents. Ces réseaux sont spécialisés dans le traitement de séquences de données. Décrivons plus précisément ce qu'est un LSTM. Soit $\mathbf{h}_t$ l'état caché, $\mathbf{c}_t$ l'état de cellule, $\mathbf{x}_t$ l'entrée au temps t et $\mathbf{h}_{t-1}$ l'état caché de la couche au temps $t-1$. De plus, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{g}_t$ et $\mathbf{o}_t$ sont des valeurs intermédiaires appelées respectivement porte d'entrée, porte d'oubli, porte de cellule et porte de sortie. Enfin, σ désigne la fonction d'activation sigmoïde.

Une couche LSTM cherche à résumer le contenu d'une séquence de données observée jusqu'à l'instant t dans les vecteurs $\mathbf{h}_t$ et $\mathbf{c}_t$, selon les règles de mise à jour suivantes :

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi})\,, \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf})\,, \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg})\,, \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho})\,, \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t\,, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)\,,
\end{aligned}
$$

où $\odot$ désigne le produit élément par élément.

La sortie à un instant t de la séquence est le vecteur caché $\mathbf{h}_t$. Une représentation de cette couche est donnée dans la Fig. B.2.

Pour réaliser l'entraînement du LSTM, nous considérons comme fonction de coût l'erreur quadratique moyenne (*Mean Square Error*, MSE). Elle est calculée sur l'ensemble du jeu de données pour la partie hors ligne (nous n'utilisons pas de descente de gradient stochastique, mais uniquement le vrai gradient sur l'ensemble complet des données). Pour l'entraînement en ligne, nous calculons la fonctionnelle de coût uniquement sur les données reçues en ligne, c'est-à-dire chaque nouveau lot de données.

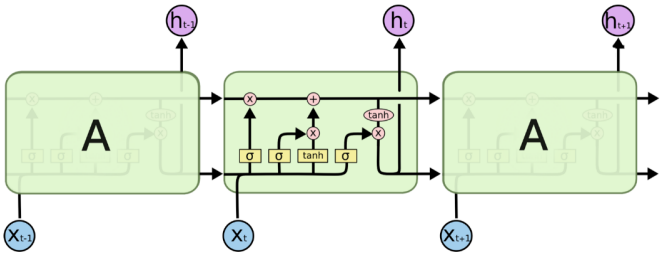

FIGURE B.2 – Architecture du LSTM utilisé (issu de [73]).

B.4 Résultats expérimentaux

B.4.1 Mise en place expérimentale

Nous considérons un jeu de données de températures couvrant la période de 2009 à 2016, fourni par le Max Planck Institute. Il est divisé en trois parties correspondant à des sous-ensembles consécutifs : les données d'entraînement, de validation et de test, comme illustré dans la Fig. B.3.

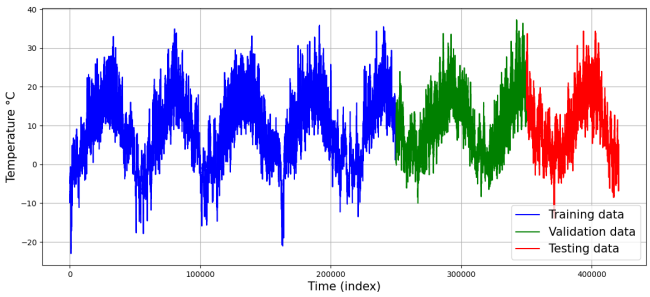

FIGURE B.3 – Séparation du jeu de données.

B.4.2 Évaluation de l'impact de la descente d'hypergradient

Nous comparons l'évolution des fonctionnelles d'entraînement et de validation en utilisant l'optimiseur Adam d'une part, et Adam combiné à la descente d'hypergradient, que nous appelons *ADAM-HD*, d'autre part. Pour cela, dans la Fig. B.4, nous entraînons un LSTM sur 100 epochs. Les lignes pleines correspondent aux valeurs de loss obtenues avec l'optimiseur ADAM classique, avec un taux d'apprentissage fixé à $\alpha = 10^{-4}$. Les lignes pointillées correspondent aux valeurs de loss obtenues avec ADAM-HD, en partant de la même valeur initiale $\alpha = 10^{-4}$. Le taux d'apprentissage d'ADAM-HD converge vers une valeur d'environ 10^{-2}, ce qui entraîne une accélération de la convergence de la fonctionnelle, et donc une convergence plus rapide que celle obtenue avec l'Adam original.

Dans la Figure B.5, nous comparons ADAM-HD, partant de $\alpha = 10^{-4}$, à l'optimiseur ADAM classique, mais démarrant directement avec la valeur optimale du taux d'apprentis-

sage donnée par ADAM-HD. Ici, les fonctionnelles d'ADAM convergent plus rapidement que précédemment. Ainsi, le choix du taux d'apprentissage semble optimal. Il pourrait alors être intéressant de commencer un processus d'entraînement avec ADAM-HD jusqu'à convergence du taux d'apprentissage, puis de continuer avec un Adam classique en utilisant la valeur optimale obtenue par ADAM-HD, ce qui est laissé pour des travaux futurs.

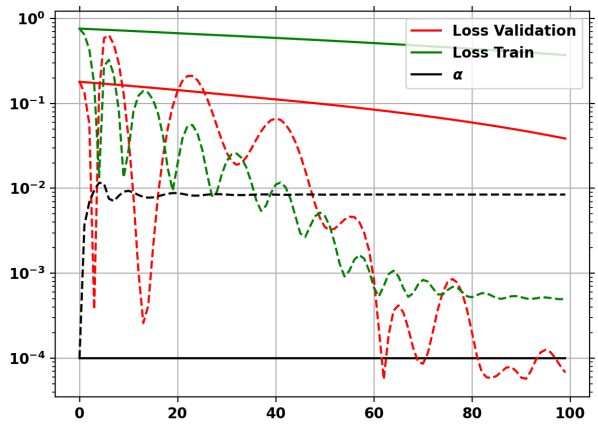

FIGURE B.4 – Pointillés : ADAM-HD. Trait plein : ADAM avec $\alpha = 10^{-4}$ initial.

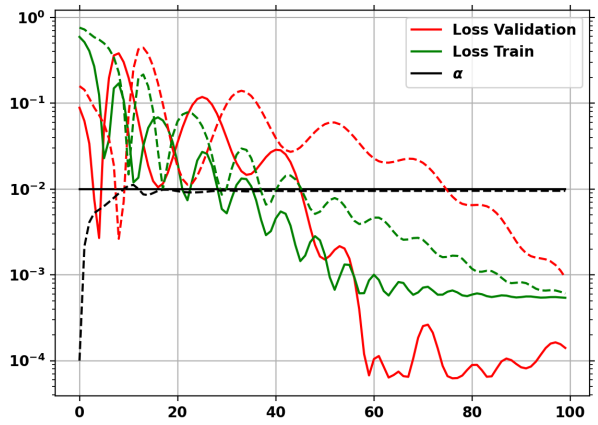

FIGURE B.5 – Pointillés : ADAM-HD. Trait plein : ADAM avec $\alpha = 10^{-2}$ initial comme donné par la méthode ADAM-HD à la Figure B.4.

B.5 Conclusion

L'utilisation de la descente de gradient hyper-paramétrique s'est révélée être une approche prometteuse, renforçant l'adaptabilité des modèles et soulignant son intérêt dans le contexte dynamique de l'apprentissage en ligne. Nous avons exploré son implémentation pratique en développant des modèles de type « Réseaux à Mémoire Long Court Terme » (LSTM), spécialement conçus pour la prédiction de séries temporelles. Ces modèles, testés sur des jeux de données réalistes, nous ont permis de comparer l'optimiseur ADAM classique avec sa version enrichie par la descente d'hypergradient. Cette analyse a non seulement renforcé notre compréhension du choix du taux d'apprentissage et de son impact, mais elle nous a également conduits à proposer une solution efficace aux défis de l'apprentissage en ligne dans des contextes où la distribution des données d'entrée évolue au cours du temps.

Bibliographie

[1] L. Adams. A multigrid algorithm for immersed interface problems. In *NASA Conference Publication*, pages 1–14, 1996.

[2] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, and G.N. Wells. Archive of numerical software : The FEniCS project version 1.5. *University Library Heidelberg*, 2015.

[3] M. S. Alnaes, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language : A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40, 2014.

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

[5] I. A. Baratta, J. P. Dean, J. S. Dokken, M. Habera, J. S. Hale, C. N. Richardson, M. E. Rognes, M. W. Scroggs, N. Sime, and G. N. Wells. DOLFINx : the next generation FEniCS problem solving environment. preprint, 2023.

[6] H. Barucq, M. Duprez, F. Faucher, E. Franck, F. Lecourtier, V. Lleras, V. Michel-Dansac, and N. Victorion. Enriching continuous Lagrange finite element approximation spaces using neural networks. working paper or preprint, February 2025.

[7] A.G. Baydin, R. Cornish, D. M. Rubio, M. Schmidt, and F. Wood. Online learning rate adaptation with hypergradient descent. In *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, April 30 – May 3 2018.

[8] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge university press, 1997.

[9] J. H. Bramble and B. E. Hubbard. On the formulation of finite difference analogues of the Dirichlet problem for Poisson's equation. *Numer. Math.*, 4 :313–327, 1962.

[10] S. Brenner and L. Scott. The mathematical theory of finite element methods, 2008.

[11] Dorin Bucur, Alessandro Giacomini, and Paola Trebeschi. Best constant in poincaré inequalities with traces : A free discontinuity approach. *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, 36(7) :1959–1986, 2019.

[12] E. Burman. Ghost penalty. *C. R. Math. Acad. Sci. Paris*, 348(21-22) :1217–1220, 2010.

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

[14] E. Burman, D. Elfverson, P. Hansbo, M. Larson, and K. Larsson. Hybridized CutFEM for elliptic interface problems. *SIAM J. Sci. Comput.*, 41(5) :A3354–A3380, 2019.

[15] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements : I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199(41) :2680–2686, 2010.

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

[17] E. Burman, P. Hansbo, and M. G. Larson. Low Regularity Estimates for CutFEM Approximations of an Elliptic Problem with Mixed Boundary Conditions, 2020.

[18] T. Carraro and S. Wetterauer. On the implementation of the eXtended Finite Element Method (XFEM) for interface problems, 2015.

[19] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.

[20] M. Cisternino and L. Weynans. A parallel second order cartesian method for elliptic interface problems. *Communications in Computational Physics*, 12(5) :1562–1587, 2012.

[21] S. Claus and P. Kerfriden. A stable and optimally convergent LaTIn-CutFEM algorithm for multiple unilateral contact problems. *International Journal for Numerical Methods in Engineering*, 113(6) :938–966, 2018.

[22] S. Cotin, M. Duprez, V. Lleras, A. Lozinski, and K. Vuillemot. ϕ-FEM : An efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer. *Partition of Unity Methods*, pages 191–216, 2023.

[23] M. Duprez, V. Lleras, and A. Lozinski. A new ϕ-FEM approach for problems with natural boundary conditions. *Numer. Methods Partial Differential Equations*, 39(1) :281–303, 2023.

[24] M. Duprez, V. Lleras, and A. Lozinski. φ-FEM : an optimally convergent and easily implementable immersed boundary method for particulate flows and Stokes equations. *ESAIM Math. Model. Numer. Anal.*, 57(3) :1111–1142, 2023.

[25] M. Duprez, V. Lleras, A. Lozinski, V. Vigon, and K. Vuillemot. φ-FD : A well-conditioned finite difference method inspired by φ-FEM for general geometries on elliptic PDEs. *Journal of Scientific Computing*, 104(1) :1–27, 2025.

[26] M. Duprez, V. Lleras, A. Lozinski, V. Vigon, and K. Vuillemot. φ-FEM-FNO : A new approach to train a Neural Operator as a fast PDE solver for variable geometries. *Communications in Nonlinear Science and Numerical Simulation*, 152 :109131, 2026.

[27] M. Duprez, V. Lleras, A. Lozinski, and K. Vuillemot. φ-FEM for the heat equation : optimal convergence on unfitted meshes in space. *Comptes Rendus. Mathématique*, 361 :1699–1710, 2023.

[28] M. Duprez and A. Lozinski. φ-FEM : a finite element method on domains defined by level-sets. *SIAM J. Numer. Anal.*, 58(2) :1008–1028, 2020.

[29] W. Ee and B. Yu. The Deep Ritz Method : A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Communications in Mathematics and Statistics*, 6, 09 2017.

[30] S. El Hadramy, N. Padoy, and S. Cotin. Hyperu-mesh : Real-time deformation of soft-tissues across variable patient-specific parameters. In M. Kobielarz, A. Wittek, M. P. Nash, P. Nielsen, A. R. Babu, and K. Miller, editors, *Computational Biomechanics for Medicine*, pages 129–139, Cham, 2025. Springer Nature Switzerland.

[31] Charles M Elliott and Thomas Ranner. Finite element analysis for a coupled bulk–surface partial differential equation. *IMA Journal of Numerical Analysis*, 33(2) :377–402, 2013.

[32] A. Ern and J.L. Guermond. *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer New York, 2004.

[33] L.C. Evans. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.

[34] R. P. Fedorenko. Iteration methods for the solution of elliptic difference equations. *Uspehi Mat. Nauk*, 28(2(170)) :121–182, 1973.

[35] J. A. Ferreira and R. D. Grigorieff. On the supraconvergence of elliptic finite difference schemes. *Applied numerical mathematics*, 28(2-4) :275–292, 1998.

[36] A. Fortin and A. Garon. Les éléments finis : de la théorie à la pratique. `https://giref.ulaval.ca/afortin/elements_finis.pdf`, 1999.

[37] F. Gibou, R.P. Fedkiw, L.-T. Cheng, and K. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *Journal of Computational Physics*, 176(1) :205–227, 2002.

[38] V. Girault and R. Glowinski. Error analysis of a fictitious domain method applied to a Dirichlet problem. *Japan Journal of Industrial and Applied Mathematics*, 12(3) :487, 1995.

[39] R. Glowinski, T. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3-4) :283–303, 1994.

[40] T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb. Can physics-informed neural networks beat the finite element method ? *IMA Journal of Applied Mathematics*, 89(1) :143–174, 05 2024.

[41] W. Hackbusch. A fast iterative method for solving Poisson's equation in a general region. In *Numerical treatment of differential equations (Proc. Conf., Math. Forschungsinst., Oberwolfach, 1976)*, volume Vol. 631 of *Lecture Notes in Math.*, pages 51–62. Springer, Berlin-New York, 1978.

[42] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47) :5537–5552, 2002.

[43] P. Hansbo. Nitsche's method for interface problems in computational mechanics. *Gamm-mitteilungen*, 28 :183–206, 2005.

[44] P. Hansbo, M. Larson, and K. Larsson. Cut finite element methods for linear elasticity problems. In *Geometrically unfitted finite element methods and applications*, volume 121 of *Lect. Notes Comput. Sci. Eng.*, pages 25–63. Springer, Cham, 2017.

[45] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825) :357–362, September 2020.

[46] J. Haslinger and Y. Renard. A new fictitious domain approach inspired by the extended finite element method. *SIAM Journal on Numerical Analysis*, 47(2) :1474–1499, 2009.

[47] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8) :1735–1780, nov 1997.

[48] G. A. Holzapfel. Nonlinear solid mechanics : a continuum approach for engineering science, 2002.

[49] H. Johansen and P. Colella. A cartesian grid embedded boundary method for Poisson's equation on irregular domains. *Journal of Computational Physics*, 147(1) :60–85, 1998.

[50] B. S. Jovanović and E. Süli. *Analysis of finite difference schemes : for linear partial differential equations with generalized solutions*, volume 46. Springer Science & Business Media, 2013.

[51] D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[52] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Neural operator : Learning maps between function spaces. *CoRR*, abs/2108.08481, 2021.

[53] C. Lehrenfeld and A. Reusken. Analysis of a high-order unfitted finite element method for elliptic interface problems. *IMA J. Numer. Anal.*, 38(3) :1351–1387, 2018.

[54] K. Li, N. Atallah, G. Main, and G. Scovazzi. The shifted interface method : A flexible approach to embedded interface computations. *International Journal for Numerical Methods in Engineering*, 121(3) :492–518, 2020.

[55] Z. Li. An overview of the immersed interface method and its applications. *Taiwanese Journal of Mathematics*, 7(1) :1 – 49, 2003.

[56] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries, 2022.

[57] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator : Graph kernel network for partial differential equations. *arXiv preprint arXiv :2003.03485*, 2020.

[58] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations, ICLR 2021.

[59] Z. Li, T. Lin, and X.-H. Wu. New cartesian grid methods for interface problems using the finite element formulation. *Numerische Mathematik*, 96 :61–98, 2003.

[60] A. Logg and G. N. Wells. Dolfin : Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37(2) :1–28, 2010.

[61] L. Lu, P. Jin, and G. E. Karniadakis. DeepONet : Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv :1910.03193*, 2019.

[62] D. Maclaurin, D. Duvenaud, and R. P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2113–2122. JMLR.org, 2015.

[63] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I : Poisson and Stokes problems. *J. Comput. Phys.*, 372 :972–995, 2018.

[64] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2) :239–245, 1979.

[65] A. Mendizabal, P. Márquez-Neila, and S. Cotin. Simulation of hyperelastic materials in real-time using deep learning. *Medical Image Analysis*, 59 :101569, 2019.

[66] L. Meunier, G. Chagnon, D. Favier, L. Orgéas, and P. Vacher. Mechanical experimental characterisation and numerical modelling of an unfilled silicone rubber. *Polymer testing*, 27(6) :765–777, 2008.

[67] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37 :239–261, 2005.

[68] Mmg Platform. Mmg - open source software for mesh generation and adaptation, 2025.

[69] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1) :131–150, 1999.

[70] L. Monigatti. A visual guide to learning rate schedulers in PyTorch. `https://www.leoniemonigatti.com/blog/pytorch-learning-rate-schedulers.html`, Dec 2022.

[71] M. Nastorg, M.-A. Bucci, T. Faney, J.-M. Gratien, G. Charpiat, and M. Schoenauer. An implicit gnn solver for poisson-like problems. *Computers & Mathematics with Applications*, 176 :270–288, 2024.

[72] A. Odot, R. Haferssas, and S. Cotin. Deepphysics : A physics aware deep learning framework for real-time simulation. 2022.

[73] C. Olah. Understanding LSTM networks, 2015.

[74] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf : Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[75] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch : An imperative style, high-performance deep learning library, 2019.

[76] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378 :686–707, February 2019.

[77] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015 : 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[78] A. A. Ruggiu, P. Weinerfelt, and J. Nordström. A new multigrid formulation for high order finite difference methods on summation-by-parts form. *Journal of Computational Physics*, 359 :216–238, 2018.

[79] M. W. Scroggs, I. A. Baratta, C. N. Richardson, and G. N. Wells. Basix : a runtime finite element basis evaluation library. *Journal of Open Source Software*, 7(73) :3982, 2022.

[80] M. W. Scroggs, J. S. Dokken, C. N. Richardson, and G. N. Wells. Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes. *ACM Transactions on Mathematical Software*, 48(2) :18 :1–18 :23, 2022.

[81] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4) :1591–1595, 1996.

[82] G. H. Shortley and R. Weller. The numerical solution of Laplace's equation. *Journal of Applied Physics*, 9(5) :334–348, 1938.

[83] J. Sirignano and K. Spiliopoulos. Dgm : A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375 :1339–1364, 2018.

[84] S. Smeets, N. Renaud, and L. J. C. Van Willenswaard. Nanomesh : A python workflow tool for generating meshes from image data. *Journal of Open Source Software*, 7(78) :4654, 2022.

[85] V. Thomée. *Galerkin finite element methods for parabolic problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1997.

[86] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image : image processing in Python. *PeerJ*, 2 :e453, 6 2014.

[87] T. van Erven and W.M. Koolen. MetaGrad : Multiple learning rates in online learning. In *Conference on Neural Information Processing Systems (NIPS)*, pages 1 – 9, Barcelona, Spain, 2016.

[88] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0 : Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17 :261–272, 2020.

[89] L. Weynans. Convergence of a cartesian method for elliptic problems with immersed interfaces. *INRIA research report 8872*, 2017.

[90] Y. Xiao, F. Zhai, L. Zhang, and W. Zheng. High-order finite element methods for interface problems : Theory and implementations. In S. Sherwin, D. Moxey, J. Peiró, P. Vincent, and C. Schwab, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, pages 167–177, 2020.

[91] G. Yoon and C. Min. A review of the supra-convergences of Shortley-Weller method for Poisson equation. *Journal of the Korean Society for Industrial and Applied Mathematics*, 18 :51–60, 2014.

[92] X. Yuanming, X. Jinchao, and W. Fei. High-order extended finite element methods for solving interface problems. *Computer Methods in Applied Mechanics and Engineering*, 364 :112964, 2020.

Abstract

φ-FEM is a new finite element method, proposed to solve partial differential equations on complex domains, using simple non-conforming meshes. The method relies on the use of a level-set function φ, which defines the domain and its boundary. In this manuscript, we recall the method in the simple case of the resolution of the Poisson equation with Dirichlet boundary conditions. We further propose a new way to treat this problem with a penalized version of the method, which provides optimal convergence. Then, we extend our study beyond this simple case, and we propose a numerically optimal φ-FEM scheme to solve the Poisson equation with mixed Dirichlet/Neumann boundary conditions. We also propose different schemes to treat the Heat equation, linear elasticity equation and hyperelastic problems. The rest of the manuscript is devoted to the presentation of different evolutions of φ-FEM. We first propose a Finite Difference scheme based on the φ-FEM paradigm. To provide a real-time method, we then explore the combination of φ-FEM with Neural Operators, where we propose φ-FEM-FNO, which is capable of predicting precise results much faster than FEM based methods. Finally, in the idea of providing opening evolutions, we propose two combinations with the multigrid approach : one based only on φ-FEM, the second on φ-FEM-FNO. The proposed results open many interesting perspectives and challenges for φ-FEM.

Résumé

φ-FEM est une nouvelle méthode éléments finis, proposée pour résoudre des équations aux dérivées partielles sur des domaines complexes, en utilisant des maillages simples non-conformes. La méthode repose sur l'utilisation d'une fonction level-set φ, décrivant le domaine et sa frontière. Dans ce manuscrit, nous rappelons d'abord la méthode appliquée à la résolution du problème de Poisson avec conditions de Dirichlet. Nous proposons ensuite une nouvelle façon de traiter ce problème avec une version pénalisée de la méthode, offrant une convergence optimale. Par la suite, nous étendons notre étude à des problèmes complexes et nous proposons en particulier un schéma φ-FEM numériquement optimal pour résoudre le problème de Poisson avec conditions mixtes Dirichlet/Neumann. Nous proposons également différents schémas φ-FEM permettant de résoudre l'équation de la chaleur ou des problèmes d'élasticité linéaire et non linéaire. La suite du manuscrit est dédiée à la présentation de plusieurs évolutions de φ-FEM. Dans un premier temps, nous proposons un schéma aux différences finies basé sur l'approche φ-FEM. Pour obtenir une méthode temps réel, nous explorons ensuite la combinaison de φ-FEM avec les opérateurs neuraux, où nous proposons φ-FEM-FNO, capable de prédire des résultats précis beaucoup plus rapidement que les méthodes éléments finis. Finalement, dans l'idée de proposer diverses pistes d'évolutions, nous proposons deux combinaisons avec l'approche multigrid : l'une basée uniquement sur φ-FEM, l'autre sur φ-FEM-FNO. Les résultats proposés ouvrent alors de nombreux challenges et perspectives pour φ-FEM.