Simulation of two-fluid immiscible Stokes flows using hybrid nonconforming methods and geometrically unfitted meshes

Stefano Piccardo

Doctoral Degree in Applied Mathematics
Ecole des Ponts ParisTech
Universitat Politècnica de Catalunya

Simulation of two-fluid immiscible Stokes flows using hybrid nonconforming methods and geometrically unfitted meshes

by

Stefano Piccardo

CERMICS
Centre d'Enseignement et de Recherche en MathématIques et Calcul Scientifique
LaCàN
Laboratori de Càlcul Numèric

Advisors: Alexandre Ern, Antonio Huerta, Matteo Giacomini

Barcelona, December 4, 2023

‘‘Nothing in life is to be feared,
it is only to be understood.
Now is the time to understand more,
so that we may fear less.’’
–Maria Skłodowska Curie

Abstract

We consider two-fluid problems involving immiscible Stokes fluids in contact and with surface tension at their interface. We develop, analyze, and compare two approaches for space discretization, the Hybrid High-Order (HHO) and the Hybridizable Discontinuous Galerkin (HDG) methods, both combined with a geometrically unfitted approach to handle fluid interfaces. Both methods do not place discrete unknowns on the interface, and for both methods, we observe optimal error decay and condition number growth. The first part of the Thesis deals with the unfitted HHO solver, where the interface is described by a level-set function and discretized using isoparametric finite elements. We use the methodology to study the equilibrium problem with a shear flow imposed at infinity, and we investigate the relationship between the eccentricity of the ellipse-shaped interface at equilibrium and the ratio of shear to surface tension force. Then, we explore settings where the shape of the interface is unknown. We devise a fixed-point iterative procedure that alternates a transport step for the level-set function with the unfitted HHO solver with frozen interface. In the second part of the Thesis, the unfitted HDG solver is combined with a NURBS description of the interface and of the external boundary, enabling a seamless transition from CAD geometries. We use this solver to study one- and two-fluid Stokes problems with applications in microfluidics related to microfluidic mixers and to emulsion flows in porous media. Both applications employ a polynomial-adaptivity error estimator, delivering results with at least two significant digits. Finally, in the third part of the Thesis, the HDG and HHO methods are compared for one- and two-fluid Stokes problems. In the simplest setting, we prove that the HHO and HDG methods can differ only in the choice of the approximation spaces and of the stabilization operator.

Keywords: Hybrid discretization methods, Unfitted meshes, Stokes flows, Immiscible incompressible fluids, Surface tension, Level-set methods, NURBS

Résumé

On considère des écoulements de deux fluides de Stokes immiscibles séparés par une interface avec tension surfacique. On développe, analyse et compare deux approches pour la discrétisation spatiale, la méthode hybride d'ordre élevé (HHO) et celle de Galerkin discontinue hybridisable (HDG), toutes deux combinées avec des maillages immergés ne tenant pas compte de la position de l'interface. Les deux méthodes ne placent pas d'inconnue discrète sur l'interface, et on observe une décroissance optimale de l'erreur et une croissance optimale du conditionnement du système linéaire. La première partie de la thèse traite du solveur HHO avec une interface décrite par une fonction de niveau et approchée par des éléments finis isoparamétriques. On utilise la méthode pour étudier le problème d'équilibre avec un cisaillement imposé à l'infini, en particulier la dépendance de l'eccentricité de l'interface (qui est de forme elliptique) avec le rapport entre forces de cisaillement et tension de surface. On explore ensuite des situations où la forme de l'interface est inconnue. On met au point une méthode itérative de point fixe qui alterne entre une étape de transport de la fonction de niveau pour l'interface et le solveur de Stokes sur maillage immergé. Dans la deuxième partie de la thèse, le solveur HDG est combiné avec une description de l'interface et de la frontière extérieure par des splines (NURBS), ce qui simplifie grandement le traitement de données de type CAD. Le solveur HDG est utilisé pour des écoulements à un et deux fluides dans des applications en micro-fluidique, comme l'écoulement dans un micro-mélangeur et l'émulsions en milieu poreux. Dans les deux cas, on utilise un estimateur d'erreur permettant d'adapter le degré polynomial, et d'obtenir une précision à deux chiffres significatifs. Dans la troisième partie de la thèse, les méthodes HHO et HDG sont comparées pour des problèmes de Stokes à un et deux fluides. Dans les cas les plus simples, on montre que HHO et HDG peuvent seulement différer dans le choix des espaces discrets locaux et la forme de la stabilisation.

Mots-clés: Méthodes de discrétisation hybrides, Maillages immergés, Écoulements de Stokes, Fluides immiscibles et incompressibles, Tension surfacique, Fonction de niveau, NURBS

Resumen

Consideramos problemas de dos fluidos de Stokes inmiscibles separados por una interfaz con tensión superficial. Desarrollamos, analizamos y comparamos dos enfoques para la discretización espacial, el método híbrido de alto orden (HHO) y el método Galerkin discontinuo hybridizable (HDG), ambos combinados con mallas inmersas que no tienen en cuenta la posición de la interfaz. Ambos métodos no colocan incógnitas discretas en la interfaz y observamos una disminución óptima del error y un crecimiento óptimo de la condición del sistema lineal. La primera parte de la tesis se centra en el solucionador HHO, con una interfaz descrita por una función de nivel y aproximada mediante elementos finitos isoparamétricos. Utilizamos este método para estudiar el problema del equilibrio con un cizallamiento impuesto en el infinito, especialmente la dependencia de la excentricidad de la interfaz (que tiene forma elíptica) con la relación entre las fuerzas de cizallamiento y la tensión superficial. Luego, exploramos situaciones en las que la forma de la interfaz es desconocida. Desarrollamos un procedimiento iterativo de punto fijo que alterna entre un paso de transporte de la función de nivel para la interfaz y el solucionador de Stokes en una malla inmersas. En la segunda parte de la tesis, el solucionador HDG se combina con una descripción de la interfaz y el límite exterior mediante splines (NURBS), lo que simplifica el manejo de datos de tipo CAD. Utilizamos este solucionador HDG para resolver problemas de flujo de uno y dos fluidos en aplicaciones de microfluidos, como el flujo en un mezclador microfluídico y emulsiones en medios porosos. En ambos casos, empleamos un estimador de error que permite adaptar el grado polinómico y obtener resultados con al menos dos dígitos significativos. En la tercera parte de la tesis, comparamos los métodos HHO y HDG para problemas de Stokes de uno y dos fluidos. En los casos más simples, demostramos que HHO y HDG solo pueden diferir en la elección de los espacios discretos locales y la forma de la estabilización.

Palabras clave: Métodos de discretización híbrida, Mallas inmersas, Flujos de Stokes, Fluidos inmiscibles e incompresibles, Tensión superficial, Función de nivel, NURBS

Acknowledgments

I would like to express my gratitude to my esteemed supervisors, Alexandre Ern, Matteo Giacomini, and Antonio Huerta, for entrusting me with this research project and their guidance, support, and encouragement throughout my doctoral journey.

Alexandre, I want to express my heartfelt appreciation for your exceptional availability and patience. You have introduced me to the world of research, instilled in me the core principles of the scientific method, and driven me to strive for excellence consistently. I am immensely grateful for your invaluable assistance, particularly during the final months; this achievement would not have been possible without you.

Matteo and Antonio, thank you very much for your mentorship. You have empowered me to become a more independent researcher and have instilled in me the importance of clear communication and idea-sharing. I am profoundly indebted to each of you for your invaluable insights and constructive feedback. Completing this thesis would not have been possible without your invaluable help and support.

I would also like to extend my gratitude to other mentors who have contributed significantly to my academic journey. Guillaume Delay for introducing me to HHO, Omar Duran Triana and Zhaonan Dong for engaging in exciting discussions and helpful hints, Matteo Cicuttin for the assistance with cluster, Sonia Fernández-Méndez for the precious advice at the MFET congress, Isabelle Simunic for all the administrative help, and, especially, Derya Gok and Géraldine Pichot for their friendship, assistance, and empathy. I extend my sincere appreciation to Simona Perotto for her profound influence on my academic growth and development.

My heartfelt thanks go to all my friends and colleagues with whom I have shared my path: Davide, Rafel, Pau, Mariano, Luan, Alvaro, Nadeem, Hector, Stephan, Shardool, Sergi, Monica, Fabiola, Christina, Jean, Robert, Frédéric, Ari, and Gaspard. Each of you has been very important to me in this experience.

My heartfelt thanks go to my family and all my friends for their constant support throughout this journey. They have been by my side every step of the way, offering invaluable encouragement. Special thanks to my mom and dad for their wise advice during the most challenging moments of the thesis. To my grandparents, for never forgetting me and consistently remembering to reach out and call. To Matteo, who, at times, seemed to reside more in Barcelona than in Milano. To Carola and Mattia, among many other things, for nourishing me during the darkest hours as the submission deadline was approaching. To Filippo, for all the artistic advice in creating cool images. To all my friends in Genova and Milano, who make me feel at home every time I return. To my friends in Paris and Barcelona, for making all this possible by accompanying me on this journey. It would be unfair to list some names, as I would surely forget others, and one page won't be enough to say all that you mean to me. I hold you all dear, and thank you for your unwavering support and friendship.

Lastly, I wish to express my gratitude to the funding bodies that have supported my research, including CERMICS and LaCàN labs. I am especially grateful to Antonio for his discretionary funding, which played a crucial role in enabling me to carry out my research and complete this thesis.

Table of Contents

Abstract **i**

Résumé **iii**

Resumen **v**

Acknowledgments **vii**

Table of Contents **ix**

List of Figures **xiii**

List of Tables **xvii**

Résumé étendu **xix**
0.1 Motivations . . . xix
0.2 Écoulements de Stokes avec deux fluides incompressibles et immiscibles . . . xx
0.2.1 Équations du modèle . . . xxi
0.2.2 Description de l'interface et de la frontière extérieure . . . xxii
0.2.2.1 Description par fonction de niveau . . . xxii
0.2.2.2 Interface et frontière extérieure en CAO . . . xxii
0.3 Méthodes de discretisation hybrides pour les problèmes elliptiques . . . xxiii
0.4 Maillages immergés . . . xxv
0.5 Aperçu de la thèse et objectifs . . . xxvii

1 Introduction **1**
1.1 Motivations . . . 1
1.2 Stokes interface problem . . . 2
1.2.1 Governing equations . . . 3
1.2.2 Interface and boundary description . . . 4
1.2.2.1 Level-set description . . . 4
1.2.2.2 Unknown interface: level-set transport problem . . . 4
1.2.2.3 CAD interface and boundary . . . 5
1.3 Hybrid discretization methods for elliptic problems . . . 5
1.4 Unfitted meshes . . . 8
1.4.1 Literature review . . . 8
1.4.2 Geometric discretization . . . 9
1.4.2.1 C^0-piecewise polynomial representation . . . 10
1.4.2.2 NURBS representation . . . 10
1.5 Overview and objectives . . . 11

I Hybrid High-Order methods **13**

2 Unfitted HHO solver **15**
2.1 Introduction . . . 15
2.2 Model problem . . . 17
2.3 Unfitted HHO solver with fixed interface . . . 19
2.3.1 Unfitted meshes . . . 19
2.3.2 HHO discretization . . . 20
2.3.3 Interface discretization and quadratures in the cut cells . . . 23

3 HHO for immiscible Stokes fluids **27**
3.1 Equilibrium with a pure shear flow . . . 27
3.2 Fixed-point solver for unknown interface problems . . . 32
3.2.1 Fixed-point scheme . . . 32
3.2.2 Normal and curvature for unknown interface . . . 33
3.3 Numerical results for unknown interface problems . . . 35
3.3.1 Verification test cases . . . 35
3.3.2 Test cases with unknown interface . . . 41
3.3.3 Conclusions . . . 43

II Hybridizable Discontinuous Galerkin methods **45**

4 Unfitted HDG solver **47**
4.1 Introduction . . . 47
4.2 Problem statement . . . 48
4.3 Computational setting . . . 49
4.3.1 Local problems . . . 52
4.3.1.1 Immersed boundary elements . . . 53
4.3.1.2 Interface elements . . . 53
4.3.2 Global problem . . . 55
4.4 Domain definition and quadratures in the interface elements . . . 56
4.5 Computational details for high-fidelity simulations . . . 58
4.5.1 Degree adaptivity . . . 58
4.5.2 Element extension . . . 59

5 HDG for microfluid dynamics problems **61**
5.1 Numerical validation . . . 61
5.1.1 Badly cut faces . . . 62
5.1.2 Badly cut cells . . . 65
5.2 Taylor–Couette with immersed boundary . . . 66
5.3 Circular bubble at equilibrium . . . 72
5.4 Passive microfluidic mixer with immersed boundary . . . 73
5.5 Emulsion flows in a porous medium . . . 76
5.6 Conclusions . . . 80

Appendices **81**
5.A Quadrature triangulation . . . 81
5.B Domain definition . . . 82

III Bridging HHO and HDG **87**

6 Bridging HHO and HDG for one- and two-fluid Stokes problems **89**
6.1 One-fluid Stokes problem . . . 89
6.1.1 Model problem . . . 89
6.1.2 Discrete setting . . . 90
6.1.3 HDG formulation . . . 93
6.1.3.1 Global formulation . . . 93
6.1.3.2 Static condensation . . . 94
6.1.4 HHO formulation . . . 95
6.1.4.1 Global formulation . . . 95
6.1.4.2 Static condensation . . . 96
6.1.5 Bridging HHO and HDG . . . 97
6.1.6 More general boundary conditions . . . 99
6.2 Two-fluid Stokes problem . . . 100
6.2.1 Model problem . . . 101
6.2.2 Discrete setting . . . 101
6.2.3 HDG formulation . . . 103
6.2.3.1 Global formulation . . . 103
6.2.3.2 Static condensation . . . 104
6.2.4 HHO formulation . . . 105
6.2.4.1 Global formulation . . . 105
6.2.4.2 Static condensation . . . 107
6.2.5 Bridging HHO and HDG . . . 107

7 Conclusions and perspectives **111**

References **115**

List of Figures

1 Instantané de trois micro-gouttelettes, se déplaçant de gauche à droite, contenant un colorant de couleur sombre (image inspirée de [60]) xix
2 Domaine de calcul Ω, sous-domaines Ω_i, $i \in \{1,2\}$, interface Υ (en bleu), frontière extérieure de Dirichlet $\partial\Omega^{\mathrm{D}}$ (en orange), frontière extérieure de Neumann $\partial\Omega^{\mathrm{N}}$ (en rouge), normale unitaire $\boldsymbol{n}_\Upsilon$ orientant Υ, et normale unitaire sortante à Ω, $\boldsymbol{n}_\Omega$. xx
3 Inconnues de vitesse pour les méthodes cG-FEM (à gauche), dG-FEM (au centre) et dG-FEM hybrides (à droite) en utilisant des approximations polynomiales du second ordre pour les inconnues des maille et de face. xxiv
4 Inconnues de vitesse pour les méthodes de Galerkin discontinu hybrides utilisant des approximations polynomiales d'ordre zéro dans les mailles et les faces. Rangée supérieure: problème d'interface; rangée inférieure: problème de avec frontière courbe; colonne de gauche: avant agglomération; colonne de droite: après agglomération. xxvii

1.1 Snapshot of three microdroplets, moving from left to right, containing dark-colored dye (image inspired by [60]). 1
1.2 Computational domain Ω, subdomains Ω_i, $i \in \{1,2\}$, interface Υ (blue), Dirichlet boundary $\partial\Omega^{\mathrm{D}}$ (orange), Neumann boundary $\partial\Omega^{\mathrm{N}}$ (red), unit normal $\boldsymbol{n}_\Upsilon$ orienting Υ, and outward unit normal to Ω, $\boldsymbol{n}_\Omega$. 2
1.3 Discrete velocity unknowns for cG-FEM (left), dG-FEM (center) and hybridized dG-FEM (right) using second-order polynomial approximations for both the cell and the hybrid unknowns. 6
1.4 Discrete velocity unknowns for hybridized discontinuous Galerkin methods using zero-order polynomial approximations for the cell and the hybrid unknowns. Top row: interface problem; bottom row: curved boundary problem; left column: before agglomeration; right column: after agglomeration. 9

2.1 Computational domain Ω, subdomains Ω_i, $i \in \{1,2\}$, interface Γ, and unit normal $\boldsymbol{n}_\Gamma$. 18
2.2 Decomposition of a cut cell $T \in \mathcal{T}^\Gamma$ and of its boundary ∂T. 20
2.3 Discrete velocity unknowns ($k = 0$) for a cut (left) and uncut (right) cell of hexagonal shape. Notice the doubling of the cell unknowns in the cut cell and of the face unknowns on the cut faces. 21
2.4 Example of interface discretization. 25

2.5 Example of cut cells $T \in \mathcal{T}^\Gamma$ and of their decomposition into curved subtriangles. 25

3.1 Streamlines of the velocity field $\boldsymbol{u}$ obtained by solving (2.5) with elliptic interface having radii $R_1 = 1/6$, $R_2 = 1/3$, and parameters $\varepsilon = 0.59$, $\gamma = 1$, $\mu_1 = \mu_2 = 1$ in the domain $\Omega = (-2, 2)^2$. 28
3.2 Figure 3.2a: normal velocities along the interface. Figure 3.2b: $\boldsymbol{u}(0, \gamma = 1) \cdot \boldsymbol{n}_\Gamma$ versus $\boldsymbol{u}(\varepsilon = 1, 0) \cdot \boldsymbol{n}_\Gamma$ (blue bullets). Elliptic interface with $D = 1/3$ and $\mu_1 = \mu_2 = 1$. 29
3.3 Normal velocity (blue, top row) and curvature (red, bottom row) along the interface. Shear-surface tension equilibrium with elliptic interface for $D = 1/3$, $\mu_1 = \mu_2 = 1$, $Ca \in \{0.308, 0.274, 0.278\}$ using meshes composed of 32×32, 64×64 and 128×128 square cells, respectively. 30
3.4 Shear-surface tension equilibrium with elliptic interface: isovalues of the velocity components $\boldsymbol{u} = (u_1, u_2)$ for $D = 1/3$, $\mu_1 = \mu_2 = 1$, $Ca = 0.28$ in the domain $\Omega = (-2, 2)^2$. 30
3.5 Figures 3.5a and 3.5b display the ratio D/Ca (Taylor deformation parameter over capillary number) as a function of the viscosity ratio λ. Figure 3.5c displays D as a function of Ca. 31
3.6 First verification test case. In each panel, the red bullets represent the initial interface, and the blue ones the interface at iteration m of the fixed-point procedure. 36
3.7 First verification test case: error indicators. 37
3.8 First verification test case. Shifted curvature $H_\Gamma^{m,*} - H_*$ (see (3.23)) as a function of the curvilinear abscissa along the interface at iteration $m \in \{40, 80, 100, 140, 200\}$ of the fixed-point procedure. 38
3.9 Second verification test case: $Ca = 0.27$. In each panel, the red bullets represent the initial interface, and the blue ones the interface at iteration m of the fixed-point procedure. 39
3.10 Second verification test case: error indicators (inner area, normal velocity at the interface, and Taylor deformation parameter) for $Ca \in \{0.007, 0.07, 0.27\}$. 40
3.11 Second verification test case: error indicators for $Ca = 0.007$. 40
3.12 Streamlines for the test case with unknown interface. 41
3.13 Interface evolution for the test case with unknown interface with $Ca = 0.27$. . . . 42
3.14 Error indicators for $Ca = 0.27$ for various levels of mesh refinement. 43

4.1 Domain Ω_o partitioned into 6×6 square elements. The external boundary $\partial\Omega$ and the interface Υ are not aligned with the mesh skeleton. 51
4.2 Element- and face-based unknowns used for the different types of cut elements. . 52
4.3 Example of a square element Ω_e cut by the interface Υ into two regions. 57
4.4 Illustration of the element extension strategy for an element badly cut by the external boundary. 60

5.1 M-shaped domain with $\varepsilon = 0.15$ ($\beta = 60\%$ of the whole face). 63
5.2 Example of nodal (Lagrange) and modal (Legendre) face basis functions in the reference face $\hat{\Gamma} = [-1, 1]$. 63
5.3 M-shaped domain with $\varepsilon \in \{0.15, 0.1, 0.05, 0.005\}$ (i.e., $\beta \in \{0.60, 0.40, 0.20, 0.02\}$). Comparison of $\log_{10}(\kappa^{\text{glob}})$ with respect to β using Legendre $\hat{\mathcal{P}}^k_{\text{Le}}$ (continuous lines) and Lagrange $\hat{\mathcal{P}}^k_{\text{La}}$ (dotted lines) face basis functions. 64

5.4 Smoothed square domain. 65
5.5 Smoothed square domain. Comparison on the local condition number $\log_{10}(\kappa^{\text{loc}})$ (top row) and of the element L^2-velocity error (bottom row) without (left) and with (right) element extension. 66
5.6 Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Example of the element extension strategy applied to a mesh composed of 8×8 square elements. 67
5.7 Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Absolute value of the approximated velocity for $k = 4$. 68
5.8 Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Absolute value of the velocity and pressure errors for $k = 4$. 68
5.9 Taylor–Couette domain. The L^2-error of the mixed variable (continuous lines) and pressure (dashed lines) using element extension. 69
5.10 Taylor–Couette domain. In Figure 5.10a, the L^2-velocity error, and in Figure 5.10b, L^2-superconvergent velocity error using element extension. 69
5.11 Taylor–Couette domain. The maximum value of the global and local condition numbers as a function of the polynomial order k. 70
5.12 Taylor–Couette domain. Global condition number as a function of the mesh-size. 70
5.13 Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Degree-adaptive approximation space ensuring 4 significant digits. 71
5.14 Equilibrium bubble. The element extension strategy is applied. 72
5.15 Equilibrium bubble. Absolute value of the error of the approximated velocity and pressure. 73
5.16 Mixer. 74
5.17 Mixer. Approximated velocity and pressure. The element extension strategy is applied. 75
5.18 Mixer. Degree-adaptive approximation space ensuring 2 significant digits. The element extension strategy is applied. 75
5.19 Porous media emulsion. 77
5.20 Porous media emulsion. Approximated velocity and pressure. The element extension strategy is applied. 78
5.21 Porous media emulsion. Degree-adaptive approximation space ensuring 2 significant digits. The element extension strategy is applied. 79
5.22 Triangulation procedure for a region Ω_e^i, $i \in \{1, 2\}$. 82
5.23 Example of computational domain Ω with fitted external boundary. 83
5.24 Example of a square element Ω_e cut by the interface Υ into two regions. 84

List of Tables

1.1 Notations HHO vs. HDG. 12

5.1 Mixer: circular inclusions data (ordered from left to right). 73
5.2 Porous media emulsion: circular bubbles data. 76

0.1 Motivations

De nos jours, les simulations haute-fidélité des problèmes de dynamique des fluides sont de plus en plus fréquemment utilisées, ce qui engendrent une demande croissante en méthodes de calcul précises. Néanmoins, ces simulations posent des défis considérables, notamment dans des problèmes impliquant des géométries complexes ou évolutions comme pour la microfluidique [10], [61], [120], les réservoirs de pétrole [102], les écoulements souterrains [109], les réservoirs de stockage d'énergie [88], et la séquestration géologique du carbone [104]. Un défi particulièrement est celui de la résolution précise de problèmes multi-fluides, où des fluides immiscibles interagissent entre aux, voir figure 1. Dans de telles configurations, l'interface entre les fluides peut être sujette à des déformations, des ruptures et des changements de forme complexes, nécessitant des méthodes numériques robustes capables de capturer avec précision ces phénomènes dynamiques.

Figure 1: Instantané de trois micro-gouttelettes, se déplaçant de gauche à droite, contenant un colorant de couleur sombre (image inspirée de [60])

Cette thèse se concentre sur des problèmes d'écoulements de deux fluides de Stokes immiscibles séparés par une interface avec tension surfacique. Nous ne considérons que des problèmes stationnaires où chaque fluide est régi par les équations de Stokes stationnaires et incompressibles. L'utilisation des équations de Stokes plutôt que des équations de Navier-Stokes est raisonnable en raison des petites échelles de longueur et donc du nombre de Reynolds modéré impliqué dans le problème. Une autre difficulté est que la position de l'interface est fait en général partie des inconnues du problème. Pour simplifier, nous nous concentrons uniquement sur le cas de deux fluides et sur les écoulements bidimensionnels. Il convient de noter que dans la microfluidique, telle que les dispositifs de laboratoire sur puce, les problèmes bidimensionnels ne sont souvent pas très éloignés de la réalité, notamment lorsque l'épaisseur du domaine est très faible.

Cette thèse vise à explorer et à comparer la méthode de Galerkin discontinue hybridisable (HDG) et la méthode hybride d'ordre élevé (HHO), combinées avec des maillages immergés ne tenant pas compte de la position de l'interface. Dans la section 0.2, nous présentons les équations du modèle et décrivons diverses techniques pour représenter l'interface entre les deux fluides ou la frontière extérieure du domaine de calcul. Dans la section 0.3, nous motivons l'utilisation de méthodes de discrétisation hybrides, comme HDG et HHO. Dans la section 0.4, nous motivons l'utilisation de maillages immergés. Finalement, dans la section 0.5, nous exposons le plan de la thèse et ses objectifs.

0.2 Écoulements de Stokes avec deux fluides incompressibles et immiscibles

Considérons deux fluides tels qu'une goutte d'un fluide, indexé par 1, soit immergée dans un fluide externe, indexé par 2. Soit $\Omega \subset \mathbb{R}^d$, $d := 2$, le domaine de calcul occupé par les deux fluides, de telle sorte que chaque fluide i réside dans Ω_i (ouvert, borné, connexe, de frontière lipschitzienne), avec $\overline{\Omega} := \overline{\Omega}_1 \cup \overline{\Omega}_2$. La variété $\Upsilon := \partial\Omega_1 \cap \partial\Omega_2$ représente l'interface entre les deux fluides. Pour simplifier, nous supposons que Υ ne touche pas la frontière $\partial\Omega$ de Ω, de sorte que l'index 1 fait référence au sous-domaine intérieur Ω_1, avec $\partial\Omega_1 = \Upsilon$, et l'index 2 fait référence au sous-domaine extérieur Ω_2, avec $\partial\Omega_2 = \partial\Omega \cup \Upsilon$, voir la figure 2. Nous supposons que l'interface Υ est orientable, et que sa normale unitaire $\boldsymbol{n}_\Upsilon$ et sa courbure H_Υ peuvent être définies partout sur Υ. Par convention, $\boldsymbol{n}_\Upsilon$ pointe de Ω_1 vers Ω_2, c'est-à-dire que Υ est orientée dans le sens antihoraire. Sur la frontière extérieure $\partial\Omega$, des conditions de

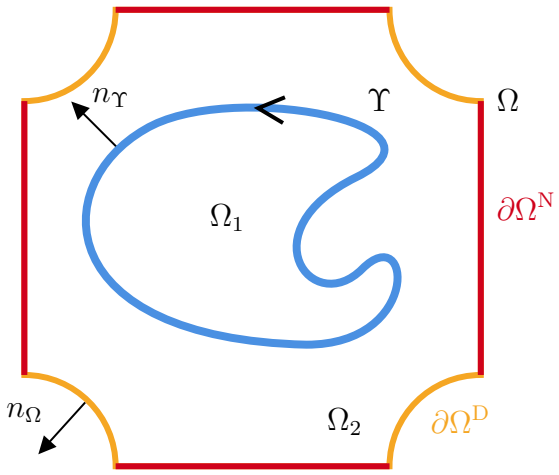


Figure 2: Domaine de calcul Ω, sous-domaines Ω_i, $i \in \{1, 2\}$, interface Υ (en bleu), frontière extérieure de Dirichlet $\partial\Omega^{\mathrm{D}}$ (en orange), frontière extérieure de Neumann $\partial\Omega^{\mathrm{N}}$ (en rouge), normale unitaire $\boldsymbol{n}_\Upsilon$ orientant Υ, et normale unitaire sortante à Ω, $\boldsymbol{n}_\Omega$.

Dirichlet et de Neumann sont prescrites sur $\partial\Omega^{\mathrm{D}}$ et $\partial\Omega^{\mathrm{N}}$ respectivement, avec $\overline{\partial\Omega^{\mathrm{D}}} \cup \overline{\partial\Omega^{\mathrm{N}}} = \partial\Omega$ et $\partial\Omega^{\mathrm{D}} \cap \partial\Omega^{\mathrm{N}} = \emptyset$. Nous imposons la vitesse $\boldsymbol{g} \in H^{1/2}(\partial\Omega^{\mathrm{D}}; \mathbb{R}^d)$ sur la frontière de Dirichlet, et la traction $\boldsymbol{t} \in L^2(\partial\Omega^{\mathrm{N}}; \mathbb{R}^d)$ sur la frontière de Neumann. De plus, nous imposons la force volumique $\boldsymbol{f}_i \in L^2(\Omega_i; \mathbb{R}^d)$ dans le sous-domaine Ω_i, $i \in \{1, 2\}$.

0.2.1 Équations du modèle

Nous recherchons les champs de vitesse et de pression $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i; \mathbb{R}^d) \times L^2(\Omega_i)$, $i \in \{1, 2\}$, tels que

$$-\nabla \cdot \boldsymbol{\sigma}_i = \boldsymbol{f}_i \qquad \text{dans } \Omega_i, \quad i \in \{1,2\}, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u}_i = 0 \qquad \text{dans } \Omega_i, \quad i \in \{1,2\}, \tag{1b}$$

$$\boldsymbol{u}_2 = \boldsymbol{g} \qquad \text{sur } \partial\Omega^{\mathrm{D}} \tag{1c}$$

$$\boldsymbol{\sigma}_2 \boldsymbol{n}_\Omega = \boldsymbol{t} \qquad \text{sur } \partial\Omega^{\mathrm{N}}, \tag{1d}$$

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}]\!]\boldsymbol{n}_\Upsilon = \gamma H_\Upsilon \boldsymbol{n}_\Upsilon \qquad \text{sur } \Upsilon, \tag{1e}$$

où le tenseur des contraintes de Cauchy pour chaque fluide est

$$\boldsymbol{\sigma}_i := 2\mu_i \nabla^{\mathrm{s}} \boldsymbol{u}_i - p_i \boldsymbol{I}_d, \quad \nabla^{\mathrm{s}} \boldsymbol{u}_i := \frac{1}{2}(\nabla \boldsymbol{u}_i + \nabla \boldsymbol{u}_i^{\mathrm{T}}), \tag{2}$$

μ_i est la viscosité du fluide i, $\boldsymbol{n}_\Omega$ est la normale extérieure unitaire à Ω, et $\boldsymbol{I}_d$ est le tenseur identité. De plus, pour une grandeur générique $\odot$ définie sur $\Omega_1 \cup \Omega_2$, pouvant être scalaire, vectorielle ou tensorielle, nous posons

$$[\![\odot]\!] := (\odot|_{\Omega_1})|_\Upsilon - (\odot|_{\Omega_2})|_\Upsilon \tag{3}$$

pour désigner le saut composante par composante à travers Υ. Les conditions (1e) imposent la continuité de la vitesse et l'équilibre des forces à travers l'interface en tenant compte de la tension superficielle, qui est modélisée selon la loi de Laplace, avec γ la tension superficielle. On notera que la composante tangentielle de la contrainte normale est toujours continue à travers Υ.

Une alternative à la formulation de Cauchy pour le tenseur des contraintes est la *formulation en gradient complet*. Dans ce cas, nous recherchons les champs de vitesse et de pression $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i; \mathbb{R}^d) \times L^2(\Omega_i)$, $i \in \{1, 2\}$, résolvant (1), mais avec (2) remplacée par

$$\boldsymbol{\sigma}_i' := \mu_i \nabla \boldsymbol{u}_i - p_i \boldsymbol{I}_d. \tag{4}$$

Bien que $\boldsymbol{\sigma}_i$ et $\boldsymbol{\sigma}_i'$ aient la même divergence à l'intérieur de Ω_i en raison de (1b), l'utilisation de $\boldsymbol{\sigma}_i'$ entraîne des conditions différentes à l'interface et de Neumann, car (1d) et (1e) deviennent maintenant

$$\boldsymbol{\sigma}_2' \boldsymbol{n}_\Omega = \big(\mu_2 \nabla \boldsymbol{u}_2 - p_2 \boldsymbol{I}_d\big)\boldsymbol{n}_\Omega = \boldsymbol{t} \qquad \text{sur } \partial\Omega^{\mathrm{N}}, \tag{5a}$$

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}']\!]\boldsymbol{n}_\Upsilon = [\![\mu \nabla \boldsymbol{u} - p\boldsymbol{I}_d]\!]\boldsymbol{n}_\Upsilon = \gamma H_\Upsilon \boldsymbol{n}_\Upsilon \quad \text{sur } \Upsilon, \tag{5b}$$

Remark 0.1 (Unicité). Lorsque $\partial\Omega^{\mathrm{N}} = \emptyset$, le problème ci-dessus est soluble à une constante additive globale près sur la pression, qui peut être fixée en imposant une condition supplémentaire sur la pression. Par exemple, nous pouvons imposer

$$\sum_{i \in \{1,2\}} \int_{\Omega_i} p_i = 0. \tag{6}$$

Remark 0.2 (Unités). Pour simplifier, nous pouvons supposer que les équations sont exprimées sous forme adimensionnelle. Sinon, les unités sont les suivantes: $[m \cdot s^{-1}]$ pour $\boldsymbol{u}$, $[Pa]$ pour p, $[Pa \cdot s]$ pour μ, $[m^{-1}]$ pour H_Υ et $[Pa \cdot m]$ pour γ.

0.2.2 Description de l'interface et de la frontière extérieure

Dans cette section, nous expliquons brièvement comment l'interface Υ et la frontière extérieure $\partial\Omega$ peuvent être décrites mathématiquement. Il convient de noter que la frontière $\partial\Omega$ est toujours connue a priori, mais ce n'est pas nécessairement le cas pour l'interface Υ.

0.2.2.1 Description par fonction de niveau

Une possibilité pour décrire l'interface est de supposer qu'une fonction de niveau est disponible. Cette fonction peut être connue a priori, comme pour une interface fixe, ou résulter d'une approximation, comme pour une interface inconnue. Une fonction de niveau $\phi : \Omega \to \mathbb{R}$ décrit implicitement une courbe générique $\Upsilon \subset \Omega$ en tant que son ensemble de niveau zéro, c'est-à-dire

$$\Upsilon := \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) = 0\}. \tag{7}$$

En supposant que ϕ est de classe $\mathcal{C}^2$ dans Ω, on peut alors définir un champ de vecteurs unitaires et un champ de courbure dans Ω comme suit:

$$\boldsymbol{n}_\phi = \frac{\nabla\phi}{\|\nabla\phi\|_{\ell^2}}, \qquad H_\phi = -\nabla \cdot \boldsymbol{n}_\phi = -\frac{\Delta\phi}{\|\nabla\phi\|_{\ell^2}} + \frac{1}{\|\nabla\phi\|_{\ell^2}^3} D^2\phi(\nabla\phi, \nabla\phi), \tag{8}$$

où $|\cdot|_{\ell^2}$ désigne la norme euclidienne dans $\mathbb{R}^d$ et $D^2\phi(\cdot,\cdot)$ est la forme quadratique associée à la hessienne de ϕ. Le vecteur normal unitaire et la courbure sur Υ sont alors donnés par

$$\boldsymbol{n}_\Upsilon := \boldsymbol{n}_\phi|_\Upsilon, \qquad H_\Upsilon := H_\phi|_\Upsilon. \tag{9}$$

De plus, la connaissance de la fonction de niveau permet une détermination simple du domaine Ω_i, $i \in \{1, 2\}$, en évaluant le signe de $\phi(\boldsymbol{x})$ pour tout $\boldsymbol{x} \in \Omega$.

0.2.2.2 Interface et frontière extérieure en CAO

Lorsqu'une géométrie CAO est disponible, l'approche naturelle considère des courbes NURBS (Non-uniform rational B-spline) pour décrire précisément l'interface et la frontière extérieure. Pour simplifier, considérons une interface Υ décrite par une courbe NURBS $\boldsymbol{C}(\lambda) : \lambda \to \Upsilon$, au moins de classe C^1, avec $\lambda \in [\lambda_a, \lambda_b]$ comme domaine paramétrique. Le vecteur tangent $\boldsymbol{t}_{\boldsymbol{C}}(\lambda)$ à Υ s'obtient en calculant la dérivée première par rapport à λ, ce qui donne

$$\boldsymbol{t}_\Upsilon(\lambda) = \boldsymbol{t}_{\boldsymbol{C}}(\lambda) = \frac{\mathrm{d}\boldsymbol{C}(\lambda)}{\mathrm{d}\lambda}.$$

Le vecteur normal unitaire et la courbure sur Υ sont alors donnés par

$$\boldsymbol{n}_\Upsilon = \boldsymbol{n}_{\boldsymbol{C}} = \frac{\frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda}}{\left\|\frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda}\right\|_{\ell^2}}, \qquad H_\Upsilon = H_{\boldsymbol{C}} = \left\|\frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda}\right\|_{\ell^2}. \tag{10}$$

Il convient de noter que la connaissance de l'interface NURBS ne permet pas une détermination directe du domaine Ω_i, $i \in \{1, 2\}$. En outre, l'orientation de Υ est implicitement spécifiée par la carte $\boldsymbol{C}(\lambda)$.

0.3 Méthodes de discretisation hybrides pour les problèmes elliptiques

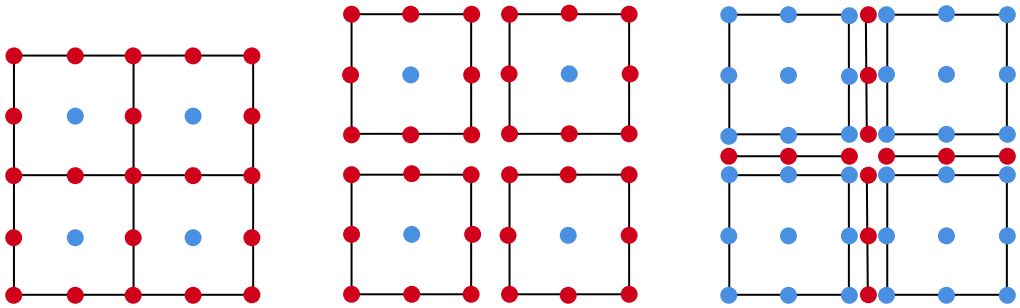
Les méthodes d'ordre élevé ont émergé récemment comme méthode concurrente aux méthodes classiques en termes d'efficacité computationnelle et de qualité des résultats [82], [127].

Les méthodes classiques de discrétisation des équations aux dérivées partielles (EDP), telles que celles rencontrées en dynamique des fluides numérique, sont les méthodes de volume finis (FVM) [56] et des éléments finis de type Galerkin continu (cG-FEM) [17], [51]–[53]. Les méthodes FVM sont obtenues en appliquant un bilan de flux dans chaque maille. Cependant, les méthodes FVM utilisent des approximations constantes par morceaux, ce qui entraîne des limitations dans la représentation précise des variations spatiales. En revanche, les méthodes cG-FEM sont basées sur une formulation variationnelle et approchent la solution à l'aide de polynômes par morceaux qui sont continus aux interfaces entre les mailles. Cette deuxième approche permet des approximations d'ordre élevé, mais elle souffre de problèmes de stabilité pour les écoulements dominés par la convection, et elle nécessite des espaces de discrétisation compatibles (stabilité inf-sup) pour la vitesse et la pression (bien que de nombreuses techniques de stabilisation existent, voir, par exemple, [53, Chap. 62-63]). De plus, les méthodes cG-FEM reposent sur des maillages conformes, ce qui peut être limitant pour des problèmes avec des géométries complexes.

Une approche différente est offerte par les méthodes d'élément finis de type Galerkin discontinu (dG-FEM) [45], [80]. Les méthodes dG-FEM utilisent des polynômes par morceaux qui peuvent entre discontinus entre les mailles. En outre, ces méthodes imposent un bilan de flux aux interfaces du maillage, dans l'esprit des FVM. Les méthodes dG-FEM présentent également l'avantage d'être robustes par rapport à la convection grâce aux flux numériques et offrent une flexibilité dans le choix des discrétisations de la vitesse et de la pression grâce à des stabilisations appropriées. L'inconvénient principal des méthodes dG-FEM réside dans l'augmentation considérable du nombre de degrés de liberté (DDL) sur un maillage donné. De plus, les méthodes dG-FEM, tout comme les cG-FEM, peuvent être sensibles aux maillages fortement déformés. Une approche pour maîtriser les coûts computationnels, tout en préservant l'ordre élevé de l'approximation, est le concept d'hybridation, introduit pour la première fois dans le cadre des méthodes mixtes [6], [57].

Les méthodes de discrétisation hybrides ont gagné en popularité durant cette dernière décennie, comme en témoignent les nombreux minisymposia consacrés à ces techniques lors de conférences internationales, telles qu'ICOSAHOM (Londres, 2018), FEF (Chicago, 2019) et WCCM-ECCOMAS (Paris, 2020). L'idée clé de l'hybridation est d'introduire des inconnues supplémentaires sur le squelette du maillage, c'est-a-dire sur l'ensemble des faces ou arêtes partagées par les mailles. Ces inconnues supplémentaires, appelées inconnues *hybrides*, permettent de réduire les DDL couplés globalement à ceux placés sur le squelette du maillage, tandis que les inconnues attachées aux mailles ne sont que localement couplées. Ainsi, les inconnues situées dans chaque maille peuvent être éliminées localement par une technique de complément

Schur appelée *condensation statique* [74]. La figure 3 montre les inconnues de vitesse pour les méthodes cG-FEM, dG-FEM et dG-FEM hybrides. Dans le contexte des écoulements incompressibles, un autre avantage des dG-FEM et des dG-FEM hybrides est la possibilité d'utiliser une approximation de même ordre pour la vitesse et la pression. Dans les méthodes dG-FEM, cela est réalisé en ajoutant un terme de pénalité sur les sauts de pression, alors que cette pénalité n'est pas nécessaire dans les méthodes dG-FEM hybrides.



(a) Inconnues de vitesse pour les méthodes cG-FEM.

(b) Inconnues de vitesse pour les méthodes dG-FEM.

(c) Inconnues de vitesse pour les méthodes dG-FEM hybrides.

Figure 3: Inconnues de vitesse pour les méthodes cG-FEM (à gauche), dG-FEM (au centre) et dG-FEM hybrides (à droite) en utilisant des approximations polynomiales du second ordre pour les inconnues des maille et de face. Chaque cercle plein représente une inconnue à valeurs dans $\mathbb{R}^d$, les cercles bleus représentent les inconnues locales, et les cercles rouges les inconnues globales. Le nombre d'inconnues globales est de 21 pour les méthodes cG-FEM, de 32 pour les méthodes dG-FEM, et de 12 pour les méthodes dG-FEM hybrides.

La formulation mixte-duale des méthodes HDG a été introduite sous sa forme actuelle il y a environ une décennie dans [35]. L'application aux problèmes d'écoulement a été étudiée dans [36], [91], [100], [101], et plus récemment, des avancées ont été accomplies dans les stratégies d'adaptation du degré [65], le couplage avec la description NURBS de la géométrie du domaine [117], et de nouvelles formulations superconvergentes [62]. Un livre a été consacré aux méthodes HDG [49]. De plus, les méthodes hybrides d'ordre élevé (HHO) ont été introduites dans [44] pour l'élasticité linéaire sans verrouillage et dans [41] pour la diffusion linéaire. Ces méthodes ont été appliquées avec succès aux écoulements de Stokes incompressibles dans [3], [46], aux écoulements de Navier-Stokes dans [15], [47], et aux écoulements viscoplastiques dans [28]. Deux livres ont été consacrés aux méthodes HHO [31], [42].

Comme le montre [33], les méthodes HHO peuvent s'intégrer dans le cadre des méthodes HDG, car on peut reformuler les équations HHO sous forme d'équations de bilan local avec des flux numériques équilibrés. De plus, les méthodes HDG et HHO sont étroitement liées aux méthodes des éléments virtuels non conformes (ncVEM) et aux méthodes de Galerkin faible (weak Galerkin, WG), comme le montrent [31]–[33], [43], [48]. Les méthodes HDG et HHO sont formulées au moyen d'opérateurs de reconstruction locaux et de stabilisation. L'opérateur de reconstruction dans HHO agitant sur la variable primaire et sa trace; il correspond à la variable mixte (ou duale) dans HDG. Il convient de noter que, en rendant explicite l'opérateur de reconstruction, les méthodes HHO peuvent être formulées à l'aide de la formulation primale du problème, tandis que les méthodes HDG reposent sur l'introduction d'une variable duale locale (appelée flux), par exemple, $\boldsymbol{L} := -\boldsymbol{\nabla} \boldsymbol{u}$. En résumé, les méthodes HHO, HDG (et

ncVEM et WG) diffèrent uniquement par le choix des inconnues discrètes pour les inconnues de maille, de face et pour le flux, ainsi que par la conception de la stabilisation utilisée pour définir la trace du flux numérique.

Les atouts des méthodes HDG et HHO comprennent sont les suivantes: (i) la capacité à obtenir aisément des discrétisations de degré élevé, (ii) la flexibilité pour utiliser des maillages généraux avec des mailles polyédriques (avec des faces planes), (iii) la conservation locale au niveau des mailles au moyen de flux équilibrés, (iv) la réduction des coûts de calcul par rapport à dG-FEM grâce au couplage global des inconnues du squelette uniquement, (v) la possibilité d'obtenir une superconvergence en norme L^2 sur la variable primaire grâce à un post-traitement local reposant sur les informations contenues dans la variable hybride.

Pour les problèmes impliquant des interfaces ou des frontières extérieures courbes, une possibilité de préserver la précision tout en utilisant des méthodes de discrétisation classiques consiste à recourir à des maillages courbes d'ordre élevé. Les exemples proposés dans la littérature comprennent [111], [128]. En particulier, la technique des NURBS permet une représentation précise de la géométrie [83], [115], [116] et permet une transition simple des descriptions CAO à des descriptions computationnelles dans un large éventail d'applications de l'ingénierie.

Cependant, la convergence optimale des méthodes HDG et HHO n'est assurée sur le plan théorique que lors de l'utilisation des mailles avec des faces planes. Une manière de gérer les interfaces et les frontières courbes dans les méthodes HDG et HHO est alors d'utiliser des maillages immergés ne tenant pas compte de la position de l'interface. Une autre motivation pour utiliser des maillages immergés est que dans des problèmes réalistes où la position de l'interface est inconnue, le suivi des interfaces nécessite des mises à jour du maillage qui peuvent entraîner de grandes déformations conduisant à des mailles fortement déformées. De plus, la génération de maillages adaptés aux frontières et aux interfaces peut être extrêmement chronophage.

0.4 Maillages immergés

Au cours de la dernière décennie, d'importants progrès ont été réalisés pour dissocier la description de la géométrie du maillage computationnel. Les méthodes qui ne tiennent pas compte de la position de l'interface appartiennent à cet ensemble d'approches où le maillage ne s'adapte pas à la géométrie. La formulation variationnelle doit alors être modifiée car l'interface ou la frontière du domaine de calcul qui peuvent couper certaines mailles.

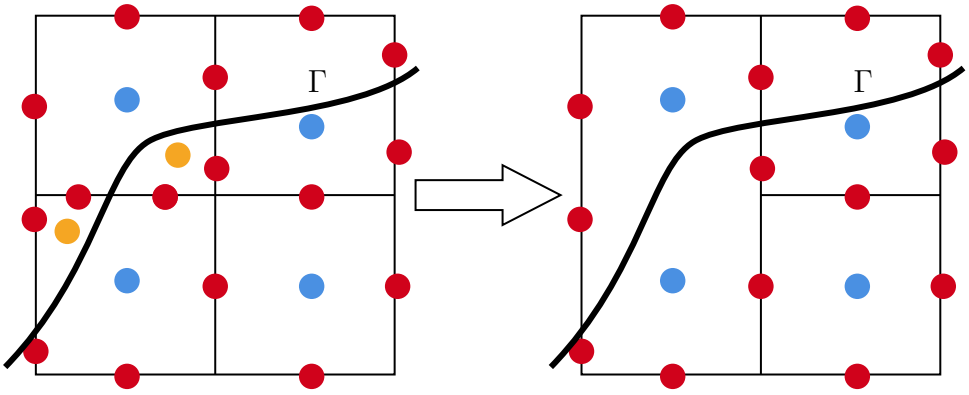
incorporer les contraintes sur l'interface et la frontière dans le calcul. Diverses techniques ont été développées, comme la méthode des domaines fictifs ou de pénalité de volume [5], [66]–[68] pour les domaines à frontière courbe. Les conditions à la frontière extérieure sont alors imposées par multiplicateurs de Lagrange ou par des termes de pénalité. Le défi de choisir un paramètre de pénalité approprié est un inconvénient de ces méthodes.

Pour pallier cette limitation, une méthode avec maillages immergés a été élaborée dans [75], [76], en s'appuyant sur le concept de la méthode des éléments finis introduite dans [9]. Pour traiter les conditions d'interface, l'idée clé de la méthode est de doubler les inconnues dans les mailles coupés par l'interface, enrichissant ainsi l'espace d'approximation et obtenant une représentation plus précise de la solution à proximité de l'interface. De plus, la définition des fonctions de base reste indépendante de la forme de la maille coupée, et aucun degré de liberté supplémentaire n'est introduit sur l'interface (ou la frontière). Cette méthode peut être combinée avec la méthode de Nitsche [103], ou les multiplicateurs de Lagrange [11], [19], [24]

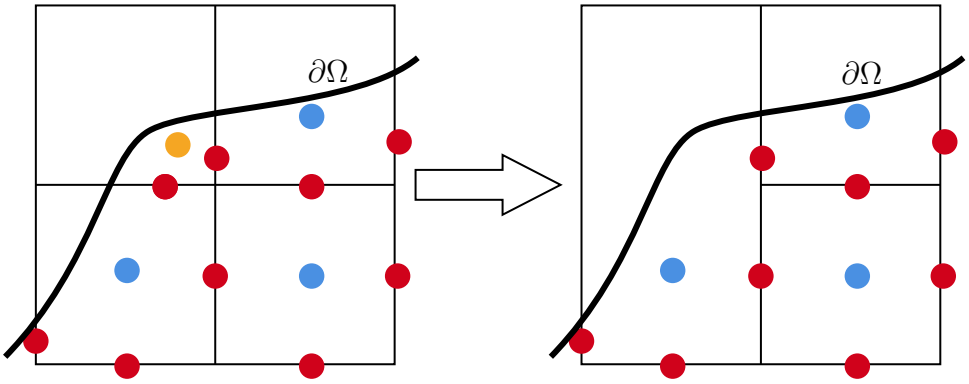
pour prescrire des conditions d'interface et de frontière.

Un inconvénient bien connu des méthodes de maillages immergés est la présence de coupes quasi-singuliers dans le maillage, créant des mailles avec une petite partie de leur volume à l'intérieur du domaine de calcul ou à l'intérieur de l'un des deux sous-domaines. Cette situation conduit souvent à des systèmes linéaires mal conditionnés, qui, à leur tour, affectent négativement la précision des calculs. Pour résoudre ce problème, des solutions possibles sont la préconditionnement [92], la pénalité fantôme [18], [25], [26], [97], la méthode de frontière déplacée [94], [95], ou l'agglomération de mailles [78], [87]. En particulier, les méthodes d'agglomération combinent les mailles voisines en effaçant les inconnues associées mal coupée à la maille, et en générant une nouvelle maille bien coupée par agglomération. Une procédure d'agglomération limitant l'agglomération aux voisins les plus proches est présentée dans [20]. Une alternative un peu plus souple aux procédures d'agglomération est la méthode d'extension [8], [99]. Cette méthode ne génère pas un nouvel espace d'approximation, mais utilise plutôt les inconnues de la maille bien coupée sélectionnée par la procédure d'agglomération et extrapole les fonctions de base dans la maille mal coupée. Dans le contexte de la cG-FEM, nous renvoyons le lecteur à [21].

Les approches d'agglomération de mailles et d'extension conviennent particulièrement aux méthodes HDG et HHO en raison de la capacité de ces méthodes à gérer des maillages non conformes. De tels développements ont commencé assez récemment, en 2018, et ont conduit à des résultats prometteurs pour des problèmes elliptiques [20], d'élasticité [38], et pour des écoulements de Stokes incompressibles immiscibles séparés par une interface fixe [22], [73]. La figure 4 présente un exemple d'agglomération de maille dans le cas d'un problème d'interface (rangée supérieure) et d'un problème de frontière courbe (rangée inférieure).



(a) Inconnues de vitesse près de l'interface pour le maillage d'origine (à gauche) et le maillage aggloméré (à droite).



(b) Inconnues de vitesse près de la frontière extérieure pour le maillage d'origine (à gauche) et le maillage aggloméré (à droite).

Figure 4: Inconnues de vitesse pour les méthodes de Galerkin discontinu hybrides utilisant des approximations polynomiales d'ordre zéro dans les mailles et les faces. Rangée supérieure: problème d'interface; rangée inférieure: problème de avec frontière courbe; colonne de gauche: avant agglomération; colonne de droite: après agglomération. Chaque cercle plein représente une inconnue à valeurs dans $\mathbb{R}^d$, les cercles bleus représentent les inconnues des mailles bien coupées, les cercles oranges représentent les inconnues des mailles mal coupées, et les cercles rouges représentent les inconnues hybrides globales. On notera le doublement des inconnues dans les mailles et les faces coupées par l'interface Υ.

0.5 Aperçu de la thèse et objectifs

Cette thèse contribue à la modélisation et à la simulation numérique des problèmes multi-fluides en explorant le potentiel des méthodes de Galerkin discontinu hybrides (HDG et HHO) pour les écoulements de Stokes avec deux fluides incompressibles et immiscibles. De plus, cette thèse étude l'utilisation des techniques de maillages immergés pour garantir des calculs précis et fiables dans le cas d'interfaces ou de frontières courbes.

Dans le chapitre 2, nous développons des méthodes HHO dans le cadre des maillages immergés pour le problème de Stokes avec deux fluides incompressibles et immiscibles en utilisant une méthode de fonction de niveau et une représentation polynomiale par morceaux et continue de l'interface. Dans le chapitre 3, nous étudions numériquement divers problèmes avec une in-

terface connue puis inconnue. Nous commençons par examiner l'équilibre pour un écoulement de cisaillement pur. Dans ce cas, la forme de l'interface est connue (elle est elliptique) [39], [125]. Ensuite, nous proposons un solveur de point fixe couplant le solveur HHO avec une méthode de fonction de niveau pour traiter les problèmes d'interface inconnue. Nous présentons des résultats numériques pour divers cas tests à fin de mettre en avant les capacités de la méthode dans la simulation d'interfaces complexes. Les résultats contenus aux chapitres 2 et 3 sont publiés dans [108].

Dans le chapitre 4, nous nous concentrons sur les méthodes HDG et leur utilisation pour résoudre des problèmes de Stokes avec deux fluides incompressibles et immiscibles avec des maillages immergés. Nous développons un schéma de discrétisation HDG efficace et précis dans lequel les interfaces et les frontières, définies en CAO, sont exactement décrites au moyen de courbes NURBS. De plus, nous présentons une stratégie d'extension d'éléments pour améliorer les performances du solveur HDG. Le chapitre 5 se penche, quant à lui, sur l'étude de la micro-fluidique en utilisant des méthodes HDG. Nous analysons la conditionnement et le comportement des erreurs, et étudions l'efficacité et la précision de l'approche. Nous présentons deux cas tests dans des domaines géométriques complexes. Dans le premier, nous étudions le problème de Stokes à un fluide dans un mélangeur micro-fluidique; dans le second, nous étudions un écoulement d'émulsion dans un milieu poreux. Nous utilisons une adaptation polynomiale basée sur un estimateur a posteriori pour obtenir des résultats avec au moins deux chiffres significatifs.

Enfin, le chapitre 6 clarifie le lien entre les méthodes HHO et HDG pour le problème de Stokes dans le cas mono-fluide puis bi-fluide.

Chapter 1

Introduction

1.1 Motivations

Nowadays, high-fidelity simulations of fluid dynamics problems have gained increasing importance, resulting in a growing demand for accurate computational methods. Nonetheless, such simulations present considerable challenges, especially in problems that involve complex or evolving geometries such as microfluidics [10], [61], [120], petroleum reservoirs [102], groundwater flows [109], energy storage reservoirs [88], and geologic carbon sequestration [104]. One particularly challenging aspect is the accurate resolution of multi-fluid problems, where immiscible fluids interact while exhibiting distinct flow characteristics, see Figure 1.1. In such problems, the fluid interface can be subject to deformations, breakups, and complex shape changes, necessitating robust numerical methods capable of accurately capturing these dynamic phenomena.

Figure 1.1: Snapshot of three microdroplets, moving from left to right, containing dark-colored dye (image inspired by [60]).

This Thesis focuses on the so-called *Stokes interface problem*, which involves immiscible fluids in contact and with surface tension effects at their interface. We only consider steady problems where each fluid is governed by the steady, incompressible Stokes equations. Using the Stokes rather than the Navier–Stokes equations is, in the present context, reasonable owing to the small length scales, and therefore Reynolds number, involved in the problem. Furthermore, the interface position is, in some cases, known a priori, but is, in general, part of the unknowns of the problem. For the sake of simplicity, we focus on the case of two fluids only and on two-dimensional flows. It is worth noting that in microfluidics, such as lab-on-a-chip devices, two-dimensional problems are often not far from reality, especially when the thickness of the domain is very small.

This Thesis aims at exploring and comparing the Hybridizable Discontinuous Galerkin (HDG) method and the Hybrid High-Order (HHO) method, combined with a geometrically unfitted approach, for solving the Stokes interface problem. In Section 1.2, we present the governing equations and discuss some techniques to represent the interface between the two fluids or the possibly curved boundary of the computational domain. Both the Cauchy stress and the full gradient formulations will be considered. In Section 1.3, we motivate the use of hybridized discretization methods, namely HDG and HHO. In Section 1.4, we discuss the use of unfitted meshes. In this Thesis, we use boldface notation for $\mathbb{R}^d$-valued fields and $\mathbb{R}^{d\times d}$-valued tensors and for functional spaces composed of such fields and tensors.

1.2 Stokes interface problem

Let us consider two fluids such that one drop of a fluid, indexed by 1, is immersed into an external fluid, indexed by 2. Let $\Omega \subset \mathbb{R}^d$, $d := 2$, be the computational domain occupied by the two fluids so that each fluid i resides in Ω_i (connected, bounded, open Lipschitz set) and $\overline{\Omega} := \overline{\Omega}_1 \cup \overline{\Omega}_2$. The manifold $\Upsilon := \partial\Omega_1 \cap \partial\Omega_2$ is the interface between the two fluids. For simplicity, we assume that Υ does not touch the boundary $\partial\Omega$ of Ω, so that the index 1 refers to the interior subdomain Ω_1, with $\partial\Omega_1 = \Upsilon$, and the index 2 refers to the exterior subdomain Ω_2, with $\partial\Omega_2 = \partial\Omega \cup \Upsilon$, see Figure 1.2. The interface Υ is assumed to be orientable, and such that its unit normal $\boldsymbol{n}_\Upsilon$, and its curvature, H_Υ, can be defined everywhere on Υ. By convention, $\boldsymbol{n}_\Upsilon$ points from Ω_1 to Ω_2, i.e., Υ is oriented counter-clockwise. On the external boundary $\partial\Omega$,

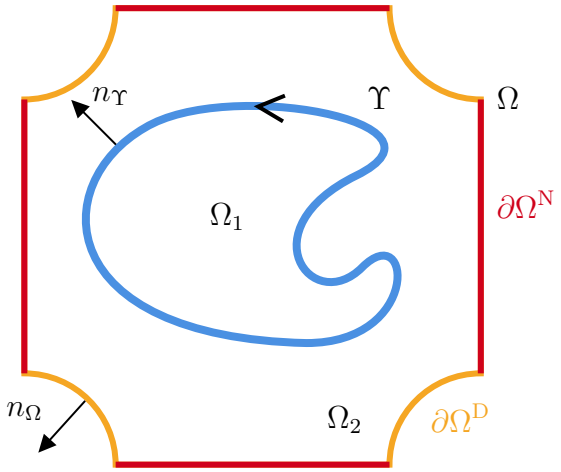


Figure 1.2: Computational domain Ω, subdomains Ω_i, $i \in \{1, 2\}$, interface Υ (blue), Dirichlet boundary $\partial\Omega^{\mathrm{D}}$ (orange), Neumann boundary $\partial\Omega^{\mathrm{N}}$ (red), unit normal $\boldsymbol{n}_\Upsilon$ orienting Υ, and outward unit normal to Ω, $\boldsymbol{n}_\Omega$.

Dirichlet and Neumann boundary conditions are prescribed on $\partial\Omega^{\mathrm{D}}$ and $\partial\Omega^{\mathrm{N}}$, respectively, with $\overline{\partial\Omega^{\mathrm{D}}} \cup \overline{\partial\Omega^{\mathrm{N}}} = \partial\Omega$ and $\partial\Omega^{\mathrm{D}} \cap \partial\Omega^{\mathrm{N}} = \emptyset$. We impose the velocity $\boldsymbol{g} \in H^{1/2}(\partial\Omega^{\mathrm{D}}; \mathbb{R}^d)$ on the Dirichlet boundary, and the traction $\boldsymbol{t} \in L^2(\partial\Omega^{\mathrm{N}}; \mathbb{R}^d)$ on the Neumann boundary. Additionally, we impose the body force $\boldsymbol{f}_i \in L^2(\Omega_i; \mathbb{R}^d)$ in the subdomain Ω_i, $i \in \{1, 2\}$.

1.2.1 Governing equations

In the *Cauchy stress formulation* of the Stokes interface problem, we seek the velocity and pressure fields $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i;\mathbb{R}^d) \times L^2(\Omega_i)$, $i \in \{1,2\}$, such that

$$-\nabla\cdot\boldsymbol{\sigma}_i = \boldsymbol{f}_i \qquad \text{in } \Omega_i, \qquad i\in\{1,2\}, \tag{1.1a}$$

$$\nabla\cdot\boldsymbol{u}_i = 0 \qquad \text{in } \Omega_i, \qquad i\in\{1,2\}, \tag{1.1b}$$

$$\boldsymbol{u}_2 = \boldsymbol{g} \qquad \text{on } \partial\Omega^{\mathrm{D}} \tag{1.1c}$$

$$\boldsymbol{\sigma}_2\boldsymbol{n}_\Omega = \boldsymbol{t} \qquad \text{on } \partial\Omega^{\mathrm{N}}, \tag{1.1d}$$

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}]\!]\boldsymbol{n}_\Upsilon = \gamma H_\Upsilon \boldsymbol{n}_\Upsilon \qquad \text{on } \Upsilon, \tag{1.1e}$$

where the total stress tensor for each fluid is

$$\boldsymbol{\sigma}_i := 2\mu_i\nabla^{\mathrm{s}}\boldsymbol{u}_i - p_i\boldsymbol{I}_d, \quad \nabla^{\mathrm{s}}\boldsymbol{u}_i := \frac{1}{2}(\nabla\boldsymbol{u}_i + \nabla\boldsymbol{u}_i^{\mathrm{T}}), \tag{1.2}$$

μ_i is the viscosity of the fluid i, $\boldsymbol{n}_\Omega$ is the outward unit normal to Ω, and $\boldsymbol{I}_d$ is the identity tensor. Moreover, for a generic quantity $\odot$ defined on $\Omega_1\cup\Omega_2$, which can be scalar-, vector- or tensor-valued,

$$[\![\odot]\!] := (\odot|_{\Omega_1})|_\Upsilon - (\odot|_{\Omega_2})|_\Upsilon \tag{1.3}$$

denotes the component-wise jump across Υ. The conditions (1.1e) enforce the continuity of the velocity and the equilibrium of forces across the interface. The latter accounts for surface tension which is modeled according to Laplace's law with γ the surface tension. Notice that the tangential component of shear stress is continuous across Υ.

In the so-called *full gradient formulation* of the Stokes interface problem, we seek the velocity and pressure fields $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i;\mathbb{R}^d) \times L^2(\Omega_i)$, $i \in \{1,2\}$, solving (1.1), but with (1.2) replaced by

$$\boldsymbol{\sigma}_i' := \mu_i\nabla\boldsymbol{u}_i - p_i\boldsymbol{I}_d. \tag{1.4}$$

Although $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}_i'$ have the same divergence inside Ω_i owing to (1.1b), the use of $\boldsymbol{\sigma}_i'$ results in different interface and Neumann conditions, since (1.1d) and (1.1e) now become

$$\boldsymbol{\sigma}_2'\boldsymbol{n}_\Omega = \big(\mu_2\nabla\boldsymbol{u}_2 - p_2\boldsymbol{I}_d\big)\boldsymbol{n}_\Omega = \boldsymbol{t} \qquad \text{on } \partial\Omega^{\mathrm{N}}, \tag{1.5a}$$

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}']\!]\boldsymbol{n}_\Upsilon = [\![\mu\nabla\boldsymbol{u} - p\boldsymbol{I}_d]\!]\boldsymbol{n}_\Upsilon = \gamma H_\Upsilon\boldsymbol{n}_\Upsilon \quad \text{on } \Upsilon, \tag{1.5b}$$

Remark 1.1 (Uniqueness). When $\partial\Omega^{\mathrm{N}} = \emptyset$, the Stokes interface problem is solvable up to a global additive constant on the pressure, which can be fixed by requiring an additional condition on the pressure. To fix the ideas, we impose

$$\sum_{i\in\{1,2\}}\int_{\Omega_i} p_i = 0. \tag{1.6}$$

Remark 1.2 (Units). For simplicity, we assume that the equations are written in non-dimensional form. Otherwise, the units are $[m\cdot s^{-1}]$ for $\boldsymbol{u}$, $[Pa]$ for p, $[Pa\cdot s]$ for μ, $[m^{-1}]$ for H_Υ and $[Pa\cdot m]$ for γ.

1.2.2 Interface and boundary description

In this section, we briefly outline how the interface Υ and the boundary $\partial\Omega$ can be described mathematically. Notice that $\partial\Omega$ is always known a priori, but this is not necessarily the case for the interface Υ.

1.2.2.1 Level-set description

A fairly general possibility to describe the interface is to assume that a level-set function is available. This function can be known a priori, as for a fixed interface, or result from an approximation, as for an unknown interface (see Section 1.2.2.2 for further insight). A level-set function $\phi : \Omega \to \mathbb{R}$ implicitly describes a generic curve $\Upsilon \subset \Omega$ as its zero level-set, i.e.,

$$\Upsilon := \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) = 0\}. \tag{1.7}$$

Assuming that ϕ is of class $\mathcal{C}^2$ in Ω, one can then define a unit normal field and a curvature field in Ω as follows:

$$\boldsymbol{n}_\phi = \frac{\nabla\phi}{\|\nabla\phi\|_{\ell^2}}, \qquad H_\phi = -\nabla\cdot\boldsymbol{n}_\phi = -\frac{\Delta\phi}{\|\nabla\phi\|_{\ell^2}} + \frac{1}{\|\nabla\phi\|_{\ell^2}^3} D^2\phi(\nabla\phi,\nabla\phi), \tag{1.8}$$

where $\|\cdot\|_{\ell^2}$ denotes the Euclidean norm in $\mathbb{R}^d$ and $D^2\phi(\cdot,\cdot)$ is the quadratic form associated with the Hessian of ϕ. The unit normal and the curvature on Υ are then

$$\boldsymbol{n}_\Upsilon := \boldsymbol{n}_\phi|_\Upsilon, \qquad H_\Upsilon := H_\phi|_\Upsilon. \tag{1.9}$$

Moreover, the knowledge of the level-set function enables a straightforward determination of the domain Ω_i, $i \in \{1,2\}$, by evaluating the sign of $\phi(\boldsymbol{x})$, for any $\boldsymbol{x} \in \Omega$.

1.2.2.2 Unknown interface: level-set transport problem

Level-set methods naturally handle unsteady problems with moving interfaces [107], [113], [122]. As such, they can be used in the present context of finding an unknown interface in a steady problem by introducing a fictitious time $t \in [0,T]$, with $T > 0$, and performing a fixed point iterative method. We extend the definition (1.7) to unsteady curves $\Upsilon : \Omega \times [0,T] \to \mathbb{R}$, as follows:

$$\Upsilon(t) := \{\boldsymbol{x} \times t \in \Omega \times [0,T] : \phi(\boldsymbol{x},t) = 0\}, \tag{1.10}$$

and evolve the level-set function in time according to the following transport problem:

$$\partial_t\phi + \nabla\cdot(\boldsymbol{u}\phi) = 0 \qquad \text{in } \Omega\times(0,T), \tag{1.11a}$$
$$\phi(\boldsymbol{x},t) = \phi^{\text{in}}(\boldsymbol{x}) \qquad \text{on } \partial\Omega_{\text{in}}\times(0,T), \tag{1.11b}$$
$$\phi(\boldsymbol{x},0) = \phi^0(\boldsymbol{x}) \qquad \text{in } \Omega, \tag{1.11c}$$

where $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ is the velocity field moving the interface, $\partial\Omega_{\text{in}} := \{\boldsymbol{x} \in \partial\Omega : \boldsymbol{u}\cdot\boldsymbol{n}_\Omega < 0\}$ denotes the inflow boundary, and ϕ^0 is the initial level-set, typically defined as a signed distance function. Whenever $\boldsymbol{u}$ results from the Stokes interface problem (1.1), one can expect that $\boldsymbol{u}\cdot\boldsymbol{n}$ approaches zero as the interface approaches its equilibrium position. This naturally leads to a fixed-point procedure that is discussed in more detail in Chapter 3.

It is useful to keep in mind that level-set methods also come with certain disadvantages. Firstly, level-set functions can become degenerate, resulting in the formation of sharp edges or in the loss of mass. To address these issues, researchers have proposed various methods, such as reinitialization [122], [123], or the incorporation of artificial viscosity [40], [72]. Secondly, level-set methods may incur significant computational costs, particularly when dealing with rapidly moving interfaces. Indeed, the continuous updates required for tracking interface evolution can lead to an increased computational overhead, potentially impacting the efficiency of the method. Despite these drawbacks, level-set methods continue to be a widely used tool in various fields of computational mechanics for simulating evolving interfaces, owing to their inherent ability to handle topological changes and to provide a versatile framework for interface tracking.

1.2.2.3 CAD interface and boundary

When a CAD geometry is available, the natural approach is to consider Non-uniform rational B-spline (NURBS) curves to exactly describe the interface and the boundary. For simplicity, let us consider an interface Υ described by one NURBS curve $\boldsymbol{C}(\lambda) : \lambda \to \Upsilon$, of class at least C^1, with $\lambda \in [\lambda_a, \lambda_b]$ its parametric domain (see Section 1.4.2.2 for more details). One can then define the tangent $\boldsymbol{t}_{\boldsymbol{C}}(\lambda)$ to Υ by computing the first derivative with respect to λ, i.e.,

$$\boldsymbol{t}_\Upsilon(\lambda) = \boldsymbol{t}_{\boldsymbol{C}}(\lambda) = \frac{\mathrm{d}\boldsymbol{C}(\lambda)}{\mathrm{d}\lambda}.$$

Then, the unit normal and the curvature on Υ are evaluated as follows:

$$\boldsymbol{n}_\Upsilon = \boldsymbol{n}_{\boldsymbol{C}} = \frac{\frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda}}{\left\| \frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda} \right\|_{\ell^2}}, \qquad H_\Upsilon = H_{\boldsymbol{C}} = \left\| \frac{\mathrm{d}\boldsymbol{t}_{\boldsymbol{C}}(\lambda)}{\mathrm{d}\lambda} \right\|_{\ell^2}. \tag{1.12}$$

Notice that the knowledge of the NURBS interface does not allow for a straightforward determination of the domain Ω_i, $i \in \{1, 2\}$. A possible strategy is presented in Chapter 4. Notice also that with the present approach, the orientation of Υ is implicitly specified by the map $\boldsymbol{C}(\lambda)$.

1.3 Hybrid discretization methods for elliptic problems

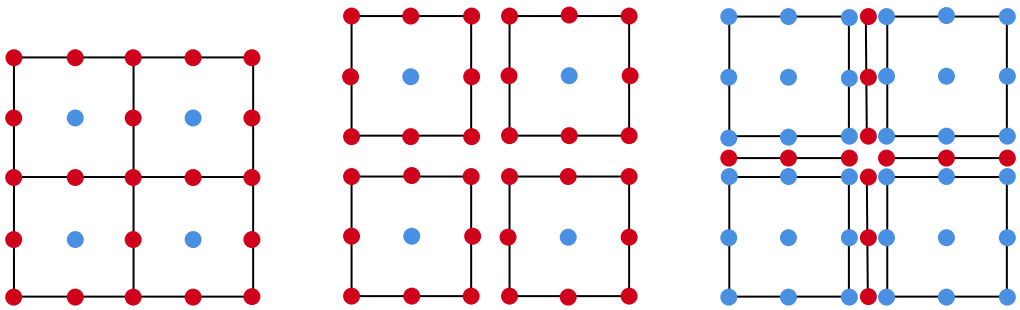
In recent years, high-order methods have emerged as a competing solution over conventional low-order methods in terms of computational efficiency and quality of the results [82], [127].

Classical methods to discretize partial differential equations (PDEs), such as those encountered in computational fluid dynamics are finite volume methods (FVM) [56] and continuous Galerkin finite element methods (cG-FEM) [17], [51]–[53]. FVM are derived by applying a flux balance in each mesh cell. However, FVM employ piecewise constant approximations, which can lead to limitations in accurately representing spatial variations. On the other hand, cG-FEM are based upon a variational formulation of the underlying PDE, and approximate the solution using cell-wise polynomials which are continuous across the mesh interfaces. This latter approach enables high-order polynomial approximations, but suffers from stability issues for convection-dominated flows, lacks of a local conservation property at the cell level (a conservation property is available at the level of macro-cells sharing a given mesh vertex), and

requires compatible (inf-sup stable) discretizations spaces for the velocity and the pressure (notice though that many stabilization techniques exist, see, e.g., [53, Chaps. 62-63] for a review). Additionally, cG-FEM rely on conforming meshes, which can be limiting for problems with complex geometries.

A different approach is offered by discontinuous Galerkin finite element methods (dG-FEM) [45], [80]. dG-FEM employ high-order cell-wise discontinuous polynomials to approximate the solution and enforce a flux balance at the mesh interfaces in the spirit of FVM. dG-FEM offer the advantage of robustness with respect to convection-dominance owing to numerical fluxes and flexibility in the choice of velocity/pressure discretizations owing to appropriate stabilizations. The main disadvantage of dG-FEM is the considerable increment of globally coupled degrees of freedom (DoFs) required to compute, on a given mesh, the approximate solution. Moreover, dG-FEM, as well as cG-FEM, can be sensitive to highly distorted meshes. One approach to tame the computational burden, while preserving the high order of approximation, is the concept of hybridization, first introduced for mixed methods in [57], see also [6].

Hybridized discretization methods have gained significant popularity in recent years, as reflected by the numerous minisymposia dedicated to these techniques in international conferences, e.g., ICOSAHOM (London, 2018), FEF (Chicago, 2019) and WCCM-ECCOMAS (Paris, 2020). The key idea of hybridization is to introduce additional unknowns on the mesh skeleton, which is the set of faces or edges shared by the mesh cells. These additional unknowns, called *hybrid* unknowns, allow one to reduce the globally coupled DoFs to those placed on the mesh skeleton, whereas the cell unknowns become only locally coupled. Thus, the unknowns located in each mesh cell can be locally eliminated by a Schur complement technique called *static condensation* [74]. Figure 1.3 displays the velocity unknowns for cG-FEM, dG-FEM, and hybridized dG-FEM. In the context of incompressible flows, another advantage of dG-FEM and hybridized dG-FEM is the possibility of using equal-order approximation for the velocity and the pressure. In dG-FEM, this is achieved by adding a penalty term on the pressure jumps, whereas this penalty is not needed in hybridized dG-FEM.



(a) Velocity unknowns for cG-FEM.

(b) Velocity unknowns for dG-FEM.

(c) Velocity unknowns for hybridized dG-FEM.

Figure 1.3: Discrete velocity unknowns for cG-FEM (left), dG-FEM (center) and hybridized dG-FEM (right) using second-order polynomial approximations for both the cell and the hybrid unknowns. Each bullet represents one $\mathbb{R}^d$-valued unknown, blue bullets represent local unknowns, red bullets global unknowns. The number of global unknowns is 21 for cG-FEM, 32 for dG-FEM, and 12 for hybridized dG-FEM.

The mixed-dual formulation of Hybridizable discontinuous Galerkin (HDG) methods has been introduced in its present-day form about a decade ago in [35]. The application to flow problems was studied in [36], [91], [100], [101] and, more recently, advances involving degree-adaptive strategies [65], coupling with NURBS description of the geometry of the domain [117] and novel superconvergent formulations [62] have been explored. One book has been devoted to HDG methods [49]. Furthermore, Hybrid High-Order (HHO) methods have been introduced in [44] for locking-free linear elasticity and in [41] for linear diffusion. They have been successfully applied to incompressible Stokes flows in [3], [46], Navier-Stokes flows in [15], [47], and viscoplastic flows in [28]. Two books have been devoted to HHO methods [31], [42].

As shown in [33], HHO methods are embedded into the broad framework of HDG methods since one can reformulate the HHO equations as local balance equations with equilibrated numerical fluxes. In addition, HDG and HHO are closely related to nonconforming virtual element methods (ncVEM) and to weak Galerkin methods (WG) as shown in [31]–[33], [43], [48]. HDG and HHO methods are formulated by means of local reconstruction and stabilization operators. The reconstruction operator in HHO acting on the primal variable and its trace corresponds to the HDG mixed (or dual) variable. Notice that, by making the reconstruction operator explicit, HHO methods can be formulated using the primal formulation of the problem, whereas HDG methods hinge on the introduction of a local dual variable (called flux), e.g., $\boldsymbol{L} := -\boldsymbol{\nabla} \boldsymbol{u}$. In summary, HHO, HDG (and ncVEM and WG) methods differ only in the choice of the discrete unknowns for the cell unknown, the flux, and the trace variable, and in the design of the stabilization used to define the numerical flux trace. In Chapter 6, we will further investigate the links between HDG and HHO.

Attractive features of HDG and HHO methods include (i) the capability of efficiently devising high-order (and non-uniform) degree discretizations, (ii) the flexibility in using general meshes with polyhedral mesh cells (with planar faces) and non-matching interfaces, (iii) the local conservation at the mesh cell level by means of equilibrated fluxes, (iv) the reduced computational costs with respect to dG-FEM owing to the global coupling of the skeleton unknowns only, and (v) the possibility of achieving L^2-superconvergence on the primal variable by a local postprocessing hinging on the local information contained in the hybrid variable.

For problems involving curved interfaces or a curved boundary, one possibility to preserve accuracy while using classical discretization methods is to resort to high-order curved mesh generation schemes, examples proposed in the literature include [111], [128]. In particular, the Non-uniform rational B-spline (NURBS) technique allows for an accurate geometry representation [83], [115], [116] and enables a seamless transition from geometric to computational descriptions in a wide range of engineering applications that rely on computer-aided design (CAD) geometries. In this case, the use of NURBS allows for an exact description of the interface or boundary.

Nonetheless, HDG and HHO methods are proven to be optimally convergent only when using mesh cells with planar faces. A way to handle curved interfaces and boundaries in HDG and HHO methods is therefore to use geometrically unfitted meshes. Another motivation for using geometrically unfitted meshes is that tracking interfaces in realistic problems where the position of the interface is unknown necessitates mesh updates that may result in large deformations leading to highly distorted mesh cells. In such cases, unstable discretizations may arise, jeopardizing accuracy and reliability. Additionally, the generation of boundary-fitted and interface-fitted meshes in the context of iteration (fixed-point) problems can be extremely

time-consuming, since a new mesh is needed at each iteration.

1.4 Unfitted meshes

In this section, we provide a short literature review on unfitted mesh techniques and we discuss some approaches to discretize the curved interface or boundary.

1.4.1 Literature review

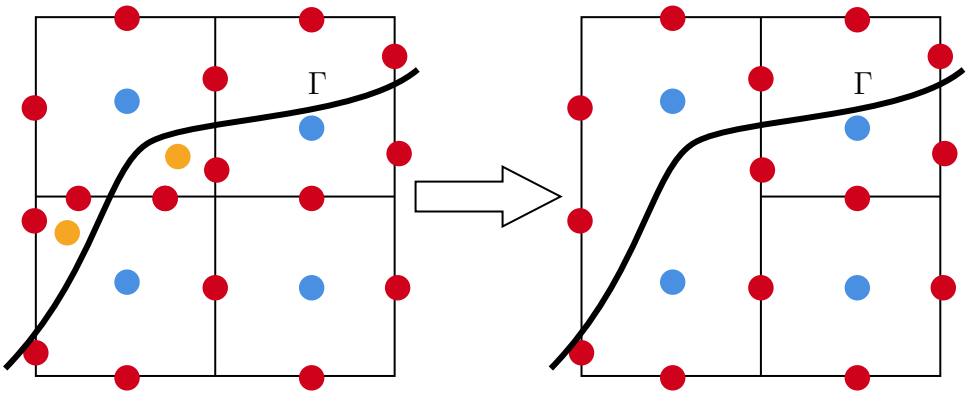
Over the past decade, much progress has been achieved to decouple the geometry description from the computational mesh. Geometrically unfitted methods belong to such approaches where the mesh is not tailored to fit the geometry, and the variational formulation is modified to incorporate constraints on the interface and the boundary into the computation. Various techniques have been developed, as reviewed, e.g., in the proceedings of the 2016 UCL Workshop on the subject [14]. One of the earliest methods is the fictitious domain or volume penalty method [5], [66]–[68]. This method employs larger computational domains with simplified shapes that extend beyond the physical domain. Boundary conditions are enforced using techniques such as Lagrange multipliers or penalty terms. The challenge of selecting an appropriate penalty parameter is a significant drawback of these methods.

To address this limitation, an unfitted method was devised in [75], [76], building upon the concept of the unfitted finite element method introduced in [9]. To handle interface conditions, the key idea of the method is the doubling of the unknowns in the mesh cells cut by the interface, thus enriching the approximation space and achieving a more precise representation of the solution in the vicinity of the interface. Moreover, the definition of the basis functions remains independent of the cut cell shape, and no additional DoFs are introduced on the interface (or the boundary). The method can be combined with Nitsche's method [103], or Lagrange multipliers [11], [19], [24] to prescribe boundary and interface conditions.

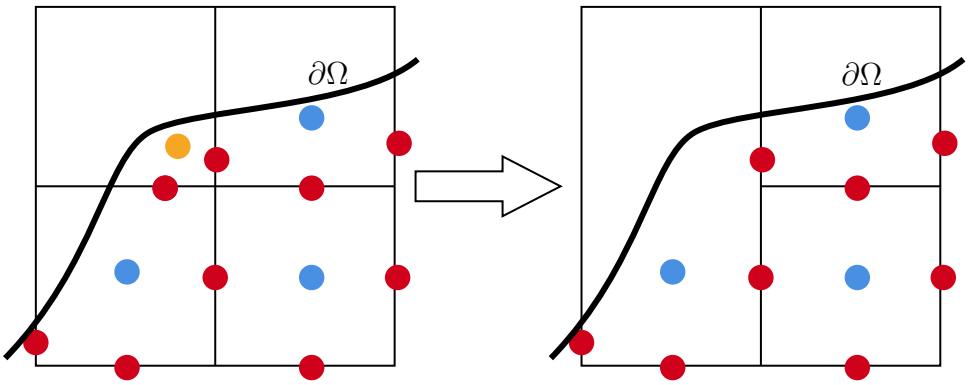
A well-known drawback in unfitted methods is the presence of unfavorable cuts in the mesh, resulting in mesh cells with a small portion of their volume within the computational domain or within one of the two computational domains. This situation often leads to ill-conditioned linear systems, which, in turn, adversely affect the accuracy of the computations. To address this issue, possible solutions are preconditioning [92], weakly consistent penalty (so-called ghost penalty) [18], [25], [26], [97], the shifted boundary method [94], [95], or cell agglomeration [78], [87]. In particular, cell-agglomeration methods address the issue of cut cells by combining neighboring cells, erasing the unknowns associated with the ill-cut cell, and generating a new unique well-cut, larger cell. A cell-agglomeration procedure limiting the agglomeration to the nearest neighbors is presented in [20]. Loose alternatives to cell-agglomeration procedures are element extension and element aggregation methods [8], [99]. Element extension methods do not generate a new approximation space for the agglomerated mesh cell, but utilize the unknowns from the well-cut cell selected by the cell-agglomeration procedure and extrapolate the basis functions into the ill-cut cell. On the other hand, element aggregation methods are suitable to cG-FEM since they are tailored to preserve mesh conformity. In the context of cG-FEM, we refer the reader to [21] for a review.

Interestingly, the cell-agglomeration and the element extension approaches are particularly suitable for HDG and HHO methods owing to the capability of these methods to handle gen-

eral nonconforming meshes with hanging nodes. The extension to hybridized discretization methods has started quite recently, in 2018, delivering promising results for elliptic interface problems [20], elasticity problems [38] and for immiscible incompressible Stokes flows separated by an interface [22], [73]. Figure 1.4 displays an example of cell agglomeration in the case of an interface problem (top row) and a curved boundary problem (bottom row).



(a) Discrete velocity unknowns near the interface for the original (left) and agglomerated (right) mesh.



(b) Discrete velocity unknowns near the external boundary for the original (left) and agglomerated (right) mesh.

Figure 1.4: Discrete velocity unknowns for hybridized discontinuous Galerkin methods using zero-order polynomial approximations for the cell and the hybrid unknowns. Top row: interface problem; bottom row: curved boundary problem; left column: before agglomeration; right column: after agglomeration. Each bullet represents one $\mathbb{R}^d$-valued unknown, blue bullets represent local cell unknowns from well-cut cells, orange bullets local cell unknowns from ill-cut cells, and red bullets global hybrid unknowns. Notice the doubling of the unknowns in the mesh cells and faces cut by the interface Υ.

1.4.2 Geometric discretization

An accurate geometric description of the interface and the boundary is essential to preserve the high-order convergence of HHO and HDG methods. In this context, we explore in this Thesis two geometric descriptions: one based on C^0-piecewise polynomials and one based on Non-uniform rational B-splines (NURBS). The former is simpler and hinges on classical techniques

used in the context of isoparametric FEM. The drawback is that the discrete interface is only C^0, which may impact the accuracy owing to the presence of kinks in the discrete interface at each mesh interface. The NURBS-based description improves on this by allowing for a smoother discrete interface. However, difficulties emerge in the handling of evolving geometries.

1.4.2.1 C^0-piecewise polynomial representation

In this representation, the interface Υ (or the boundary $\partial\Omega$) is approximately represented by a collection of arcs, each arc being generated from the reference interval $\widehat{I} := [0,1]$ by using a geometric mapping whose components are polynomial-valued. The approach we consider is as follows (see Section 2.3.3 for more details). For every cut cell $T \in \mathcal{T}^\Upsilon$ (the collection of all the cut cells), Υ is approximated by a collection of 2^n arcs, where $n \geq 0$ is a user-specified parameter. These arcs are denoted by $\Upsilon^{T,j}$, $j \in \{1{:}2^n\}$. To build each arc $\Upsilon^{T,j}$, we consider $(l+1)$ equidistributed interpolation nodes $\{\boldsymbol{x}_m^{T,j}\}_{m\in\{0:l\}} \subset \Upsilon$. The integer l is a user-specified parameter such that $l \geq k+1$, where k is the degree of the hybrid unknowns. This lower bound is coherent with the consistency analysis of elliptic problems on curved domains, where the geometry error is at least one order higher than the approximation error in the H^1-norm (which is of order $(k+1)$ for the HHO method).

Considering the Lagrange basis of order l, $\{\widehat{\psi}_m\}_{m\in\{0:l\}}$, defined on the reference interval $\widehat{I}$ using equidistributed nodes, the arc $\Upsilon^{T,j}$ is defined as

$$\Upsilon^{T,j} := \boldsymbol{r}^{T,j}(\widehat{I}) \qquad \text{where} \qquad \boldsymbol{r}^{T,j}(\widehat{s}) := \sum_{m\in\{0:l\}} \boldsymbol{x}_m^{T,j}\widehat{\psi}_m(\widehat{s}), \quad \forall \widehat{s} \in \widehat{I}. \tag{1.13}$$

Finally, the fully discrete interface is defined as follows:

$$\Upsilon^{n,l} := \bigcup_{T\in\mathcal{T}^\Upsilon} \bigcup_{j\in\{1:2^n\}} \Upsilon^{T,j}. \tag{1.14}$$

We notice that the fully discrete interface has only C^0-regularity.

1.4.2.2 NURBS representation

NURBS methods are well established for fitted finite element methods, such as the NURBS-enhanced Finite Element Method (NEFEM) [115], [116], [118], and have also recently been integrated into unfitted finite element methods, [90], [96], [99].

A NURBS curve of degree q is a piecewise rational function defined as follows:

$$\boldsymbol{C}(\lambda) = \frac{\left(\sum_{i=0}^{\mathtt{n}_{\mathtt{cp}}} \nu_i \boldsymbol{B}_i C_i^q(\lambda)\right)}{\left(\sum_{i=0}^{\mathtt{n}_{\mathtt{cp}}} \nu_i C_i^q(\lambda)\right)}, \quad \forall \lambda \in [\lambda_a, \lambda_b], \tag{1.15}$$

where $\boldsymbol{B}_i$ and ν_i are the i-th control point and control weight, respectively, $\mathtt{n}_{\mathtt{cp}}+1$ is the number of control points, and $C_{i,q}(\lambda)$ is the i-th B-spline basis function of degree q. The B-spline basis

functions are defined recursively in the parametric space $[\lambda_a, \lambda_b]$ as:

$$\begin{aligned} C_i^0(\lambda) &:= \begin{cases} 1 & \text{if } \lambda \in [\lambda_i, \lambda_{i+1}), \\ 0 & \text{elsewhere,} \end{cases} \\ C_i^k(\lambda) &:= \frac{\lambda - \lambda_i}{\lambda_{i+k} - \lambda_i} C_i^{k-1}(\lambda) + \frac{\lambda_{i+k+1} - \lambda}{\lambda_{i+k+1} - \lambda_{i+1}} C_{i+1}^{k-1}(\lambda), \end{aligned} \tag{1.16}$$

for all $k \in \{1, \dots, q\}$. Moreover, $\Lambda = \{\lambda_0, \cdots, \lambda_{\mathtt{n_k}}\} = \{\lambda_a, \cdots, \lambda_b\}$ is the knot vector of dimension $\mathtt{n_k} + 1$, with $\{\lambda_i\}$ the knots or breakpoints. The amount of control points and knots is related as $\mathtt{n_k} = \mathtt{n_{cp}} + q + 1$. Notice that the curve definition using NURBS changes at each breakpoint, allowing for flexible and smooth representations.

An interface or a boundary is said to be of NURBS-type when it is composed of a finite number $\mathtt{n_c}$ of NURBS curves, such that

$$\bigcup_{j=1}^{\mathtt{n_c}} \boldsymbol{C}^j([\lambda_a^j, \lambda_b^j]). \tag{1.17}$$

NURBS curves allow for a seamless description of complex realistic geometries without the introduction of geometrical error whenever the physical boundary is indeed described using NURBS as is the case in CAD. One drawback to keep in mind is the dependence on the control points and thus the difficulty in describing evolving interfaces.

1.5 Overview and objectives

This Thesis contributes to the field of multi-fluid problems by exploring the potential of HDG and HHO methods in high-fidelity simulations of the Stokes interface problem. Additionally, the Thesis seeks to address the complexities of curved interfaces through the utilization of geometrically unfitted mesh techniques, ensuring accurate and reliable computations. The two methods employ a different notation summarized in Table 1.1.

In Chapter 2, we devise HHO methods in the unfitted framework for the Stokes interface problem using a level-set method and a C^0-piecewise polynomial representation of the interface. In Chapter 3, we study numerically various Stokes interface problems with known and unknown interfaces. We first investigate the equilibrium with a pure shear flow in which case the shape of the interface is known to be elliptic [39], [125]. Then, we propose a fixed-point solver coupling the HHO solver with a level-set method to handle unknown interface problems. Numerical results for various scenarios of unknown interface problems are presented to showcase the capabilities of HHO in simulating complex interfaces. The contents of Chapters 2 and 3 are published in [108].

In Chapter 4, we shift our focus to HDG methods and their utilization in solving interface problems with unfitted meshes. We develop an efficient and accurate HDG discretization scheme where the interfaces and the boundaries, defined in CAD, are exactly described by means of NURBS curves. Additionally, we present an element extension strategy to further improve on the performance of the HDG solver. Chapter 5 delves into the study of microfluidics using HDG methods. We analyze the conditioning and error behavior and showcase the efficiency and accuracy of the approach through numerical results for immersed Stokes problems. Furthermore,

Table 1.1: Notations HHO vs. HDG.

	HHO	HDG
Dimension	d	$\mathtt{n_{sd}}$
Boundaries	$\partial\Omega^{\mathrm{D}}$, $\partial\Omega^{\mathrm{N}}$	Γ_{D}, Γ_{N}
Dirichlet condition	$\boldsymbol{g}$	$\boldsymbol{u}_{\mathrm{D}}$
Source term	$\boldsymbol{f}$	$\boldsymbol{s}$
Spaces	H^1, L^2	$\mathcal{H}^1$, $\mathcal{L}^2$
Spatial discretizations	$\mathbb{P}^k_{d'}(S;\mathbb{R}^d)$	$\mathcal{P}^k(S)$ for elements, $\hat{\mathcal{P}}^k(S)$ for faces
Gradient	∇	$\boldsymbol{\nabla}$
Fluid index i	subscript	superscript
Normals	$\boldsymbol{n}_\Upsilon$, $\boldsymbol{n}_\Omega$	$\boldsymbol{n}^i$
Mesh	cells mesh $T \in \mathcal{T}$	elements Ω_e, $e \in \{1{:}\mathtt{n_{el}}\}$
Skeleton	$\mathcal{F}$ ($\mathcal{F}^o$ the inner)	Γ inner skeleton
Face	(∂T), F	$\partial\Omega_e \to \Gamma_f$
$\mathcal{L}^2$-inner product on faces	$(\cdot,\cdot)$	$\langle\cdot,\cdot\rangle$
Local cell velocity unknowns	$\boldsymbol{v}_T$	$\boldsymbol{v}_e$
Global face velocity unknowns	$\boldsymbol{v}_{\partial\mathcal{T}}$	$\hat{\boldsymbol{v}}$
Jumps	$[\![\odot]\!] = \odot_1 - \odot_2$	$[\![\odot\boldsymbol{n}]\!] = \odot_1\boldsymbol{n}_1 + \odot_2\boldsymbol{n}_2$

we present two main test cases within complex geometrical domains. In the first one, we study the one-fluid Stokes problem in a microfluidic mixer; in the second one, we study the emulsion flow in a porous medium. We employ a polynomial-adaptivity error estimator to achieve results with at least two significant digits.

Finally, Chapter 6 establishes a bridge between HHO and HDG methods for Stokes interface problems. By employing static condensation, the HHO discretization is reformulated in the form of a global transmission problem and local equilibrium conditions. This chapter includes a theoretical comparison between HHO and HDG methods, shedding further light on their respective features.

Part I

Hybrid High-Order methods

Chapter 2

Unfitted HHO solver

2.1 Introduction

In this work, we study the equilibrium of two immiscible, incompressible Stokes fluids separated by a single interface where surface tension effects are present [16], [110], [124]. The interface, whose shape is part of the unknowns of the problem, splits the computational domain into two subdomains, and each subdomain is occupied by a fluid governed by the steady, incompressible Stokes equations. At the interface, the fluid velocities are continuous, the normal velocity vanishes, and the jump of the normal component of the total stress is proportional to the curvature of the interface (Laplace's law). The present model, albeit simplified, has relevant applications in microfluidics, where the surface tension dominates the emulsion process [60]. Over the last decades, microfluidics has gained growing importance in domains such as medicine, biology and chemistry [1], [81].

Solving the above problem computationally is quite challenging because the shape of the interface is unknown. A natural approach is to resort to a fixed-point iterative procedure where each iteration is decomposed into two substeps. In the first substep, the shape of the interface is kept fixed, and a so-called Stokes interface problem is solved, whereby the interface conditions enforce only the continuity of the fluid velocities and the jump of the normal stresses, but the normal velocity at the interface may be nonzero. In the second substep, the flow field of both fluids is kept fixed, and the interface is evolved using a level-set scheme. In this context, using an unfitted method in the first substep is quite attractive since it allows one to use the same background mesh for all the iterations of the fixed-point procedure. The main goal of the present chapter is to develop an unfitted hybrid high-order (HHO) method coupled with a level-set scheme to solve the above interface problem.

HHO methods on fitted meshes have been introduced in [44] for locking-free linear elasticity and in [41] for linear diffusion, and they are closely related to hybridizable discontinuous Galerkin, nonconforming virtual element, and weak Galerkin methods [33]. When applied to incompressible Stokes flows, the HHO method employs hybrid unknowns (face- and cell-based) for the velocity and only cell-based unknowns for the pressure [3], [46]. The method is inf-sup stable, locally conservative, supports polytopal meshes, and is computationally efficient owing to its compact stencil and to the possibility of a local elimination of the cell velocity unknowns by a static condensation procedure. Unfitted HHO methods for elliptic interface problems with

known interface have been derived in [20], [23] using two key ideas. First, one doubles the (cell and face) unknowns in every cut cell without introducing any face unknown at the interface, and the jump conditions at the interface are enforced by means of a consistent penalty technique in the spirit of [76]. Second, a local cell-agglomeration procedure is used to counter the adverse effects of ill-cut cells (see also [87], [119] for cell-agglomeration procedures in the discontinuous Galerkin context). These ideas were extended in [22] to the Stokes interface problem, still assuming a known interface.

In the paper, we extend [20] in three directions, while using the same cell-agglomeration procedure. First, we consider surface tension effects. This, in particular, requires to approximate the curvature of the interface at all the integration points along the interface. Second, the quadratures in the cut cells used in [22] are based on a subpartition of the cut cell using affine triangles, whereas we introduce here a more effective approach based on an isoparametric description of the interface. Third, this work devises, for the first time, a coupling between the unfitted HHO method and a level-set scheme to track the interface. While HHO methods offer various assets (high-order, support of polyhedral meshes, computational efficiency, local conservation), we observe that several other methods are available in the literature to approximate the Stokes interface problem; see, e.g., [12], [27], [30], [77], [85], [112] for finite element and [2], [79], [119] for discontinuous Galerkin discretizations. We also notice that considering the Stokes equations is reasonable in view of microfluidics applications. One can also consider the Navier–Stokes equations, for which HHO methods on fitted meshes were constructed, e.g., in [15].

Our computational study covers two main test cases, which are both two-dimensional. The first one is devoted to the equilibrium between a pure shear flow (enforced far away from the interface, no body forces) and surface tension effects. In this case, the equilibrium interface is known to be an ellipse (the area of the ellipse still remains a free parameter in the problem), and, fixing the viscosity ratio of both fluids, the ellipse eccentricity depends upon the ratio of the prescribed shear to surface tension (also called capillary number) [39], [69], [124]. When the capillary number is zero (no prescribed shear, flow at rest), the ellipse eccentricity is zero, that is, the ellipse becomes a circle. In the present Thesis, we investigate numerically the dependence of the ellipse eccentricity on the capillary number. A linear relationship is expected, as predicted theoretically in [124]. Our contribution is to quantify the slope of this relationship as a function of the viscosity ratio of both fluids. Since the shape of the equilibrium interface is known in this first test case, we do not need to invoke the above fixed-point iterative procedure. In practice, we pose the Stokes interface problem on a finite computational box surrounding the elliptic equilibrium interface, and verify numerically that the expected equilibrium is fairly well attained even on moderately large computational domains.

The second test case involves the more challenging setting where the boundary conditions (and possibly the body forces) are more complex, so that the equilibrium interface has no longer an elliptic shape. The shape of the interface then becomes an unknown of the problem that is determined by the above fixed-point iterative procedure. Notice that we do not perform here a convergence analysis of this fixed-point procedure; this is a difficult problem left for future work. We merely observe that the procedure is consistent since, assuming convergence, the interface becomes stationary so that the normal flow velocity at the interface is zero. The position of the interface is classically described by means of a level-set function. To transport the level-set function, possible approaches are finite volume [105], [106], [121], discontinuous Galerkin [13],

and continuous finite element methods [72]; herein, we use the latter. One important challenge encountered in flow problems dominated by surface tension resides in the approximation of the curvature at the interface. To reduce oscillations, possible strategies are the use of coarser meshes for the level-set discretization (but possibly increasing the so-called mass loss effects), or the smoothing of the normal and the curvature by means of a global projection in the whole computational domain [106] (but possibly affecting the approximation quality). We propose here a somewhat alternative strategy, where we construct a parametric interface of arbitrary order from the level-set function, and then we project the normal and the curvature derived from the level-set function locally on this parametric interface.

The paper is organized as follows. Section 2.2 is devoted to the presentation of the model problem. Section 2.3 describes the unfitted HHO solver for the Stokes interface problem. Section 3.1 presents the results when the shape of the interface is elliptic, whereas Sections 3.2 and 3.3 deal respectively with the methodology and the results when the shape of the interface is unknown. As mentioned above, this chapter is restricted to the two-dimensional setting. The main obstacle to a three-dimensional extension is the handling of the mesh cells cut by the interface, the representation of the latter within the cut cells, and the cell-agglomeration procedure. While these steps are feasible, they represent at the time of writing a further, non-trivial implementation effort.

2.2 Model problem

Let $\Omega_i \subset \mathbb{R}^d$, $d = 2$, be two domains (connected, bounded, open Lipschitz sets) with Ω_i occupied by the fluid indexed by $i \in \{1, 2\}$. The interface is $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$, and the computational domain Ω is $\overline{\Omega} := \overline{\Omega}_1 \cup \overline{\Omega}_2$. For simplicity, we assume that Ω is a polygon/polyhedron so that it can be meshed exactly. Moreover, we assume that the interface Γ is closed and does not touch the boundary $\partial\Omega$ of Ω. By convention, the index 1 refers to the interior subdomain Ω_1 such that $\partial\Omega_1 = \Gamma$, and the index 2 refers to the exterior subdomain Ω_2 such that $\partial\Omega_2 = \partial\Omega \cup \Gamma$, see Figure 2.1. The interface Γ is assumed to be orientable, and such that a unit normal, $\boldsymbol{n}_\Gamma$, can be defined everywhere on Γ. Since $d = 2$, the interface Γ is a one-dimensional manifold. By convention, $\boldsymbol{n}_\Gamma$ points from Ω_1 to Ω_2, i.e., Γ is oriented counter-clockwise.

We consider two immiscible, incompressible fluids separated by the interface Γ. In the so-called *Stokes interface problem*, the interface Γ is prescribed, and we seek the velocity and pressure fields $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i; \mathbb{R}^d) \times L^2(\Omega_i)$, $i \in \{1, 2\}$, such that

$$-\nabla \cdot \boldsymbol{\sigma}_i = \boldsymbol{f}_i \quad \text{in } \Omega_i, \quad i \in \{1, 2\}, \tag{2.1a}$$

$$\nabla \cdot \boldsymbol{u}_i = 0 \quad \text{in } \Omega_i, \quad i \in \{1, 2\}, \tag{2.1b}$$

$$\boldsymbol{u}_2 = \boldsymbol{g} \quad \text{on } \partial\Omega, \tag{2.1c}$$

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}]\!]\boldsymbol{n}_\Gamma = \boldsymbol{g}_N \quad \text{on } \Gamma, \tag{2.1d}$$

where the total stress tensor for each fluid is

$$\boldsymbol{\sigma}_i := 2\mu_i \nabla^{\mathrm{s}} \boldsymbol{u}_i - p_i \mathbb{I}, \quad \nabla^{\mathrm{s}} \boldsymbol{u}_i := \frac{1}{2}(\nabla \boldsymbol{u}_i + \nabla \boldsymbol{u}_i^{\mathrm{T}}), \tag{2.2}$$

μ_i is the viscosity of the fluid i and $\mathbb{I}$ the identity tensor. Moreover, $[\![v]\!] := (v|_{\Omega_1})|_\Gamma - (v|_{\Omega_2})|_\Gamma$ denotes the jump of a piecewise smooth function v across Γ (the jump is defined component-

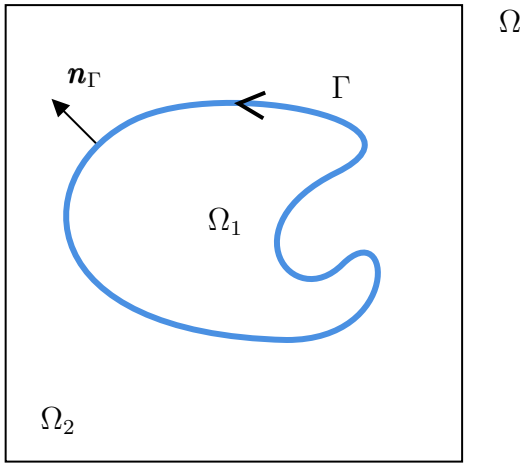


Figure 2.1: Computational domain Ω, subdomains Ω_i, $i \in \{1,2\}$, interface Γ, and unit normal $\boldsymbol{n}_\Gamma$.

wise for a vector- or tensor-valued field, and its sign is consistent with the orientation of the unit normal $\boldsymbol{n}_\Gamma$). To model the surface tension according to Laplace's law, we set

$$\boldsymbol{g}_N = \gamma H_\Gamma \boldsymbol{n}_\Gamma, \tag{2.3}$$

where γ is the surface tension and H_Γ the curvature on the interface.

The problem data are the body forces $\boldsymbol{f}_i \in L^2(\Omega_i;\mathbb{R}^d)$, $i \in \{1,2\}$, and the Dirichlet boundary condition $\boldsymbol{g} \in H^{\frac{1}{2}}(\partial\Omega;\mathbb{R}^d)$. The Stokes interface problem is well-posed if

$$\int_{\partial\Omega} \boldsymbol{g}\cdot\boldsymbol{n}_\Omega = 0 \tag{2.4}$$

(implied by the incompressibility condition), with $\boldsymbol{n}_\Omega$ the unit outward normal to Ω and if one requires $\sum_{i\in\{1,2\}}\int_{\Omega_i} p_i = 0$, thereby fixing the global additive constant on the pressure.

In the more general *interface equilibrium problem*, the interface Γ is part of the unknowns of the problem. In this setting, the Stokes interface problem is completed by requiring that the normal velocity at the interface is zero, and this condition essentially prescribes the shape of the interface. Thus, in the interface equilibrium problem, we seek the velocity and pressure fields $(\boldsymbol{u}_i, p_i) \in H^1(\Omega_i;\mathbb{R}^d)\times L^2(\Omega_i)$, $i \in \{1,2\}$, and the interface Γ such that

$$(2.1) \text{ holds true} \quad + \quad \boldsymbol{u}\cdot\boldsymbol{n}_\Gamma = 0 \text{ on } \Gamma, \tag{2.5}$$

where $\boldsymbol{u} := \boldsymbol{u}_1 = \boldsymbol{u}_2$ on Γ by (2.1d). Notice that equilibrium requires a zero normal velocity at the interface, whereas no conditions are imposed on the tangential velocity since it does not affect the shape of the interface.

It is convenient to describe the interface Γ as being the zero level-set of a function $\phi : \Omega \to \mathbb{R}$, i.e.,

$$\Gamma := \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) = 0\}, \tag{2.6}$$

and by convention, we assume that $\Omega_1 = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) < 0\}$ and $\Omega_2 = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) > 0\}$.

Assuming that ϕ is of class $\mathcal{C}^2$ in a neighborhood of Γ, we set

$$\boldsymbol{n}_\phi = \frac{\nabla\phi}{\|\nabla\phi\|_{\ell^2}}, \qquad H_\phi = -\nabla\cdot\boldsymbol{n}_\phi = -\frac{\Delta\phi}{\|\nabla\phi\|_{\ell^2}} + \frac{1}{\|\nabla\phi\|_{\ell^2}^3} D^2\phi(\nabla\phi,\nabla\phi), \tag{2.7}$$

where $\|\cdot\|_{\ell^2}$ denotes the Euclidean norm in $\mathbb{R}^d$ and $D^2\phi(\cdot,\cdot)$ is the quadratic form associated with the Hessian of ϕ, and then obtain the unit normal and the curvature of Γ by setting

$$\boldsymbol{n}_\Gamma := \boldsymbol{n}_\phi|_\Gamma, \qquad H_\Gamma := H_\phi|_\Gamma. \tag{2.8}$$

Notice that the curvature is negative everywhere on Γ if the set Ω_1 is convex.

2.3 Unfitted HHO solver with fixed interface

This section briefly describes the unfitted HHO solver for the Stokes interface problem (2.1). The interface Γ is kept fixed in this section.

2.3.1 Unfitted meshes

We consider a mesh $\mathcal{T}$ belonging to a shape-regular mesh sequence such that each mesh covers Ω exactly. We denote by T a generic mesh cell having diameter h_T and unit outward normal $\boldsymbol{n}_T$. The mesh faces are collected in the set $\mathcal{F}$, which is split as $\mathcal{F} = \mathcal{F}^o \cup \mathcal{F}^\partial$, where $\mathcal{F}^o$ (resp. $\mathcal{F}^\partial$) is the collection of the mesh internal faces (resp. boundary faces). For all $T\in\mathcal{T}$, the faces composing the boundary of T are collected in the set $\mathcal{F}_{\partial T} := \{F\in\mathcal{F} : F\subset\partial T\}$.

Since the mesh is unfitted, the interface Γ can cut arbitrarily through some of the mesh cells. Thus, we partition $\mathcal{T}$ into $\mathcal{T} = \mathcal{T}^1\cup\mathcal{T}^2\cup\mathcal{T}^\Gamma$ with

$$\mathcal{T}^i := \{T\in\mathcal{T} : T\subset\Omega_i\}, \ i\in\{1,2\}, \quad \mathcal{T}^\Gamma := \left\{T\in\mathcal{T} : T\cap\Gamma\neq\emptyset\right\}, \tag{2.9}$$

and we introduce the notation

$$T^i := T\cap\Omega_i, \quad (\partial T)^i := \partial T\cap(\Omega_i\cup\partial\Omega), \quad i\in\{1,2\}, \quad T^\Gamma := T\cap\Gamma, \tag{2.10}$$

so that $\partial T^i = (\partial T)^i\cup T^\Gamma$. See Figure 2.2 for an example. The mesh cells belonging to the set $\mathcal{T}^\Gamma$ are called cut cells, and those belonging to the set $\mathcal{T}^1\cup\mathcal{T}^2$ are called uncut cells. Consistently with the cell notation, we define $F^i := F\cap\Omega_i$ and $\mathcal{F}_{(\partial T)^i} := \left\{F^i : F\in\mathcal{F}_{\partial T}\right\}$. For simplicity, we assume that the interface does not intersect a vertex of a mesh cell and that $|\partial T\cap\Gamma| = 0$ for all $T\in\mathcal{T}^\Gamma$. These pathological situations, which can in principle be handled by specific geometrical coding procedures, were not encountered in our simulations.

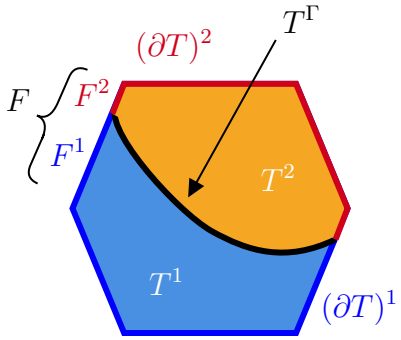


Figure 2.2: Decomposition of a cut cell $T \in \mathcal{T}^\Gamma$ and of its boundary ∂T.

2.3.2 HHO discretization

Let S be a subset of Ω of dimension $d' \in \{d-1, d\}$ (typically, S can be a mesh cell, a mesh face, or a collection thereof). For all $l \in \mathbb{N}$, we define $\mathbb{P}^l_{d'}(S)$ to be the space composed of d'-variate polynomials of total degree at most l restricted to S. Similarly, we use the notation $\mathbb{P}^l_{d'}(S;\mathbb{R}^d)$ and $\mathbb{P}^l_{d'}(S;\mathbb{R}^{d\times d}_{\rm sym})$ for the space composed of d'-variate $\mathbb{R}^d$-valued and $\mathbb{R}^{d\times d}_{\rm sym}$-valued polynomials of total degree at most l restricted to S, respectively, where $\mathbb{R}^{d\times d}_{\rm sym}$ denotes the space of symmetric matrices of order d. Moreover, $(\cdot,\cdot)_S$ and $\|\cdot\|_S$ denote, respectively, the $L^2(S)$-inner product and the associated norm with d'-dimensional Lebesgue measure.

To discretize (2.1), we introduce a local HHO(k) space of order $k \geq 0$ for every mesh cell $T \in \mathcal{T}$. If the mesh cell $T \in \mathcal{T}$ is not cut by the interface Γ, the discrete velocity unknowns in T are a vector-valued polynomial of degree $(k+1)$ in the cell T and a vector-valued polynomial of degree k on each face $F \in \mathcal{F}_{\partial T}$, whereas the discrete pressure is a polynomial of order k in the cell T. Thus, the local HHO unknowns for the velocity and the pressure are

$$\widehat{\boldsymbol{v}}_T := (\boldsymbol{v}_T, \boldsymbol{v}_{\partial T}) \in \widehat{\boldsymbol{U}}^k_T := \mathbb{P}^{k+1}_d(T;\mathbb{R}^d) \times \mathbb{P}^k_{d-1}(\mathcal{F}_{\partial T};\mathbb{R}^d), \tag{2.11a}$$

$$p_T \in P^k_T := \mathbb{P}^k_d(T), \tag{2.11b}$$

with $\mathbb{P}^k_{d-1}(\mathcal{F}_{\partial T};\mathbb{R}^d) := \bigtimes_{F\in\mathcal{F}_{\partial T}} \mathbb{P}^k_{d-1}(F;\mathbb{R}^d)$. If, instead, the mesh cell $T \in \mathcal{T}$ is cut by the interface Γ, the idea is to double the unknowns in the cut cell and on its cut faces without attaching any unknowns to the interface. The local HHO unknowns for the velocity and the pressure are then

$$\widehat{\boldsymbol{v}}_T := (\boldsymbol{v}_{T^1}, \boldsymbol{v}_{T^2}, \boldsymbol{v}_{(\partial T)^1}, \boldsymbol{v}_{(\partial T)^2}) \in \widehat{\boldsymbol{U}}^k_T, \tag{2.12a}$$

$$p_T := (p_{T^1}, p_{T^2}) \in P^k_T := \mathbb{P}^k_d(T^1) \times \mathbb{P}^k_d(T^2), \tag{2.12b}$$

with

$$\widehat{\boldsymbol{U}}^k_T := \mathbb{P}^{k+1}_d(T^1;\mathbb{R}^d) \times \mathbb{P}^{k+1}_d(T^2;\mathbb{R}^d) \times \mathbb{P}^k_{d-1}(\mathcal{F}_{(\partial T)^1};\mathbb{R}^d) \times \mathbb{P}^k_{d-1}(\mathcal{F}_{(\partial T)^2};\mathbb{R}^d), \tag{2.13}$$

and $\mathbb{P}^k_{d-1}(\mathcal{F}_{(\partial T)^i};\mathbb{R}^d) := \bigtimes_{F\in\mathcal{F}_{(\partial T)^i}} \mathbb{P}^k_{d-1}(F;\mathbb{R}^d)$ for all $i \in \{1,2\}$. See Figure 2.3 for a representation of the discrete velocity unknowns in a cut and an uncut cell of hexagonal shape. To handle uncut and cut cells in a single formalism, we set for every uncut cell $T \in \mathcal{T}^i$ with

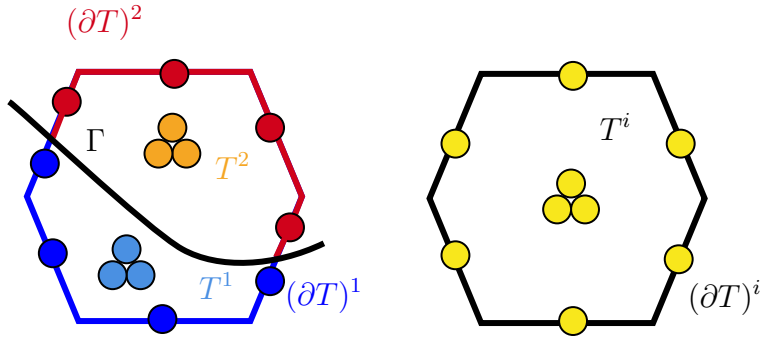


Figure 2.3: Discrete velocity unknowns ($k = 0$) for a cut (left) and uncut (right) cell of hexagonal shape. Notice the doubling of the cell unknowns in the cut cell and of the face unknowns on the cut faces. Conventionally, each bullet represents one $\mathbb{R}^2$-valued degree of freedom.

$i \in \{1,2\}$,

$$T^i := T, \quad T^{\overline{i}} := \emptyset, \quad (\partial T)^{\overline{i}} := \emptyset, \quad T^\Gamma := \emptyset, \tag{2.14}$$

where $\overline{i} := 3 - i$ (so that $\overline{1} := 2$ and $\overline{2} := 1$). We use a similar convention for the mesh faces.

Inspired by [20], [22], we define, for every mesh cell $T \in \mathcal{T}$ and all $i \in \{1,2\}$, the symmetric gradient reconstruction operators $\mathbb{E}^k_{T^i} : \widehat{\boldsymbol{U}}^k_T \to \mathbb{P}^k_d(T^i;\mathbb{R}^{d\times d}_{\rm sym})$, $i \in \{1,2\}$, such that for all $\widehat{\boldsymbol{v}}_T \in \widehat{\boldsymbol{U}}^k_T$,

$$(\mathbb{E}^k_{T^i}(\widehat{\boldsymbol{v}}_T), \mathfrak{q})_{T^i} := (\nabla^{\rm s}\boldsymbol{v}_{T^i}, \mathfrak{q})_{T^i} + (\boldsymbol{v}_{(\partial T)^i} - \boldsymbol{v}_{T^i}, \mathfrak{q}\boldsymbol{n}_T)_{(\partial T)^i} - \alpha_i([\![\boldsymbol{v}_T]\!], \mathfrak{q}\boldsymbol{n}_\Gamma)_{T^\Gamma}, \tag{2.15}$$

for all $\mathfrak{q} \in \mathbb{P}^k_d(T^i;\mathbb{R}^{d\times d}_{\rm sym})$. In the same spirit as in [20] for elliptic problems, robustness with respect to the contrast in the viscosity coefficients can be obtained defining

$$\alpha_i := \frac{\mu_{\overline{i}}}{\mu_i + \mu_{\overline{i}}}, \tag{2.16}$$

so that $\alpha_1 = \alpha_2 = 0.5$ if $\mu_1 = \mu_2$, whereas $\alpha_1 \approx 0$, $\alpha_2 \approx 1$ if $\mu_1 \gg \mu_2$ and vice versa if $\mu_2 \gg \mu_1$. Similarly, the divergence reconstruction operator $D^k_{T^i} : \widehat{\boldsymbol{U}}^k_T \to \mathbb{P}^k_d(T^i)$ is such that, for all $\widehat{\boldsymbol{v}}_T \in \widehat{\boldsymbol{U}}^k_T$,

$$D^k_{T^i}(\widehat{\boldsymbol{v}}_T) := \mathrm{trace}(\mathbb{E}^k_{T^i}(\widehat{\boldsymbol{v}}_T)). \tag{2.17}$$

Then, the local HHO bilinear and linear forms are defined as follows: For all $(\widehat{\boldsymbol{v}}_T, r_T), (\widehat{\boldsymbol{w}}_T, q_T) \in \widehat{\boldsymbol{U}}^k_T \times P^k_T$,

$$A_T((\widehat{\boldsymbol{v}}_T, r_T), (\widehat{\boldsymbol{w}}_T, q_T)) := a_T(\widehat{\boldsymbol{v}}_T, \widehat{\boldsymbol{w}}_T) - b_T(\widehat{\boldsymbol{w}}_T, r_T) + b_T(\widehat{\boldsymbol{v}}_T, q_T), \tag{2.18a}$$

$$l_T(\widehat{\boldsymbol{w}}_T) := \sum_{i\in\{1,2\}} \Big\{(\boldsymbol{f}, \boldsymbol{w}_{T^i})_{T^i} + \alpha_{\overline{i}}(\boldsymbol{g}_N, \boldsymbol{w}_{T^i})_{T^\Gamma}\Big\}, \tag{2.18b}$$

with the bilinear forms

$$a_T(\widehat{\boldsymbol{v}}_T,\widehat{\boldsymbol{w}}_T) := \sum_{i\in\{1,2\}} 2\mu_i(\mathbb{E}^k_{T^i}(\widehat{\boldsymbol{v}}_T),\mathbb{E}^k_{T^i}(\widehat{\boldsymbol{w}}_T))_{T^i} + s^\Gamma_T(\widehat{\boldsymbol{v}}_T,\widehat{\boldsymbol{w}}_T) + s^{1,2}_T(\widehat{\boldsymbol{v}}_T,\widehat{\boldsymbol{w}}_T), \tag{2.19a}$$

$$b_T(\widehat{\boldsymbol{w}}_T,r_T) := \sum_{i\in\{1,2\}} (r_{T^i}, D^k_{T^i}(\widehat{\boldsymbol{w}}_T))_{T^i}, \tag{2.19b}$$

and the following stabilization bilinear forms:

$$s^\Gamma_T(\widehat{\boldsymbol{v}}_T,\widehat{\boldsymbol{w}}_T) := \mu_\# h_T^{-1}([\![\boldsymbol{v}_T]\!],[\![\boldsymbol{w}_T]\!])_{T^\Gamma}, \qquad \mu_\# := \min(\mu_1,\mu_2), \tag{2.20a}$$

$$s^{1,2}_T(\widehat{\boldsymbol{v}}_T,\widehat{\boldsymbol{w}}_T) := \sum_{i\in\{1,2\}} \mu_i h_T^{-1}(\Pi^k_{(\partial T)^i}(\boldsymbol{v}_{T^i}) - \boldsymbol{v}_{(\partial T)^i}, \Pi^k_{(\partial T)^i}(\boldsymbol{w}_{T^i}) - \boldsymbol{w}_{(\partial T)^i})_{(\partial T)^i}. \tag{2.20b}$$

The operator $\Pi^k_{(\partial T)^i}$ denotes the L^2-orthogonal projection onto $\mathbb{P}^k_{d-1}(\mathcal{F}_{(\partial T)^i};\mathbb{R}^d)$. Notice that an additional stabilization (with a small enough parameter) is considered in [22] for theoretical reasons, but the numerical results therein indicate that this stabilization can be omitted. We do not consider it in this work.

Passing to the global setting, we define for all $i\in\{1,2\}$,

$$\boldsymbol{U}^{k+1}_{\mathcal{T}^i} := \bigtimes_{T\in\mathcal{T}} \mathbb{P}^{k+1}_d(T^i;\mathbb{R}^d), \qquad \boldsymbol{U}^k_{\mathcal{F}^i} := \bigtimes_{F\in\mathcal{F}} \mathbb{P}^k_{d-1}(F^i;\mathbb{R}^d), \tag{2.21a}$$

$$P^k_{\mathcal{T}^i} := \bigtimes_{T\in\mathcal{T}} \mathbb{P}^k_d(T^i). \tag{2.21b}$$

We define the following spaces for the global HHO unknowns:

$$\widehat{\boldsymbol{U}}^k_{\mathcal{T}} := \boldsymbol{U}^{k+1}_{\mathcal{T}^1}\times\boldsymbol{U}^{k+1}_{\mathcal{T}^2}\times\boldsymbol{U}^k_{\mathcal{F}^1}\times\boldsymbol{U}^k_{\mathcal{F}^2}, \qquad P^k_{\mathcal{T}} := P^k_{\mathcal{T}^1}\times P^k_{\mathcal{T}^2}. \tag{2.22}$$

For all $\widehat{\boldsymbol{v}}_{\mathcal{T}}\in\widehat{\boldsymbol{U}}^k_{\mathcal{T}}$, we write $\widehat{\boldsymbol{v}}_{\mathcal{T}} = (\boldsymbol{v}_{\mathcal{T}^1},\boldsymbol{v}_{\mathcal{T}^2},\boldsymbol{v}_{\mathcal{F}^1},\boldsymbol{v}_{\mathcal{F}^2})$ and for all $q_{\mathcal{T}}\in P^k_{\mathcal{T}}$, we write $q_{\mathcal{T}} = (q_{\mathcal{T}^1},q_{\mathcal{T}^2})$. Moreover, for all $T\in\mathcal{T}$, we denote by

$$\widehat{\boldsymbol{v}}_T := (\boldsymbol{v}_{T^1},\boldsymbol{v}_{T^2},\boldsymbol{v}_{(\partial T)^1},\boldsymbol{v}_{(\partial T)^2})\in\widehat{\boldsymbol{U}}^k_T, \quad q_T = (q_{T^1},q_{T^2})\in P^k_T, \tag{2.23}$$

the local components of $\widehat{\boldsymbol{v}}_{\mathcal{T}}$ and $q_{\mathcal{T}}$, respectively, attached to the mesh cell T and its faces. We denote by $\widehat{\boldsymbol{U}}^k_{\mathcal{T}0}$ the subspace of $\widehat{\boldsymbol{U}}^k_{\mathcal{T}}$ where all the velocity components attached to the boundary faces composing $\partial\Omega$ are null and by $P^k_{\mathcal{T}*}$ the subspace of $P^k_{\mathcal{T}}$ composed of functions with zero average over Ω. Finally, defining the global bilinear and linear forms

$$A_{\mathcal{T}}((\widehat{\boldsymbol{v}}_{\mathcal{T}},r_{\mathcal{T}}),(\widehat{\boldsymbol{w}}_{\mathcal{T}},q_{\mathcal{T}})) := \sum_{T\in\mathcal{T}} A_T((\widehat{\boldsymbol{v}}_T,r_T),(\widehat{\boldsymbol{w}}_T,q_T)), \tag{2.24a}$$

$$L_{\mathcal{T}}(\widehat{\boldsymbol{w}}_{\mathcal{T}}) := \sum_{T\in\mathcal{T}} l_T(\widehat{\boldsymbol{w}}_T), \tag{2.24b}$$

the discrete problem amounts to seeking $(\widehat{\boldsymbol{u}}_{\mathcal{T}},p_{\mathcal{T}})\in\widehat{\boldsymbol{U}}^k_{\mathcal{T}0}\times P^k_{\mathcal{T}*}$ such that

$$A_{\mathcal{T}}((\widehat{\boldsymbol{u}}_{\mathcal{T}},p_{\mathcal{T}}),(\widehat{\boldsymbol{w}}_{\mathcal{T}},q_{\mathcal{T}})) = L_{\mathcal{T}}(\widehat{\boldsymbol{w}}_{\mathcal{T}}), \qquad \forall(\widehat{\boldsymbol{w}}_{\mathcal{T}},q_{\mathcal{T}})\in\widehat{\boldsymbol{U}}^k_{\mathcal{T}0}\times P^k_{\mathcal{T}*}. \tag{2.25}$$

The error analysis for the discrete problem (2.25) is performed in [22] by establishing inf-sup

stability, consistency, and approximation properties. (Therein, the parameters α_i are such that $\alpha_i = 0$ if $\mu_i \le \mu_{\bar{\imath}}$, but the adaptation of the analysis to the parameters α_i prescribed as in (2.16) is straightforward.) A key tool for the analysis is to ensure, by means of the cell agglomeration procedure described in [20, Section 4.3], that any cut cell $T \in \mathcal{T}^\Gamma$ satisfies $\min_{i\in\{1,2\}} |T^i| \ge \varrho|T|$ for a given user-parameter $\varrho \in (0, \frac{1}{2})$ (hereafter, we use $\varrho := 0.3$ consistently with [20]). Then, assuming that $(\boldsymbol{u}_i, p_i) \in H^{k+2}(\Omega_i;\mathbb{R}^d) \times H^{k+1}(\Omega_i)$, $i \in \{1,2\}$, [22, Theorem 12] states that there is a constant C, independent of the mesh-size $h := \max_{T\in\mathcal{T}} h_T$ and of the fluid viscosities μ_i, $i \in \{1,2\}$, such that

$$\begin{aligned}&\left\{\sum_{T\in\mathcal{T}}\sum_{i\in\{1,2\}} \mu_i\|\nabla^{\mathrm{s}}(\boldsymbol{u}_i - \boldsymbol{u}_{T^i})\|_{T^i}^2 + \mu_i^{-1}\|p_i - p_{T^i}\|_{T^i}^2\right\}^{\frac{1}{2}}\\ &\le Ch^{k+1}\left\{\sum_{i\in\{1,2\}} \mu_i|\boldsymbol{u}_i|_{H^{k+2}(\Omega_i;\mathbb{R}^d)}^2 + \mu_i^{-1}\|p_i\|_{H^{k+1}(\Omega_i)}^2\right\}^{\frac{1}{2}}.\end{aligned} \tag{2.26}$$

2.3.3 Interface discretization and quadratures in the cut cells

An important novelty of the present Thesis with respect to [20], [22] is that the interface Γ is no longer considered to be analytically known. Instead, it is known here only through a discrete approximation $\phi_\mathcal{T}$ of the level-set function ϕ introduced in (2.6). Thus, we set

$$\Gamma_\mathcal{T} := \{\boldsymbol{x} \in \Omega : \phi_\mathcal{T}(\boldsymbol{x}) = 0\}. \tag{2.27}$$

The construction of the discrete approximation $\phi_\mathcal{T}$ is discussed in Section 3.2.

In practice, the interface $\Gamma_\mathcal{T}$ is approximately represented by a collection of arcs, each arc being generated from the reference interval $\widehat{I} := [0,1]$ by using a geometric mapping whose components are polynomial-valued. Specifically, for every cut cell $T \in \mathcal{T}^\Gamma$, $\Gamma_\mathcal{T} \cap T$ is approximated by a collection of 2^n arcs, where $n \ge 0$ is a user-specified parameter. These arcs are denoted by $\Upsilon^{T,j}$, $j \in \{1{:}2^n\}$. To build each arc $\Upsilon^{T,j}$, we consider $(l+1)$ equidistributed interpolation nodes $\{\boldsymbol{x}_m^{T,j}\}_{m\in\{0:l\}} \subset \Gamma_\mathcal{T}$. The construction of these nodes is discussed below. The integer l is a user-specified parameter such that $l \ge k+1$, where k is the degree of the face unknowns in the HHO method. This lower bound is coherent with the consistency analysis of elliptic problems on curved domains, where the geometry error is at least one order higher than the approximation error in the H^1-norm (which is of order $(k+1)$ for the HHO method).

Considering the Lagrange basis of order l, $\{\widehat{\psi}_m\}_{m\in\{0:l\}}$, defined on the reference interval $\widehat{I}$ using equidistributed nodes, the arc $\Upsilon^{T,j}$ is defined as

$$\Upsilon^{T,j} := \boldsymbol{r}^{T,j}(\widehat{I}) \qquad \text{where} \qquad \boldsymbol{r}^{T,j}(\widehat{s}) := \sum_{m\in\{0:l\}} \boldsymbol{x}_m^{T,j}\widehat{\psi}_m(\widehat{s}), \quad \forall \widehat{s} \in \widehat{I}. \tag{2.28}$$

Finally, the fully discrete interface is defined as follows:

$$\Gamma_\mathcal{T}^{n,l} := \bigcup_{T\in\mathcal{T}^\Gamma}\bigcup_{j\in\{1:2^n\}} \Upsilon^{T,j}. \tag{2.29}$$

We notice that the fully discrete interface has only C^0-regularity. An interesting perspective,

left to future work, is to study whether a smooth discrete interface, e.g., built using splines to achieve C^1-regularity, can have a sizable effect in improving solution accuracy.

To construct the interpolation nodes $\{\boldsymbol{x}_m^{T,j}\}_{m\in\{0:l\}}$ for all $T\in\mathcal{T}^\Gamma$ and all $j\in\{1{:}2^n\}$, we assume for simplicity that the interface Γ intersects ∂T at two points only and that these points are located on two distinct edges of T. Let us denote by $\boldsymbol{a}_0$ and $\boldsymbol{a}_{2^nl}$ the two intersection points, ordered according to the orientation of Γ (see Figure 2.1). We proceed as follows (see Figure 2.4):

1. Construction of the points $\{\boldsymbol{a}_{jl}\}_{j\in\{0:2^n\}}$, all lying on $\Gamma_\mathcal{T}$. If $n=0$, there is nothing to do. Otherwise, $n\geq 1$ and the points $\{\boldsymbol{a}_{jl}\}_{j\in\{0:2^n\}}$ are constructed recursively. Let $m\in\{1{:}n\}$ and assume that the points $\{\boldsymbol{a}_{k2^{n-m+1}l}\}_{k\in\{0:2^{m-1}\}}$ are available (all on $\Gamma_\mathcal{T}$). Then we construct the points $\{\boldsymbol{a}_{(2k+1)2^{n-m}l}\}_{k\in\{0:2^{m-1}-1\}}$ as follows. For all $k\in\{0:2^{m-1}\}$, we let $\overline{\boldsymbol{x}}_k$ be the midpoint of the segment $[\boldsymbol{a}_{k2^{n-m+1}l},\boldsymbol{a}_{(k+1)2^{n-m+1}l}]:=\{(1-\theta)\boldsymbol{a}_{k2^{n-m+1}l}+\theta\boldsymbol{a}_{(k+1)2^{n-m+1}l},\theta\in[0,1]\}$, and L_k be the line passing through $\overline{\boldsymbol{x}}_k$ and orthogonal to this segment. We define $\boldsymbol{a}_{(2k+1)2^{n-m}l}$ to be the closest point in L_k to $\overline{\boldsymbol{x}}_k$ such that $\phi_\mathcal{T}(\boldsymbol{a}_{(2k+1)2^{n-m}l})=0$, i.e., $\boldsymbol{a}_{(2k+1)2^{n-m}l}\in\Gamma_\mathcal{T}$. This point is found by dichotomy.

2. For all $j\in\{0:2^n-1\}$, we now construct the points $\{\boldsymbol{a}_{jl+m}\}_{m\in\{1:l-1\}}$ as follows. We consider the segment $[\boldsymbol{a}_{jl},\boldsymbol{a}_{(j+1)l}]:=\{(1-\theta)\boldsymbol{a}_{jl}+\theta\boldsymbol{a}_{(j+1)l},\theta\in[0,1]\}$. For all $m\in\{1{:}l-1\}$, we set $\overline{\boldsymbol{x}}_m:=(1-\theta_m)\boldsymbol{a}_{jl}+\theta_m\boldsymbol{a}_{(j+1)l}$ with $\theta_m:=m/l$, we let L_m be the line passing through $\overline{\boldsymbol{x}}_m$ and orthogonal to this segment, and we define $\boldsymbol{a}_{jl+m}$ to be the closest point in L_m to $\overline{\boldsymbol{x}}_m$ such that $\phi_\mathcal{T}(\boldsymbol{a}_{jl+m})=0$, i.e., $\boldsymbol{a}_{jl+m}\in\Gamma_\mathcal{T}$. This point is found by dichotomy.

3. We have now built the collection of points $\{\boldsymbol{a}_j\}_{j\in\{0:2^nl\}}$, all in $\Gamma_\mathcal{T}$. Finally, we set

$$\boldsymbol{x}_m^{T,j}:=\boldsymbol{a}_{(j-1)l+m},\quad j\in\{1{:}2^n\},\ m\in\{0{:}l\}. \tag{2.30}$$

   Notice that $\boldsymbol{x}_l^{T,j}=\boldsymbol{x}_0^{T,j+1}$ by construction, ensuring the matching of the endpoint of each arc with the starting point of the next arc.

To realize the unfitted HHO method, high-order quadratures need to be performed along the fully discrete interface $\Gamma_\mathcal{T}^{n,l}$ and in the cut cells (for the uncut cells, the procedure is straightforward). One-dimensional quadratures along $\Gamma_\mathcal{T}^{n,l}$ are straightforward to implement by using the above geometric mappings. Moreover, to implement quadratures in the cut cells, we decompose T^i, $i\in\{1,2\}$, into curved subtriangles which are mapped into a reference triangle where quadratures based on Dunavant points are performed. The construction of geometric nonaffine mappings based on sets of geometric nodes is described, e.g., in [51, §8.1]. The whole procedure in a cut cell is illustrated in Figure 2.5. We refer the reader to [58], [59], [114] for further insight. We emphasize that substantial computational savings are achieved compared to [22] where quadratures are realized using a very large number of flat triangles ($n\gg 1$, $l=1$).

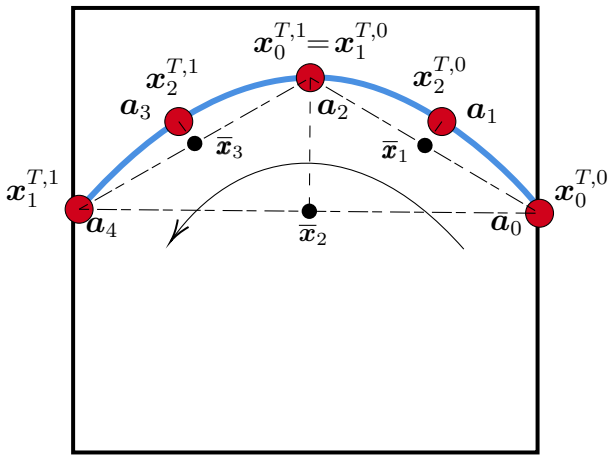
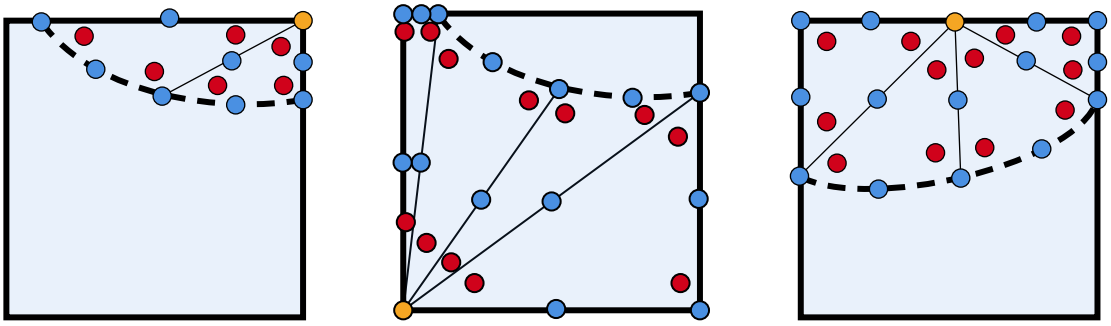


Figure 2.4: Example of a square cut cell $T \in \mathcal{T}^\Gamma$ with interface (blue line) and interface points (red bullets) $\{\boldsymbol{x}_m^{T,j}\}_{m\in\{0:l\}}$, $j \in \{1{:}2^n\}$. The interface discretization parameters are ($l = 2$, $n = 0$). The construction of the points $\boldsymbol{a}_m$ and $\overline{\boldsymbol{x}}_m$, for all $m \in \{0{:}2^n l\}$, is shown. The arrow indicates the interface orientation.



(a) The interface cuts two adjacent faces of T and leaves one vertex in T^i.

(b) The interface cuts two adjacent faces of T and leaves three vertices in T^i.

(c) The interface cuts two opposite faces of T.

Figure 2.5: Example of cut cells $T \in \mathcal{T}^\Gamma$ and of their decomposition into curved subtriangles. The subtriangulation is performed by connecting the interface points $\{\boldsymbol{x}_m^{T,j}\}_{m\in\{0:2\}}$, $j \in \{1{:}2^n\}$ to a specific point (orange bullet, a vertex or an edge midpoint according to the cut configuration). The interface discretization parameters are ($l = 2$, $n = 1$). The blue bullets, plus the orange bullet, are the physical points used for defining the mapping from the reference triangle into the physical one (quadratic mapping in the present example). Moreover, the red bullets represent the mapped integration Dunavant points for the second-order quadrature.

Chapter 3

HHO for immiscible Stokes fluids

3.1 Equilibrium with a pure shear flow

In this section, the interface is kept fixed, i.e., we solve the Stokes interface problem (2.1). Specifically we study the shear-surface tension equilibrium problem in the absence of body forces, i.e., $\boldsymbol{f}_i := \boldsymbol{0}$, $i \in \{1,2\}$. Let the box $\Omega := (-a,a)^2$, $a > 0$, be the computational domain. We consider the following non-homogeneous Dirichlet condition enforcing a pure shear flow on $\partial\Omega$:

$$\boldsymbol{g} := \boldsymbol{u}_{\varepsilon|\partial\Omega} \qquad \text{where} \qquad \boldsymbol{u}_\varepsilon(x,y) := \varepsilon(x,-y)^{\mathrm{T}}, \tag{3.1}$$

with the shear parameter $\varepsilon > 0$. In the limit $a \to \infty$, the interface is elliptic of radii $0 < R_1 \leq R_2$ [39], [124]. It is convenient to define the Taylor deformation parameter

$$D := \frac{R_2 - R_1}{R_1 + R_2}. \tag{3.2}$$

We also define the capillary number

$$Ca := \mu_2 \frac{\varepsilon L_*}{\gamma}, \tag{3.3}$$

where the shear parameter ε is prescribed by (3.1) and, consistently with Taylor's study dealing with the deformation of an initial circular droplet, the reference length scale L_* is defined as

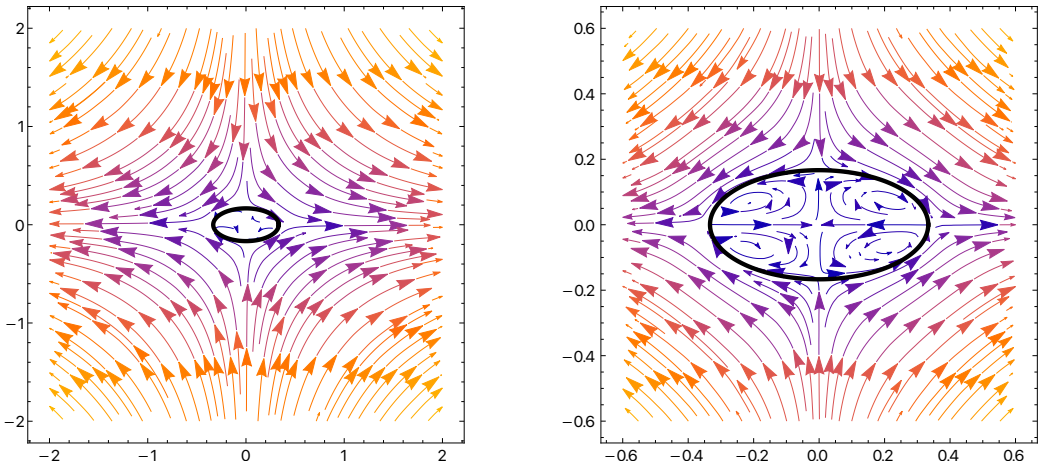
$$L_* := 2R_* \qquad \text{with} \qquad R_* := \sqrt{\frac{|\Omega_1|}{\pi}} = \sqrt{R_1 R_2}. \tag{3.4}$$

Putting everything together, the capillary number reads

$$Ca := 2\mu_2 \frac{\varepsilon\sqrt{R_1 R_2}}{\gamma}. \tag{3.5}$$

Notice that one needs to prescribe one additional length scale to uniquely solve the problem; for instance, one can prescribe the reference length scale L_* or the radius of the inner subdomain R_1.

In the limit case where $\varepsilon = 0$ (i.e., null shear), the equilibrium interface has a circular form, i.e., $D = 0$ meaning that $R_1 = R_2$. Whenever $\varepsilon > 0$, the equilibrium interface has an elliptic shape: the shear flow stretches the interface along the x-axis and compresses it along the y-axis. Figure 3.1 shows the streamlines of the equilibrium velocity field with shear $\varepsilon = 0.59$ and radii $R_1 = 1/6$ and $R_2 = 1/3$ (so that $D = 1/3$ and $R_* = 1/(3\sqrt{2})$).



(a) Streamlines in the domain $\Omega = (-2, 2)^2$.

(b) Zoom in $(-0.6, 0.6)^2$.

Figure 3.1: Streamlines of the velocity field $\boldsymbol{u}$ obtained by solving (2.5) with $\gamma = 1$, $\mu_1 = \mu_2 = 1$ in the domain $\Omega = (-2, 2)^2$, employing the unfitted HHO discretization with polynomial order $k = 1$. The shear parameter is $\varepsilon = 0.59$. The domain is discretized by a mesh composed of 128×128 square cells. The ellipse has radii $R_1 = 1/6$, $R_2 = 1/3$.

Our goal is to study numerically the capillary number Ca leading to equilibrium. In the first part of our study, we fix $\mu_1 = \mu_2 = 1$, and we fix an elliptic interface having radii R_1 and R_2, thus prescribing the value of the Taylor deformation parameter D. We want to find numerically the ratio $m_{\text{eq}} := \frac{\varepsilon_{\text{eq}}}{\gamma_{\text{eq}}}$ leading to equilibrium. Owing to the linearity of the Stokes interface problem (2.1), the velocity field $\boldsymbol{u}(\varepsilon, \gamma)$ depends linearly on ε and γ, so that we have

$$\boldsymbol{u}(\varepsilon, \gamma) = \boldsymbol{u}(\varepsilon, 0) + \boldsymbol{u}(0, \gamma), \tag{3.6}$$

where $\boldsymbol{u}(\varepsilon, 0)$ depends linearly on ε and $\boldsymbol{u}(0, \gamma)$ depends linearly on γ. Notice in passing that $\boldsymbol{u}(\varepsilon, 0)$ differs from $\boldsymbol{u}_\varepsilon$ when the viscosities of the two fluids are not equal. The key observation is that, since the equilibrium of the interface is achieved when the normal velocity at the interface is null, we must have

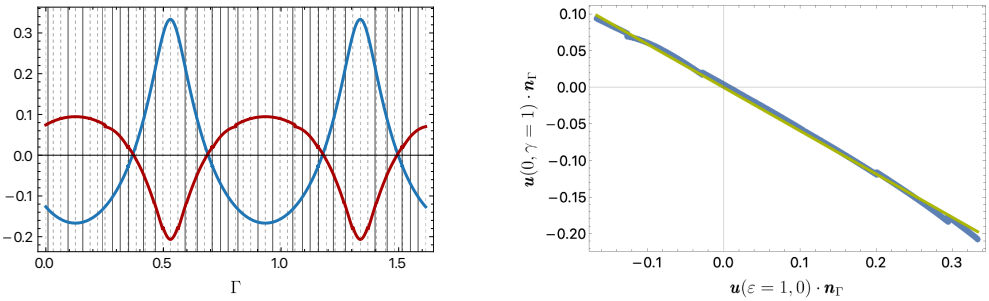
$$\frac{\boldsymbol{u}(\varepsilon_{\text{eq}}, 0) \cdot \boldsymbol{n}_\Gamma}{\boldsymbol{u}(0, \gamma_{\text{eq}}) \cdot \boldsymbol{n}_\Gamma} = -1 \qquad \text{a.e. on } \Gamma, \tag{3.7}$$

up to the few points on Γ where both numerator and denominator vanish. Therefore, the value of m_{eq} can be found by computing the two reference velocity fields $\boldsymbol{u}(0, \gamma = 1)$ and $\boldsymbol{u}(\varepsilon = 1, 0)$. These two reference velocity fields are such that their normal component at the interface are

linearly correlated, i.e., we have

$$\frac{\boldsymbol{u}(0,\gamma=1)\cdot\boldsymbol{n}_\Gamma}{\boldsymbol{u}(\varepsilon=1,0)\cdot\boldsymbol{n}_\Gamma} = -m_{\mathrm{eq}} \qquad \text{a.e. on } \Gamma. \tag{3.8}$$

By determining the value of m_{eq}, one readily deduces from (3.3) the value of the capillary number Ca associated with the prescribed value of the Taylor deformation parameter D.



(a) $\boldsymbol{u}(\varepsilon=1,0)\cdot\boldsymbol{n}_\Gamma$ (blue) and $\boldsymbol{u}(0,\gamma=1)\cdot\boldsymbol{n}_\Gamma$ (red) along the interface.

(b) $\boldsymbol{u}(0,\gamma=1)\cdot\boldsymbol{n}_\Gamma$ w.r.t. $\boldsymbol{u}(\varepsilon=1,0)\cdot\boldsymbol{n}_\Gamma$.

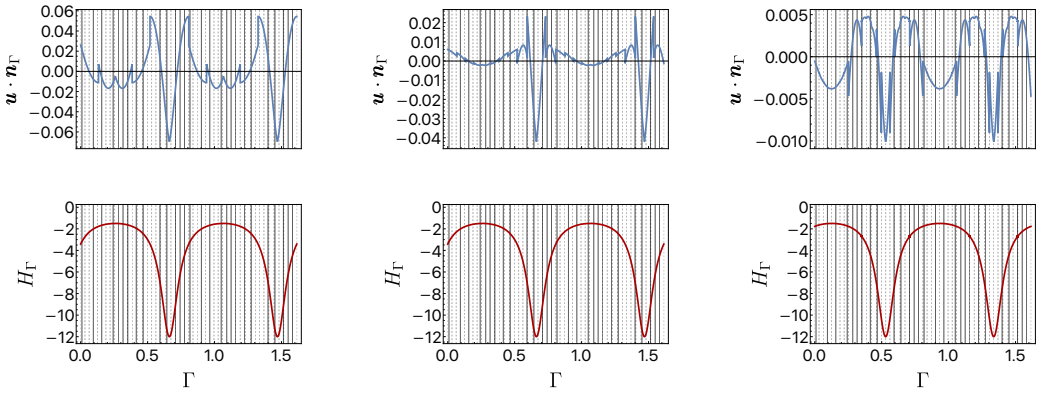
Figure 3.2: Figure 3.2a: normal velocities along the interface; the dashed vertical lines in the background represent the intersections between the interface and the cell faces, whereas the solid lines represent the intersections between the interface and the erased inner faces of the agglomerated cells. Figure 3.2b: $\boldsymbol{u}(0,\gamma=1)\cdot\boldsymbol{n}_\Gamma$ versus $\boldsymbol{u}(\varepsilon=1,0)\cdot\boldsymbol{n}_\Gamma$ (blue bullets); the green line represents the linear regression curve with slope $-m_{\mathrm{eq}} \approx 0.590$. Elliptic interface with $D = 1/3$ and $\mu_1 = \mu_2 = 1$. The velocity fields $\boldsymbol{u}(\varepsilon=1,0)$ and $\boldsymbol{u}(0,\gamma=1)$ are computed in the domain $\Omega = (-2,2)^2$ discretized by a mesh composed of 128×128 square cells. We employ the unfitted HHO discretization with polynomial order $k = 1$.

Let us exemplify the procedure for the elliptic interface with radii $R_1 = 1/6$ and $R_2 = 1/3$ (so that $D = 1/3$ and $R_* = 1/(3\sqrt{2})$). Recall that $\mu_1 = \mu_2 = 1$. The computational domain is set to $\Omega = (-2,2)^2$ and is initially discretized by 128×128 square cells. After using the cell-agglomeration procedure from [20], the mesh is composed of 16352 (instead of 16384) cells. We set the interface discretization parameters to $(l = 4,\ n = 0)$. Figure 3.2a shows the normal velocities $\boldsymbol{u}(\varepsilon=1,0)\cdot\boldsymbol{n}_\Gamma$ and $\boldsymbol{u}(0,\gamma=1)\cdot\boldsymbol{n}_\Gamma$ along the interface Γ. Interestingly, the contribution of the jump of the viscous stress tensor tends to vanish at the interface, so that the pressure jump absorbs all the surface tension force. As Figure 3.2b shows, the linear correlation between both normal components is rather well established numerically. The slope of the linear regression curve is $-m_{\mathrm{eq}} \approx -0.590$. Therefore, in the case $D = 1/3$, $\mu_1 = \mu_2 = 1$, $\gamma = 1$, we conclude that the equilibrium is obtained for $\varepsilon_{\mathrm{eq}} = -m_{\mathrm{eq}}$, i.e.,

$$Ca = 2\varepsilon_{\mathrm{eq}}\sqrt{R_1 R_2} \approx 0.278. \tag{3.9}$$

For this value, the peak normal flow velocity computed numerically at the interface is 1×10^{-2} (which can be considered to be reasonably close to zero). Figure 3.3 displays the normal velocity and the curvature along the interface for three meshes composed of 32×32, 64×64 and 128×128 square cells. Notice that peaks on normal velocities coincide with maximal curvature in absolute value. Using mesh refinement, the maximal normal velocity at the interface is reduced to 7×10^{-2}, 3×10^{-2} and 1×10^{-2}, respectively. Moreover, the three meshes lead

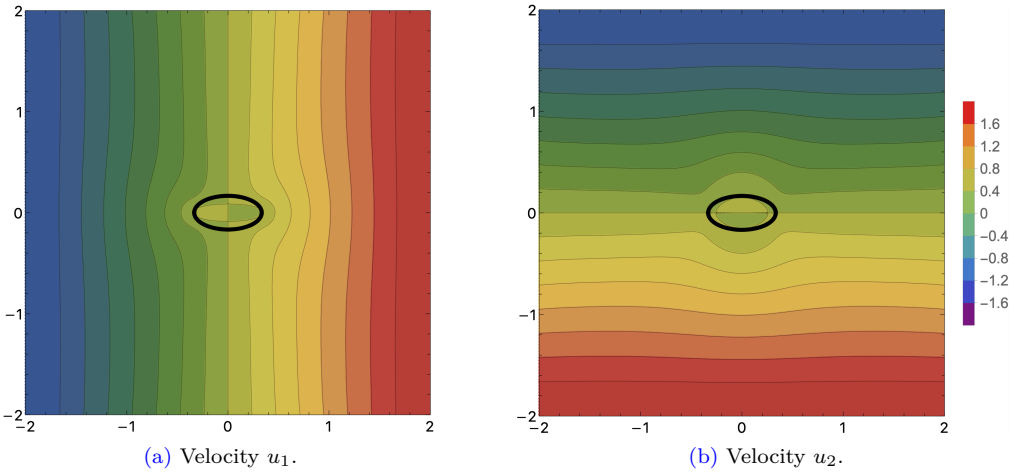
to the values $\varepsilon_{\text{eq}} \in \{0.654, 0.581, 0.590\}$ and $Ca \in \{0.308, 0.274, 0.278\}$, indicating satisfactory convergence on the finest mesh. Notice that the interface discretization parameters (l, n) have a marginal influence on the accuracy of the numerical predictions. The only differences were observed for $(l = 1, n = 1)$ giving $Ca = 0.280$ and $(l = 1, n = 2)$ giving $Ca = 0.279$. All the other choices for $(l, n) \in \{1{:}6\} \times \{1{:}4\}$ give the same value for Ca as above. Figure 3.4 displays



(a) Mesh 32×32. (b) Mesh 64×64. (c) Mesh 128×128.

Figure 3.3: Normal velocity (blue, top row) and curvature (red, bottom row) along the interface. Shear-surface tension equilibrium with elliptic interface for $D = 1/3$, $\mu_1 = \mu_2 = 1$, $Ca \in \{0.308, 0.274, 0.278\}$ using meshes composed of 32×32, 64×64 and 128×128 square cells, respectively (i.e., $\varepsilon \in \{0.654, 0.581, 0.590\}$, $\gamma = 1$, $R_1 = 1/6$ and $R_2 = 1/3$) in the domain $(-2, 2)^2$. We employ the unfitted HHO discretization with polynomial order $k = 1$.

isocontours for the two velocity components of the equilibrium flow.



(a) Velocity u_1. (b) Velocity u_2.

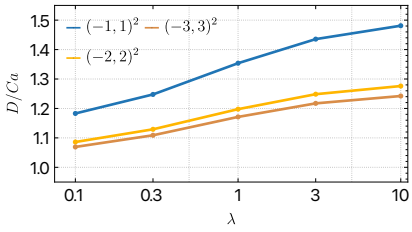
Figure 3.4: Shear-surface tension equilibrium with elliptic interface: isovalues of the velocity components $\boldsymbol{u} = (u_1, u_2)$ for $D = 1/3$, $\mu_1 = \mu_2 = 1$, $Ca = 0.28$ (i.e., $\varepsilon = 0.59$, $\gamma = 1$, $R_1 = 1/6$ and $R_2 = 1/3$) in the domain $\Omega = (-2, 2)^2$ discretized by a mesh composed of 128×128 square cells. We employ the unfitted HHO discretization with polynomial order $k = 1$.

A relevant numerical parameter is the size of the computational domain which has to be large enough so as not to affect the shear-surface tension equilibrium. To evaluate quantitatively the possible influence of this size on our results, we consider two additional computational domains, $\Omega_m := (-1,1)^2$ and $\Omega_M := (-3,3)^2$, still for $D = 1/3$ and $\mu_1 = \mu_2 = 1$. The mesh-size is the same for the three computational domains: this means using a mesh composed of 64×64 (resp., 128×128 and 192×192) square cells for Ω_m (resp., Ω and Ω_M). The predicted values of Ca are 0.246, 0.278 and 0.284 on Ω_m, Ω and Ω_M, respectively. We notice, as expected, that the difference between the predicted capillary numbers decreases by increasing the size of the computational domain owing to the minor influence of the external boundary.

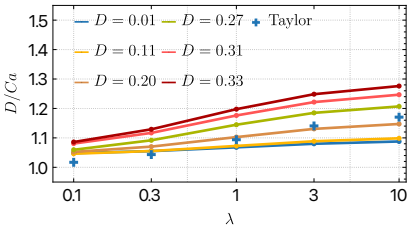
It is interesting to study how the ratio D/Ca depends on the viscosity ratio

$$\lambda := \frac{\mu_1}{\mu_2}. \tag{3.10}$$

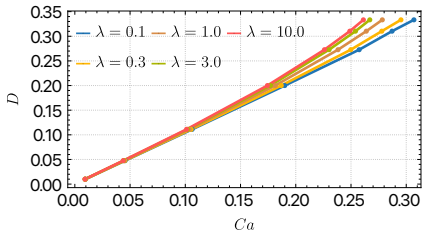
The results are summarized in Figure 3.5. In Figure 3.5a, we present the results obtained for $D = 0.33$ on the three computational domains considered above to verify that our predictions are essentially independent of the size of Ω as long as $(-2,2)^2 \subseteq \Omega$. In Figure 3.5b, we



(a) $D = 0.33$.



(b) $0.01 \leq D \leq 0.33$.



(c) Ratio D/Ca for $\lambda \in \{0.1, 0.3, 1, 3, 10\}$.

Figure 3.5: Figures 3.5a and 3.5b display the ratio D/Ca (Taylor deformation parameter over capillary number) as a function of the viscosity ratio λ. Figure 3.5a: $D = 0.33$ and computational domains $(-a,a)^2$ with $a \in \{1,2,3\}$. Figure 3.5b: $D \in \{0.01, 0.11, 0.20, 0.27, 0.31, 0.33\}$ and computational domain $(-2,2)^2$; the blue crosses refer to the estimate from [124] derived under the assumption of small deformations. Figure 3.5c displays D as a function of Ca; $\lambda \in \{0.1, 0.3, 1, 3, 10\}$ and computational domain $(-2,2)^2$. We employ the unfitted HHO discretization with polynomial order $k = 1$.

fix $\Omega = (-2,2)^2$ and we consider various values for the Taylor deformation parameter, $D \in \{0.01, 0.11, 0.20, 0.27, 0.31, 0.33\}$. Our results for small D are in reasonable agreement with the estimate from [124] which is indeed derived under the assumption of small deformations, i.e.,

when the shape of the interface only slightly departs from a circle. This estimate states that

$$D \approx Ca\frac{19\lambda+16}{16\lambda+16}. \tag{3.11}$$

The discrepancy between our simulations and (3.11) can be attributed to the fact that (3.11) appears to be derived from some analytical expressions for the velocity and pressure derived in a fully 3D (spherical) setting. Finally, in Figure 3.5c, we plot D as a function of Ca for the above values of λ; the results are in good agreement with those reported in [124].

3.2 Fixed-point solver for unknown interface problems

This section describes the various ingredients composing the fixed-point solver used to solve numerically shear-surface tension equilibrium problems with unknown interface.

3.2.1 Fixed-point scheme

The mesh $\mathcal{T}$ is fixed during the whole iterative process and is therefore unfitted to the interface. For simplicity, we consider that the computational domain is a square and that the mesh is composed of square cells. The level-set function is discretized in the continuous finite element space

$$Q_q^c(\mathcal{T}) := \{v_\mathcal{T} \in \mathcal{C}^0(\overline{\Omega}) : v_{\mathcal{T}|T} \in \mathbb{Q}_{q,d},\ \forall T \in \mathcal{T}\}, \tag{3.12}$$

where $\mathbb{Q}_{q,d}$ denotes the space composed of d-variate polynomials of order at most $q \geq 1$ in each spatial variable. In our computations, we take $q := k+1$, and we work with the Bezier–Bernstein basis functions, which present the advantage of taking values in $[0,1]$ (see, e.g., [89, Chap. 2] and [4]). We define the initial discrete interface as

$$\Gamma_\mathcal{T}^0 := \left\{\boldsymbol{x} \in \Omega : \Phi_\mathcal{T}^0(\boldsymbol{x}) = 0\right\}, \tag{3.13}$$

where $\Phi_\mathcal{T}^0 := I_\mathcal{T}(\phi(t=0,\cdot))$ is obtained with the Lagrange interpolation operator $I_\mathcal{T}$ onto $Q_q^c(\mathcal{T})$.

At each iteration of the fixed-point scheme, we first solve, for a given interface, the velocity field (HHO solver, Section 2.3). Then we transport for some (fictitious) time the interface driven by the obtained velocity field. Specifically, at the iteration step $m \geq 0$ of the fixed-point scheme, the following two substeps are performed:

1. Given the discrete interface $\Gamma_\mathcal{T}^m$ (kept fixed in this substep), we solve the HHO unfitted Stokes problem (2.1). Using the cell-agglomeration procedure from [20] produces a mesh $\mathcal{T}^m$ avoiding bad cuts (the superscript refers to $\Gamma_\mathcal{T}^m$), on which we seek approximations $\widehat{\boldsymbol{u}}_\mathcal{T}^m \in \widehat{\boldsymbol{U}}_\mathcal{T}^k$ and $\widehat{p}_\mathcal{T}^m \in P_\mathcal{T}^k$ of the velocity and pressure fields $(\boldsymbol{u}_i^m, p_i^m) \in H^1(\Omega_i;\mathbb{R}^d)\times L^2(\Omega_i)$, $i \in \{1,2\}$, such that

$$-\nabla\cdot\boldsymbol{\sigma}_i^m = \boldsymbol{f}_i \qquad \text{in } \Omega_i^m, \quad i\in\{1,2\}, \tag{3.14a}$$
$$\nabla\cdot\boldsymbol{u}_i^m = 0 \qquad \text{in } \Omega_i^m, \quad i\in\{1,2\}, \tag{3.14b}$$
$$\boldsymbol{u}_2^m = \boldsymbol{g} \qquad \text{on } \partial\Omega, \tag{3.14c}$$
$$[\![\boldsymbol{u}^m]\!] = \boldsymbol{0}, \quad [\![\boldsymbol{\sigma}^m]\!]\boldsymbol{n}_\Gamma^m = \gamma H_\Gamma^m \boldsymbol{n}_\Gamma^m \qquad \text{on } \Gamma_\mathcal{T}^m. \tag{3.14d}$$

More details on the calculation of the normal, $\boldsymbol{n}_\Gamma^m$, and the curvature, H_Γ^m, on the interface $\Gamma_\mathcal{T}^m$ are given in Section 3.2.2.

2. Let $\triangle t^m = t^{m+1} - t^m$ be the fictitious time step such that the fixed-point iteration m is linked with the fictitious discrete time node $t^{m+1} := \sum_{j\in\{0:m\}} \triangle t^j$. We solve

$$\partial_t \widetilde{\Phi}_\mathcal{T}^m + \nabla\cdot\widetilde{\boldsymbol{f}}^m = 0 \qquad \text{in } \Omega\times(0,\triangle t^m), \tag{3.15a}$$

$$\widetilde{\Phi}_\mathcal{T}^m(\boldsymbol{x},t) = \Phi_\mathcal{T}^0(\boldsymbol{x}) \qquad \text{on } \partial\Omega_{\text{in}}\times(0,\triangle t^m), \tag{3.15b}$$

$$\widetilde{\Phi}_\mathcal{T}^m(\boldsymbol{x},0) = \Phi_\mathcal{T}^m(\boldsymbol{x}) \qquad \text{in } \Omega, \tag{3.15c}$$

where

$$\widetilde{\boldsymbol{f}}^m(\boldsymbol{x},t) := \boldsymbol{u}_\mathcal{T}^{m,c}(\boldsymbol{x})\widetilde{\Phi}_\mathcal{T}^m(\boldsymbol{x},t). \tag{3.16}$$

The inflow boundary $\partial\Omega_{\text{in}} := \{\boldsymbol{x}\in\partial\Omega : \boldsymbol{u}_\mathcal{T}^{m,c}\cdot\boldsymbol{n}_\Omega < 0\}$ is assumed to be independent of m (for simplicity). Moreover, $\boldsymbol{u}_\mathcal{T}^{m,c}\in Q_q^c(\mathcal{T};\mathbb{R}^d)$ is a post-processed velocity field obtained by averaging at the interpolation nodes the values of the cell components of the HHO velocity field $\widehat{\boldsymbol{u}}_\mathcal{T}^m$ obtained in the first substep. The averaging is made using weights corresponding to the cell volumes. The problem (3.15) is solved numerically using continuous finite elements (of degree q) and a first-order graph viscosity ensuring a discrete maximum principle on the level-set function; we refer the reader to [70]–[72] for further insight. Finally, we update the level-set function by setting $\Phi_\mathcal{T}^{m+1} := \widetilde{\Phi}_\mathcal{T}^m(\triangle t^m)$ and evaluate $\Gamma_\mathcal{T}^{m+1}$ from $\Phi_\mathcal{T}^{m+1}$ by setting

$$\Gamma_\mathcal{T}^{m+1} := \left\{\boldsymbol{x}\in\Omega : \Phi_\mathcal{T}^{m+1}(\boldsymbol{x}) = 0\right\}. \tag{3.17}$$

Notice that the space discretization of (3.15) uses the original mesh $\mathcal{T}$ of Ω (without any agglomeration) and that we have $\widetilde{\Phi}_\mathcal{T}^m\in Q_q^c(\mathcal{T})$.

The above iterative scheme is stopped when the normal velocity at the interface is sufficiently small. It is not easy to give a quantitative tolerance a priori, but all the simulations reported in Section 3.3 achieve a reduction of the normal velocity at the interface by one to two orders of magnitude. A difficulty is that pushing the iterations too far generally results in some oscillations of the normal velocity, a phenomenon that can be attributed to the loss of mass and the resulting inaccuracy of the transport velocity when updating the interface.

Remark 3.1 (Fictitious time step). The time step used for the time discretization of the problem (3.15) is set to $dt^m := \triangle t^m/\widetilde{N}^m$, where $\widetilde{N}^m > 0$ is a prescribed value so that dt^m is small enough and

$$\triangle t^m \le \min(c_1\triangle t_\gamma, c_2\triangle t_{\text{CFL}}^m). \tag{3.18}$$

Here, $\triangle t_\gamma := \frac{\mu_\sharp}{\gamma}h$ is considered to avoid that the interface moves too much during a fictitious time step. Moreover, $\triangle t_{\text{CFL}}^m$ results from the CFL condition ensuring a discrete maximum principle (see [70]–[72]). Finally, c_1, c_2 are user-dependent parameters. Hereafter, we set $\widetilde{N}^m\in[10,100]$, $c_1 := 2$ and $c_2 := 0.05$, which typically leads to $dt^m \le 10^{-2}$.

3.2.2 Normal and curvature for unknown interface

A well-known issue in problems driven by surface tension is the approximation of the normal and curvature on the interface [55], [106]. Indeed, although the level-set formulation has the

advantage to provide analytic expressions for these quantities (see (2.7)), oscillations can appear (especially for the curvature) when discrete approximations of the level-set function are considered.

Let $\boldsymbol{n}_{\phi_{\mathcal{T}}^m}$ be the normal derived consistently with (2.7) from $\Phi_{\mathcal{T}}^m \in Q_q^{\mathrm{c}}(\mathcal{T})$, i.e.,

$$\boldsymbol{n}_{\phi_{\mathcal{T}}^m} = \frac{\nabla \Phi_{\mathcal{T}}^m}{\|\nabla \Phi_{\mathcal{T}}^m\|_{\ell^2}}. \tag{3.19}$$

At this stage, $\boldsymbol{n}_{\phi_{\mathcal{T}}^m}$ is a piecewise discontinuous field, so that its divergence is not well-defined; it can, however, be evaluated inside each mesh cell. A classical workaround is to smooth $\boldsymbol{n}_{\phi_{\mathcal{T}}^m}$ by using, for each component of $\nabla \Phi_{\mathcal{T}}^m$, the global L^2-orthogonal projection $\Pi_{\mathcal{T}}^q : L^2(\Omega) \to Q_q^{\mathrm{c}}(\mathcal{T})$ (a more local projection on patches can be considered to alleviate the costs). The resulting normal vector field being continuous and piecewise smooth, its divergence is well-defined. Unfortunately, setting

$$\begin{cases} \boldsymbol{n}_{\phi_{\mathcal{T}}^m}^{\mathrm{c}} := \dfrac{\Pi_{\mathcal{T}}^q(\nabla \Phi_{\mathcal{T}}^m)}{\|\Pi_{\mathcal{T}}^q(\nabla \Phi_{\mathcal{T}}^m)\|_{\ell^2}}, & H_{\phi_{\mathcal{T}}^m}^{\mathrm{c}} := -\nabla \cdot \boldsymbol{n}_{\phi_{\mathcal{T}}^m}^{\mathrm{c}}, \\ \boldsymbol{n}_{\Gamma}^{m,\mathrm{c}} := \boldsymbol{n}_{\phi_{\mathcal{T}}^m}^{\mathrm{c}}|_{\Gamma_{\mathcal{T}}^m}, & H_{\Gamma}^{m,\mathrm{c}} := H_{\phi_{\mathcal{T}}^m}^{\mathrm{c}}|_{\Gamma_{\mathcal{T}}^m}, \end{cases} \tag{3.20}$$

does not counter spurious oscillations. Additional smoothing is performed by means of a global L^2-orthogonal projection onto a suitable space of functions defined on the interface and that are continuous and piecewise polynomials. Recalling that the interface $\Gamma_{\mathcal{T}}^m$ is actually discretized by the parameters (l, n) (see (2.29)), we have

$$\Gamma_{\mathcal{T}}^m = \bigcup_{T \in \mathcal{T}^{\Gamma^m}} \bigcup_{j \in \{1:2^n\}} \Upsilon^{T,j}, \qquad \Upsilon^{T,j} := \boldsymbol{r}^{T,j}(\widehat{I}), \tag{3.21}$$

where $\widehat{I} := [0,1]$ and $\boldsymbol{r}^{T,j}$ is the geometric mapping defined by means of suitable interpolation nodes as in (2.28). Then, we introduce the space composed of continuous, piecewise polynomials of order $l' \in \{1{:}l\}$ over the interface, $P_{l'}^{\mathrm{c}}(\Gamma_{\mathcal{T}}^m)$, such that

$$P_{l'}^{\mathrm{c}}(\Gamma_{\mathcal{T}}^m) := \left\{ v \in \mathcal{C}^0\left(\Gamma_{\mathcal{T}}^m\right) : \; v_{|\Upsilon^{T,j}} \circ \boldsymbol{r}^{T,j} \in \mathbb{P}^{l'}(\widehat{I}), \; \forall T \in \mathcal{T}^{\Gamma^m}, \; \forall j \in \{1{:}2^n\} \right\}. \tag{3.22}$$

Let $\Pi_{l'}^{\Gamma} : L^2(\Gamma_{\mathcal{T}}^m) \to P_{l'}^{\mathrm{c}}(\Gamma_{\mathcal{T}}^m)$ be the L^2-orthogonal projection onto $P_{l'}^{\mathrm{c}}(\Gamma_{\mathcal{T}}^m)$. Then, we set

$$\boldsymbol{n}_{\Gamma}^{m,*} := \Pi_{l-1}^{\Gamma}(\boldsymbol{n}_{\Gamma}^{m,\mathrm{c}}), \qquad H_{\Gamma}^{m,*} := \Pi_{1}^{\Gamma}(H_{\Gamma}^{m,\mathrm{c}}), \tag{3.23}$$

where Π_{l-1}^{Γ} is applied component-wise for the normal vector. As above, more local projections on patches can be considered to alleviate the costs, although we observe that the projections in (3.23) are on a one-dimensional manifold, and are therefore relatively cheap. Notice also that, for the curvature, we always pick a piecewise affine representation to temper oscillations. Hereafter, we employ the definitions (3.23) to determine the normal and curvature on the interface.

3.3 Numerical results for unknown interface problems

In this section, we present our results for the equilibrium problem with unknown interface. First, we shortly discuss a couple of verification test cases. Then, we solve the equilibrium problem when the forcing flow at the far field is not of pure shear type. In all the figures presented in this section, the interface is obtained and plotted as a cloud of points which are all zeroes of the discrete level-set function $\Phi_{\mathcal{T}}^m$; we consider the $2^n l$ sampling points per mesh cell constructed as described in Section 2.3.3.

3.3.1 Verification test cases

In this section, we briefly present two verification test cases. Let us first consider an initial flower-like interface described by the level-set function

$$\phi^0(x,y) := x^2 + y^2 - R^2 + c\cos(m\theta), \tag{3.24}$$

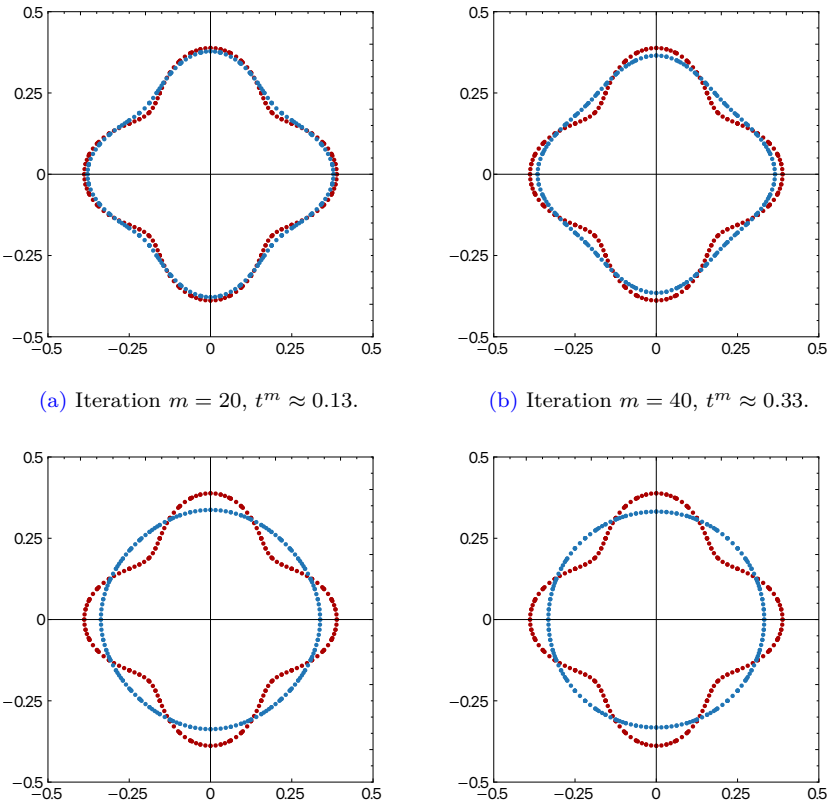
with $\theta := \arctan\left(\frac{y}{x}\right)$ if $x \geq 0$, $\theta := \pi + \arctan\left(\frac{y}{x}\right)$ if $x < 0$, $R := 1/3$, $m := 4$ and $c := 0.04$. We set $\boldsymbol{f}_i := \boldsymbol{0}$, $\mu_i := 1$ (for $i \in \{1,2\}$), $\boldsymbol{g} := \boldsymbol{0}$, and $\gamma := 1$, and there is no shear flow prescribed at the boundary. Hence, the interface is determined only by the surface tension force $\gamma H_\Gamma^{m,*}\boldsymbol{n}_\Gamma^{m,*}$, and the equilibrium shape is a circle. Since there is no shear flow prescribed at the boundary, we can consider a computational domain with a somewhat small size, namely $\Omega := (-0.5, 0.5)^2$.

Starting from the flower-like interface, a few samples of the computed interface at some selected iterations of the fixed-point iterative procedure are illustrated in Figure 3.6. In this fashion, the time step is controlled by $\triangle t_{\mathrm{CFL}}^m$ for the first iterations, when the velocity field is strong owing to the high curvature produced by the flower-like interface, whereas the time step is controlled by $\triangle t_\gamma$ in the later iterations, when the velocity field is weaker so that larger time steps can be employed. Typical time steps are of the order of 10^{-3} for the first iterations and 10^{-1} for the later iterations.

Figure 3.7 displays some error indicators for various mesh-sizes (i.e., $h = 1/16, 1/32, 1/64$) as a function of the pseudo-time resulting from the fixed-point iterative scheme. We consider the curvature and the normal velocity at the interface (measured either in the ℓ^∞- or L^1-norms, the former meaning that the maximum is computed by sampling over the nodes discretizing the interface), and the normalized error on the inner area Ω_1^m defined as

$$\mathrm{diff}(A^m) := \frac{|\Omega_1^m| - |\Omega_1^0|}{|\Omega_1^0|}. \tag{3.25}$$

Notice that $\mathrm{diff}(A^m)$ should vanish in the absence of discretization errors. This quantity is useful to study area loss, a phenomenon typical of the level-set approach. Reference quantities are calculated by using the fact that the circular equilibrium interface conserves the inner area of the initial interface. In all cases, the results reported in Figure 3.7 illustrate well how mesh refinement improves solution accuracy. We notice a somewhat unexpected result in Figure 3.7d, where the coarsest mesh leads to a slightly better ℓ^∞-velocity error. We believe that this can be attributed to some compensation of errors, keeping in mind that all the errors are anyway very small (less than 10^{-7}). We also notice that this behavior is observed only in the special case where no shear flow is prescribed. In all the other cases studied in this section, mesh refinement always improves solution accuracy (see, e.g., Figure 3.11d).



(a) Iteration $m = 20$, $t^m \approx 0.13$.

(b) Iteration $m = 40$, $t^m \approx 0.33$.

(c) Iteration $m = 80$, $t^m \approx 1.13$.

(d) Iteration $m = 180$, $t^m \approx 7.30$.

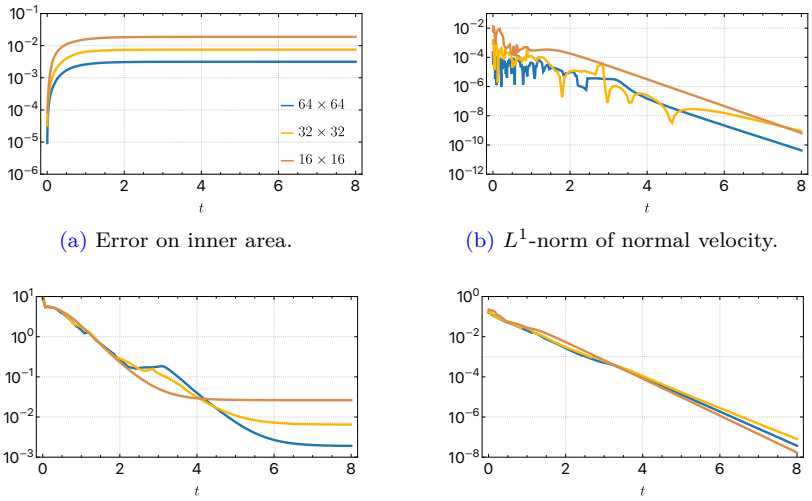
Figure 3.6: First verification test case. In each panel, the red bullets represent the initial interface, and the blue ones the interface at iteration m of the fixed-point procedure. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The mesh is composed of 32×32 square cells. The interface discretization parameters are ($l = 2$, $n = 0$).

In Figure 3.8, we show a few samples of the interface curvature at some selected iterations of the fixed-point iterative procedure. Since the equilibrium interface is circular, we expect that the curvature tends to a constant value equal to $H_* = -3$ (the reciprocal of the circle radius at equilibrium). Consistently, we find numerically that the curvature tends towards a constant value, that is slightly larger than the expected one owing to a slight loss of area.

In the second verification test case, we consider an initial circular interface described by the level-set function

$$\phi^0(x, y) := x^2 + y^2 - R_*^2, \tag{3.26}$$

with $R_* := 1/3$, and we set $\boldsymbol{f}_i := \boldsymbol{0}$, $\mu_i := 1$ (for $i \in \{1, 2\}$), and $\gamma := 1$. In contrast to the previous test case, we now consider the non-homogeneous Dirichlet boundary condition (3.1) enforcing a pure shear flow at the far field. The computational domain is set to $\Omega := (-1, 1)^2$. This size is a bit smaller than the size identified in Section 3.1 to achieve results independent of the size of the domain (namely, $\Omega := (-2, 2)^2$). The present size is chosen to reduce computational costs, keeping in mind that the numerical errors induced by the fixed-



(a) Error on inner area.

(b) L^1-norm of normal velocity.

(c) Curvature error at the interface.

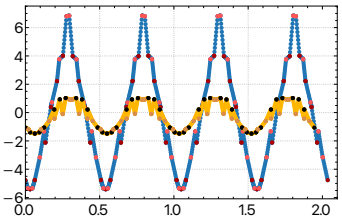
(d) l^∞-norm of normal velocity.

Figure 3.7: First verification test case: error indicators. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The meshes are composed of 16×16 (brown line), 32×32 (yellow line) and 64×64 (blue line) square cells. The interface discretization parameters are $(l = 2, n = 0)$. The number of fixed-point iterations is 100 for the case of the mesh composed of 16×16 square cells, and 180 for the case of meshes composed of 32×32 and 64×64 square cells.

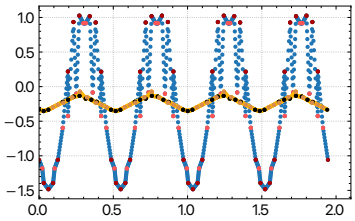
point iterative scheme and the geometric representation of the interface will dominate over the influence of the size of the domain (this has been verified numerically by comparing with some calculations on $\Omega = (-2, 2)^2$).

As above, we control the intensity of the shear by fixing the value of the capillary number to $Ca = \mu_2 \varepsilon L_* / \gamma$. Starting from the circular interface, the fixed-point iterative scheme described in Section 3.2.1 converges toward an elliptic equilibrium interface. We repeat the study for different values of the capillary number $Ca \in \{0.007, 0.07, 0.27\}$, corresponding to the values $\varepsilon \in \{0.01, 0.1, 0.4\}$. Some samples of the interface during the fixed-point scheme are illustrated in Figure 3.9 for $Ca = 0.27$.

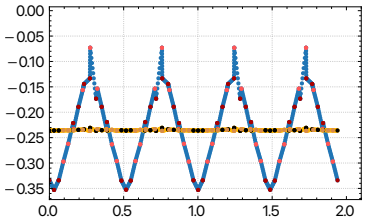
Figure 3.10 reports some of the above error indicators for $Ca \in \{0.007, 0.07, 0.27\}$. In Figure 3.10a, we observe that the convergence of the fixed-point procedure degrades when increasing the capillary number Ca. However, at a fixed capillary number Ca, convergence in space remains clearly visible in Figure 3.11. In Figure 3.10c, we report the normalized Taylor deformation parameter D/D_*. Here, D is estimated by assuming that the interface is always elliptic with axes parallel to the Cartesian axes (notice that the shear flow does not rotate the interface), and D_* corresponds to the value obtained using the steady calculations reported in Figure 3.5c. The agreement is satisfactory owing to the geometric errors, with relative errors of the order of 20%.



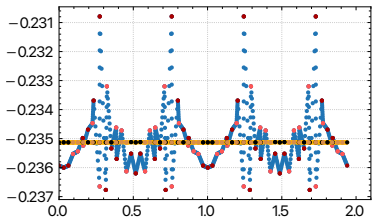
(a) Shifted curvature $H_\Gamma^{m,*} - H_*$ at iterations $m = 40$ (blue, red, pink) and $m = 80$ (yellow, black, brown).



(b) Shifted curvature $H_\Gamma^{m,*} - H_*$ at iterations $m = 80$ (blue, red, pink) and $m = 100$ (yellow, black, brown).

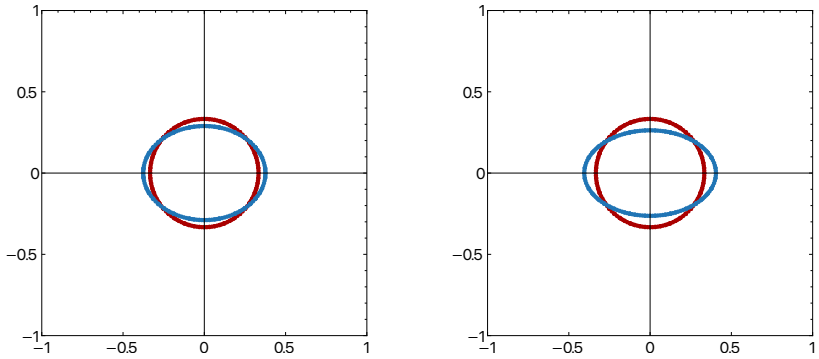


(c) Shifted curvature $H_\Gamma^{m,*} - H_*$ at iterations $m = 100$ (blue, red, pink) and $m = 140$ (yellow, black, brown).



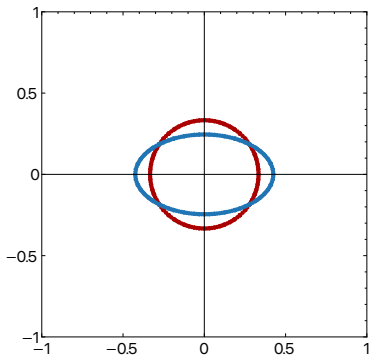
(d) Shifted curvature $H_\Gamma^{m,*} - H_*$ at iterations $m = 140$ (blue, red, pink) and $m = 200$ (yellow, black, brown).

Figure 3.8: First verification test case. Shifted curvature $H_\Gamma^{m,*} - H_*$ (see (3.23)) as a function of the curvilinear abscissa along the interface at iteration $m \in \{40, 80, 100, 140, 200\}$ of the fixed-point procedure. For all the cut cells $T \in \mathcal{T}^{\Gamma^m}$, the blue (or yellow) bullets represent the values at the points inside T, the red (or black) bullets the values at the points on the boundary of T, and the pink (or brown) bullets the values at the points on the boundary of the agglomerated cells inside T. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The mesh is composed of 32×32 square cells. The interface discretization parameters are $(l = 2, n = 0)$.

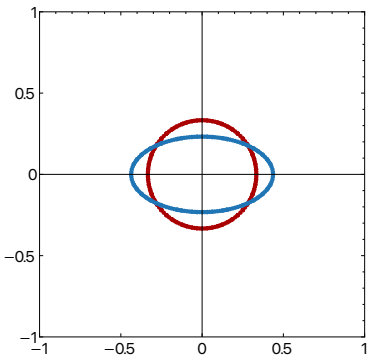


(a) Iteration $m = 200$, $t^m \approx 0.4$.

(b) Iteration $m = 400$, $t^m \approx 0.8$.

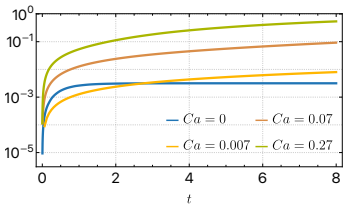


(c) Iteration $m = 600$, $t^m \approx 1.2$.

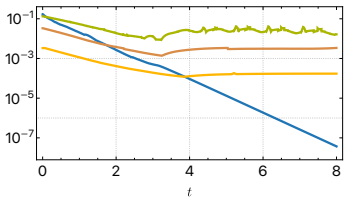


(d) Iteration $m = 800$, $t^m \approx 1.6$.

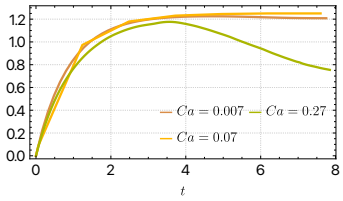
Figure 3.9: Second verification test case: $Ca = 0.27$ (i.e., $\varepsilon = 0.4$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$). In each panel, the red bullets represent the initial interface, and the blue ones the interface at iteration m of the fixed-point procedure. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The mesh is composed of 64×64 square cells. The interface discretization parameters are $(l = 2, n = 0)$.



(a) Error on inner area.

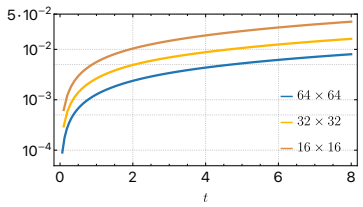


(b) l^∞-norm of normal velocity.

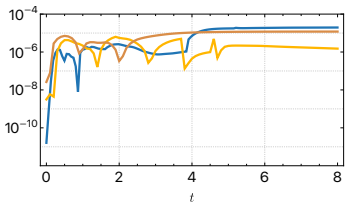


(c) Ratio D/D_* (see caption)

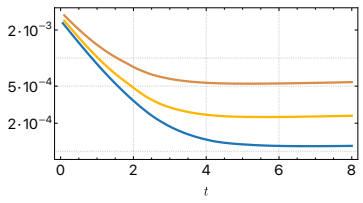
Figure 3.10: Second verification test case: error indicators (inner area, normal velocity at the interface, and Taylor deformation parameter) for $Ca \in \{0.007, 0.07, 0.27\}$ (i.e., $\varepsilon \in \{0.01, 0.1, 40\}$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$), corresponding respectively to the yellow, brown, and green bullets. In Figures 3.10a and 3.10b, we additionally report the errors for $\varepsilon = 0$ (blue bullets). The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The meshes are composed of 64×64 square cells. The interface discretization parameters are ($l = 2$, $n = 0$). The number of fixed-point iterations is 210, 260, 1900, and 3900 for the cases $Ca = 0$, $Ca = 0.007$, $Ca = 0.07$, and $Ca = 0.27$, respectively.



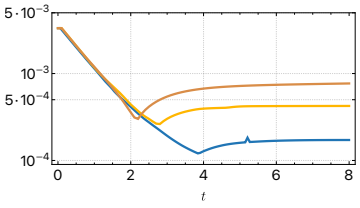
(a) Error on inner area.



(b) L^1-norm of normal velocity.



(c) Hausdorff distance $d(\Gamma_{\mathcal{T}}^{m+1}, \Gamma_{\mathcal{T}}^{m})$ evaluated by sampling at the discrete nodes.



(d) l^∞-norm of normal velocity.

Figure 3.11: Second verification test case: error indicators for $Ca = 0.007$ (i.e., $\varepsilon = 0.01$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$) for various levels of mesh refinement. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The meshes are composed of 16×16 (brown line), 32×32 (yellow line) and 64×64 (blue line) square cells. The interface discretization parameters are ($l = 2$, $n = 0$). The number of fixed-point iterations is 21, 34, and 66 for 16×16, 32×32, and 64×64 square cells, respectively.

3.3.2 Test cases with unknown interface

Let us consider the initial circular interface described by the level-set function defined in (3.26) with $R_* := 1/3$, $\boldsymbol{f}_i := \boldsymbol{0}$, $\mu_i := 1$ for $i \in \{1,2\}$ (and thus the viscosity contrast is $\lambda = \mu_1/\mu_2 := 1$), and $\gamma := 1$. As in the previous section, we set $\Omega := (-1,1)^2$. We now consider the following perturbation of the non-homogeneous Dirichlet boundary condition (3.1):

$$\boldsymbol{g} := \boldsymbol{u}_{\varepsilon|\partial\Omega}, \qquad \boldsymbol{u}_\varepsilon(x,y) := \varepsilon\left((x,-y)^{\mathrm{T}} + 0.5(\sin(\pi y),\sin(\pi x))^{\mathrm{T}}\right), \tag{3.27}$$

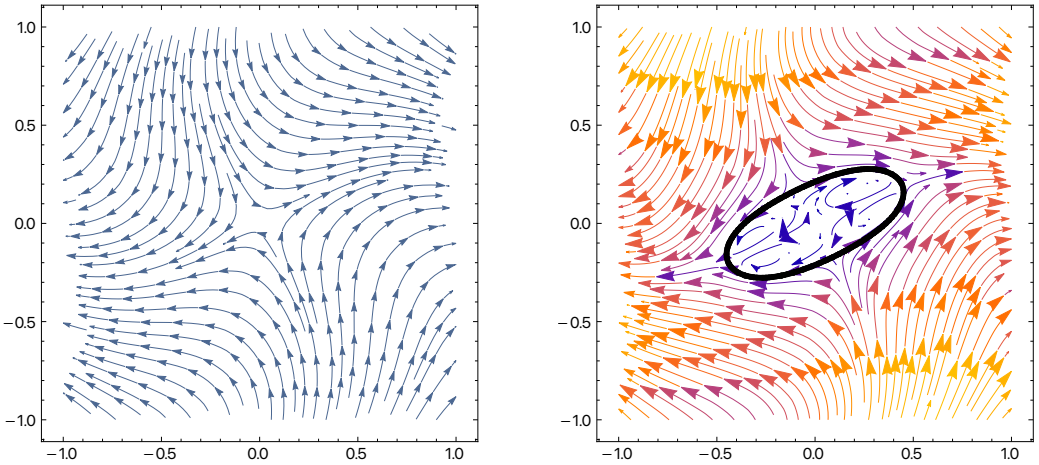
with $\varepsilon > 0$. We estimate the shear parameter as

$$\varepsilon_* = \max_{\boldsymbol{x}\in\Omega}\max_{i,j}|(\nabla^{\mathrm{s}}\boldsymbol{u}_\varepsilon)_{ij}(\boldsymbol{x})| = \varepsilon\frac{\pi}{2}, \tag{3.28}$$

so that the capillary number is now evaluated as follows:

$$Ca = \frac{1}{2}\mu\frac{\varepsilon\pi L_*}{\gamma}, \tag{3.29}$$

with $L_* := 2R_*$ and $\mu := \mu_1 = \mu_2$. In Figure 3.12a, we illustrate the velocity field (3.27) without the interface (i.e., $\gamma = 0$), whereas in Figure 3.12b the presence of the interface is taken into account for the case $Ca = 0.27$ ($\varepsilon = 0.26$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$). The interface evolution is illustrated in Figure 3.13 for the same value of the capillary number.

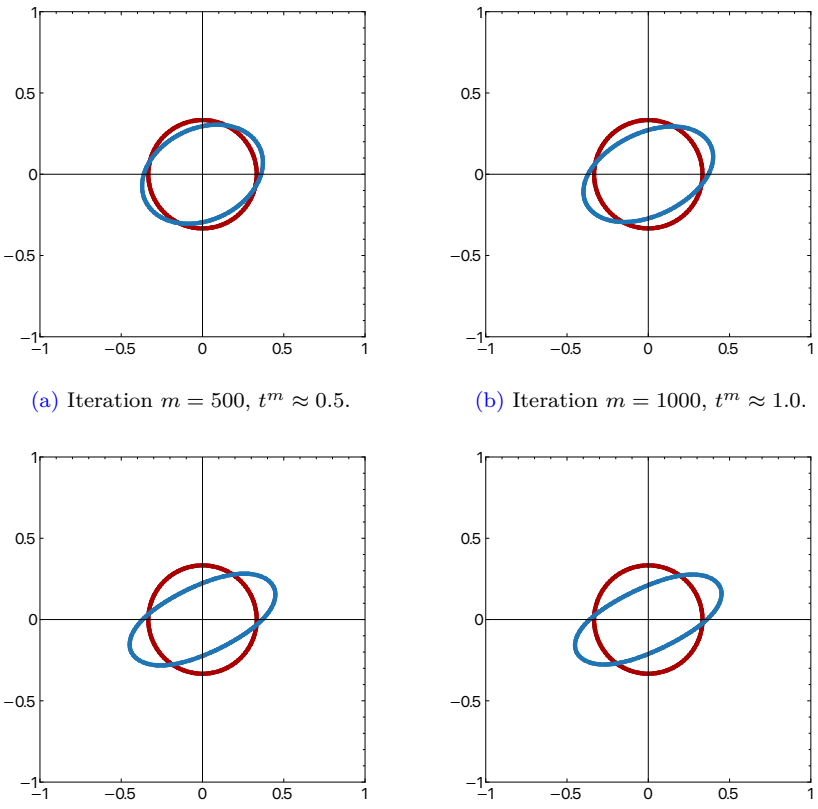


(a) Streamlines for the perturbed velocity field (3.27).

(b) Interface at iteration $m = 3000$, $t \approx 3$, for the perturbed shear flow with $Ca = 0.27$.

Figure 3.12: Streamlines for the test case with unknown interface. Figure 3.12a illustrates the velocity field (3.27) without the interface (i.e., $\gamma = 0$), Figure 3.12b includes the presence of the interface with $Ca = 0.27$ (i.e., $\varepsilon = 0.26$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$). The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The mesh is composed of 128×128 square cells. The interface discretization parameters are $(l = 2, n = 0)$.

In Figure 3.14, we fix $\varepsilon = 0.26$ ($\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$) and we study the convergence in space of the fixed-point procedure. Consistently with the previous results on the verification test cases, the proposed methodology reduces the l^∞-norm of the normal velocity by a factor



(a) Iteration $m = 500$, $t^m \approx 0.5$.

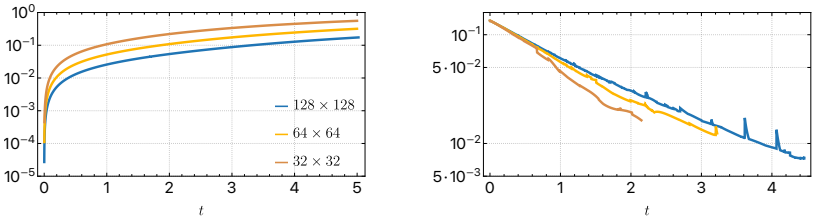
(b) Iteration $m = 1000$, $t^m \approx 1.0$.

(c) Iteration $m = 3000$, $t^m \approx 3.0$.

(d) Iteration $m = 4000$, $t^m \approx 4.0$.

Figure 3.13: Interface evolution for the test case with unknown interface with $Ca = 0.27$ (i.e., $\varepsilon = 0.26$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$). In each panel, the red bullets represent the initial interface, and the blue ones the interface at iteration m of the fixed-point procedure. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The mesh is composed of 128×128 square cells. The interface discretization parameters are ($l = 2$, $n = 0$).

of 10. We notice that a more pronounced error reduction can be reached on finer meshes.



(a) Error on inner area.

(b) l^∞-norm of normal velocity.

Figure 3.14: Error indicators for $Ca = 0.27$ (i.e., $\varepsilon = 0.26$, $\gamma = 1$, $\mu_2 = 1$, $R_* = 1/3$) for various levels of mesh refinement. The HHO polynomial order is $k = 1$, and we employ $q = 2$ for the level-set approximation. The meshes are composed of 32×32 (brown line), 64×64 (yellow line) and 128×128 (blue line) square cells. The interface discretization parameters are ($l = 2$, $n = 0$). The number of fixed-point iterations is 21, 2400, and 5100 for 32×32, 64×64, and 128×128 square cells, respectively.

3.3.3 Conclusions

The conclusion of the results on the verification test cases is that the convergence of the fixed-point procedure is in general satisfactory. However, some difficulties are encountered if one insists in reducing the normal velocity at the interface by more than an order of magnitude. This difficulty can possibly be alleviated by improving on the area loss, which, in turn, may be partly caused by the lack of divergence-free property of the post-processed velocity field used in substep 2 of the fixed-point iterative procedure. These points will be investigated in future work. Finally, in the case of an unknown interface, mesh refinement is again crucial to achieve a significant error reduction on the normal velocity, and thus approach a steady-state solution. Moreover, simulations become more challenging as the capillary number is increased. Indeed, higher curvatures contribute to the presence of sharp gradients, which demand finer spatial and temporal discretizations to accurately represent the associated flow patterns. It is also worth noting that as the capillary number is increased, the loss of area becomes more pronounced, inducing a larger error on the velocity approximation.

Part II

Hybridizable Discontinuous Galerkin methods

Chapter 4

Unfitted HDG solver

4.1 Introduction

At the microscale, fluid behavior is primarily governed by surface tension, energy dissipation, and fluidic resistance [10], [61], [120]. These microfluidics systems, typically ranging from 100 nanometers to 500 micrometers, are characterized by low Reynolds number, indicatives of Stokes flows. Such simulations present considerable challenges, especially in problems that involve complex or evolving geometries. One particularly challenging aspect is the high-fidelity resolution to multi-fluid problems, where immiscible fluids interact while exhibiting distinct flow characteristics. In this chapter, we focus on such problems, and we develop a high-order method based on a Hybridizable Discontinuous Galerkin (HDG) method, coupled with the NURBS-enhanced finite element method (NEFEM), and adopting a geometrically unfitted approach. We denote it as the unfitted HDG-NEFEM method.

HDG methods have been introduced in the present day form a decade ago in [35], and vastly applied to flow problems [36], [91], [100], [101]. HDG methods offer several attractive features, including degree-adaptive strategies [65] and L^2-superconvergence on the primal variable by local postprocessing [62]. Moreover, the formulation is computationally efficient, with respect to discontinuous Galerkin methods, owing to the global coupling of the skeleton unknowns by means of the static condensation procedure [50].

HDG-NEFEM methods have been introduced in [115], [116] and rely on the use of Non-uniform rational B-spline (NURBS) [83] for the geometry representation of interfaces and boundaries. This approach enables a seamless transition from computer-aided design (CAD) geometries to computation descriptions, preserving the optimality of HDG methods.

Geometrically unfitted methods refer to approaches where the mesh does not fit the geometry, and the variational formulation is modified to incorporate constraints on interfaces and boundaries into the computation. Unfitted meshes are particularly suitable for complex geometries or time-evolving interfaces, where the mesh may result extremely distorted and unsuitable for computation. The extension of unfitted methods to hybridized discretization methods is relatively recent, with applications to immiscible incompressible Stokes interface problems [22], [73], [108].

The unfitted approach adopted in the current HDG-NEFEM method builds upon the work presented in [75], [76] enforcing Dirichlet boundary conditions by means of Nitsche's

method [103]. To handle interface conditions, the key idea of the method is the doubling of the unknowns in the mesh elements cut by the interface, thus enriching the approximation space and achieving a more precise representation of the solution in the vicinity of the interface. Additionally, the method avoids the introduction of additional degrees of freedom (DoFs) either on the interface or along the boundary.

A well-known drawback in unfitted methods is the presence of unfavorable cuts in the mesh, resulting in mesh elements with a small portion of their volume within the computational domain. This situation often leads to ill-conditioned linear systems, which, in turn, adversely affect the accuracy of the computations. To address this issue, the unfitted HDG-NEFEM method employs an element extension strategy based on [99]. This approach addresses the issue of cut elements by combining neighboring elements, erasing the unknowns associated with the ill-cut element, and extrapolating the well-cut element basis functions into the ill-cut element.

The structure of this chapter is as follows. Section 4.2 introduces the model problem. Section 4.3 is devoted to the presentation of the unfitted HDG-NEFEM solver for the Stokes interface problem. Section 4.4 describes the employed strategies to handle unfitted domains. Section 4.5 presents two strategies, degree adaptivity, Subsection 4.5.1, and element extension, Subsection 4.5.2, to ensure robustness and accuracy.

4.2 Problem statement

Let $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$ be an open bounded domain with boundary $\partial\Omega$ and $\mathtt{n_{sd}} := 2$ the number of spatial dimensions. The boundary $\partial\Omega$ is composed of two disjoint parts, the Dirichlet portion Γ_{D}, where the value $\boldsymbol{u}_{\mathrm{D}} \in [\mathcal{H}^{\frac{1}{2}}(\Gamma_{\mathrm{D}})]^{\mathtt{n_{sd}}}$ of the velocity is imposed, and the Neumann portion Γ_{N}, where a pseudo-traction $\boldsymbol{t} \in [\mathcal{L}^2(\Gamma_{\mathrm{N}})]^{\mathtt{n_{sd}}}$ is applied. Formally, $\partial\Omega = \overline{\Gamma_{\mathrm{D}}} \cup \overline{\Gamma_{\mathrm{N}}}$ such that $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$.
Suppose now that Ω is split by a fixed interface Υ in two disjoint subdomains occupied each by an immiscible incompressible Stokes fluid. That is, $\overline{\Omega} = \overline{\Omega^1} \cup \overline{\Omega^2}$, $\Omega^1 \cap \Omega^2 = \emptyset$, and $\Upsilon = \overline{\Omega^1} \cap \overline{\Omega^2}$. Note that the boundary of each fluid is $\partial\Omega^i = \Gamma_{\mathrm{D}}^i \cup \Gamma_{\mathrm{N}}^i \cup \Upsilon$ with $\Gamma_{\mathrm{D}}^i \cap \Upsilon = \emptyset$, and $\Gamma_{\mathrm{N}}^i \cap \Upsilon = \emptyset$, for $i \in \{1, 2\}$. Moreover, we define the domain $\hat{\Omega}$ as $\hat{\Omega} = \Omega^1 \cup \Omega^2$, similarly with $\hat{\Gamma}_{\mathrm{D}} = \Gamma_{\mathrm{D}}^1 \cup \Gamma_{\mathrm{D}}^2$, and $\hat{\Gamma}_{\mathrm{N}} = \Gamma_{\mathrm{N}}^1 \cup \Gamma_{\mathrm{N}}^2$. The interface Υ and the boundary $\partial\Omega$ are represented using NURBS and are composed by $\mathtt{n}_\Upsilon$ and $\mathtt{n}_{\partial\Omega}$ NURBS curves, respectively. That is,

$$\Upsilon := \bigcup_{j \in \{1:\mathtt{n}_\Upsilon\}} \boldsymbol{C}_\Upsilon^j([0,1]), \qquad \partial\Omega := \bigcup_{j \in \{1:\mathtt{n}_{\partial\Omega}\}} \boldsymbol{C}_{\partial\Omega}^j([0,1]), \tag{4.1}$$

where $\boldsymbol{C}^j(\lambda) : \lambda \to \boldsymbol{C}^j(\lambda)$ is a generic NURBS curve [83] defined in the parametric domain $\lambda \in [0, 1]$.

The problem aims to find the unknown velocity and pressure fields, $(\boldsymbol{u}, p) \in [\mathcal{H}^1(\hat{\Omega})]^{\mathtt{n_{sd}}} \times \mathcal{L}^2(\Omega)$, whose restrictions to each subdomain Ω^i are $(\boldsymbol{u}, p)|_{\Omega^i} = (\boldsymbol{u}^i, p^i) \in [\mathcal{H}^1(\Omega^i)]^{\mathtt{n_{sd}}} \times \mathcal{L}^2(\Omega^i)$, $i \in \{1, 2\}$. Thus,

$$[\mathcal{H}^1(\hat{\Omega})]^{\mathtt{n_{sd}}} := \left\{ \boldsymbol{v} \in [\mathcal{L}^2(\Omega)]^{\mathtt{n_{sd}}} \mid \boldsymbol{v}|_{\Omega^i} \in [\mathcal{H}^1(\Omega^i)]^{\mathtt{n_{sd}}},\ \forall i \in \{1, 2\} \right\}.$$

Finally, the strong form of the problem can be written as find $(\boldsymbol{u}, p) \in [\mathcal{H}^1(\hat{\Omega})]^{\mathtt{n_{sd}}} \times \mathcal{L}^2(\Omega)$ such

that

$$\begin{cases} -\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u}-p\boldsymbol{I}_{\mathtt{n_{sd}}}) = \boldsymbol{s} & \text{in } \hat{\Omega}, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 & \text{in } \hat{\Omega}, \\ \boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}} & \text{on } \hat{\Gamma}_{\mathrm{D}}, \\ \left(\mu\boldsymbol{\nabla}\boldsymbol{u}-p\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n} = \boldsymbol{t} & \text{on } \hat{\Gamma}_{\mathrm{N}}, \end{cases} \tag{4.2}$$

$$\begin{cases} \boldsymbol{u}^1 = \boldsymbol{u}^2 \\ \left(\mu^1\boldsymbol{\nabla}\boldsymbol{u}^1-p^1\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n}^1+\left(\mu^2\boldsymbol{\nabla}\boldsymbol{u}^2-p^2\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n}^2 = \gamma(\boldsymbol{\nabla}\cdot\boldsymbol{n}^1)\boldsymbol{n}^1+\boldsymbol{\nabla}\gamma \end{cases} \text{on } \Upsilon,$$

where $\mu = \mu^i > 0$ on Ω^i, $i \in \{1,2\}$, is the piecewise dynamic viscosity of each fluid, assumed constant in Ω^i, $\boldsymbol{s} \in [\mathcal{L}^2(\Omega)]^{\mathtt{n_{sd}}}$ is the volumetric source term, $\boldsymbol{n}^i$ is the outward unit normal vector to the corresponding domain Ω^i, and γ is the surface tension coefficient. Note that in most cases, the surface tension, γ, is assumed constant, thus, $\boldsymbol{\nabla}\gamma = \boldsymbol{0}$, and, consequently, the interface shear stress is continuous. Finally, the last two equations impose the continuity of velocity and equilibrium of forces along the interface Υ.

Remark 4.1 (One-fluid problem). When $\Upsilon = \emptyset$, the interface conditions disappear, and we solve the one-fluid problem in the domain $\Omega^1 \equiv \Omega$.

Remark 4.2 (Uniqueness). When $\hat{\Gamma}_{\mathrm{N}} = \emptyset$, the Stokes interface problem is solvable up to a global additive constant on the pressure, which we fix by imposing

$$\int_{\hat{\Omega}} p \,\mathrm{d}\hat{\Omega} = \int_{\hat{\Omega}} p^{\mathrm{ref}} \mathrm{d}\hat{\Omega},$$

where p^{ref} is the analytical pressure typically equal to zero in physically meaningful problems.

Remark 4.3 (Multi-fluid interface problem). Formulation (4.2) can be readily extended to address multi-fluid problems, where Ω is partitioned into a generic number of connected, open, bounded sets, each occupied by an immiscible, incompressible Stokes fluid.

4.3 Computational setting

Let Ω_{o} be a shape-regular polyhedral domain containing Ω, i.e., $\Omega \subseteq \Omega_{\mathrm{o}}$. Then, we consider a shape-regular mesh composed of $\mathtt{n_{el}}$ disjoint (open) subdomains Ω_e such that Ω_{o} is exactly covered, i.e.,

$$\overline{\Omega_{\mathrm{o}}} := \bigcup_{e\in\{1:\mathtt{n_{el}}\}} \overline{\Omega_e}.$$

The boundaries $\partial\Omega_e$ of the mesh element Ω_e define the *internal skeleton*, Γ:

$$\Gamma := \Big[\bigcup_{e\in\{1:\mathtt{n_{el}}\}} \partial\Omega_e\Big] \setminus (\hat{\Gamma}_{\mathrm{D}} \cup \hat{\Gamma}_{\mathrm{N}}).$$

We denote by Γ_f a generic internal face of Γ, and by $\mathtt{n_{fc}}$ the total number of internal faces, i.e., $\Gamma = \bigcup_{f\in\{1:\mathtt{n_{fc}}\}} \Gamma_f$. It is important to recall that the interface, Υ, and the external boundary, $\hat{\Gamma}_{\mathrm{D}}$ and $\hat{\Gamma}_{\mathrm{N}}$, do not need to align with the mesh. Moreover, for all $e \in \{1:\mathtt{n_{el}}\}$, we define Ω_e^i the

region of Ω_e that belongs to the fluid indexed by i, i.e.,

$$\Omega_e^i := \Omega_e \cap \Omega^i, \qquad \forall i \in \{1,2\}.$$

Similarly, we define $\partial\Omega_e^i := \partial\Omega_e \cap (\overline{\Omega^i} \setminus \Upsilon)$ as the portion of the boundary $\partial\Omega_e$ belonging to fluid i. Finally, let $\Gamma^i := \Gamma \cap (\Omega^i \setminus \Upsilon)$ and $\Gamma_f^i := \Gamma_f \cap (\Omega^i \setminus \Upsilon)$ be the portions of the inner skeleton Γ and of a face Γ_f inside Ω^i, respectively, for all $f \in \{1{:}\mathtt{n_{fc}}\}$ and all $i \in \{1,2\}$.

On this broken domain, the problem defined in (4.2) can be written as follows:

$$\text{for all } e \in \{1{:}\mathtt{n_{el}}\} \text{ and all } i \in \{1,2\} \begin{cases} \boldsymbol{L}_e^i + \sqrt{\mu^i}\boldsymbol{\nabla}\boldsymbol{u}_e^i = \boldsymbol{0} & \text{in } \Omega_e^i, \\ \boldsymbol{\nabla}\cdot(\sqrt{\mu^i}\boldsymbol{L}_e^i) + \boldsymbol{\nabla}p_e^i = \boldsymbol{s} & \text{in } \Omega_e^i, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u}_e^i = 0 & \text{in } \Omega_e^i, \\ \boldsymbol{u}_e^i = \boldsymbol{u}_{\mathrm{D}} & \text{on } \overline{\Omega_e} \cap \Gamma_{\mathrm{D}}^i, \\ \left(\sqrt{\mu^i}\boldsymbol{L}_e^i + p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n}_e^i = -\boldsymbol{t} & \text{on } \overline{\Omega_e} \cap \Gamma_{\mathrm{N}}^i, \end{cases} \tag{4.3a}$$

$$\text{for all } e \in \{1{:}\mathtt{n_{el}}\} \begin{cases} \boldsymbol{u}^1 = \boldsymbol{u}^2 & \\ \left(\sqrt{\mu^1}\boldsymbol{L}_e^1 + p^1\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n}_e^1 & \text{on } \Upsilon \cap \Omega_e, \\ \quad + \left(\sqrt{\mu^2}\boldsymbol{L}_e^2 + p^2\boldsymbol{I}_{\mathtt{n_{sd}}}\right)\boldsymbol{n}_e^2 = -\gamma(\boldsymbol{\nabla}\cdot\boldsymbol{n}_e^1)\boldsymbol{n}_e^1 - \boldsymbol{\nabla}\gamma & \end{cases} \tag{4.3b}$$

$$\text{for all } e \in \{1{:}\mathtt{n_{el}}\} \text{ and all } i \in \{1,2\} \begin{cases} [\![\boldsymbol{u}^i \otimes \boldsymbol{n}^i]\!] = \boldsymbol{0} \\ [\![(\sqrt{\mu^i}\boldsymbol{L}^i + p^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}^i]\!] = \boldsymbol{0} \end{cases} \text{on } \Gamma \cap \Omega_e^i, \tag{4.3c}$$

where $\boldsymbol{L} := -\sqrt{\mu}\boldsymbol{\nabla}\boldsymbol{u}$ is the mixed variable so that $\boldsymbol{L}_e^i = -\sqrt{\mu^i}\boldsymbol{\nabla}\boldsymbol{u}_e^i$ is its restriction to Ω_e^i, for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1,2\}$. The interface conditions stated in (4.3b) are reformulated element-wisely in (4.2), ensuring the enforcement of the desired continuity conditions. The equations in (4.3a) and (4.3c) are, respectively, the usual local equations and the so-called *transmission conditions* common in HDG. The latter imposes, respectively, continuity of velocity and normal flux across the skeleton of the mesh. Consistently with HDG methods, we introduce the face-based variable $\hat{\boldsymbol{u}}^i$, the so-called *hybrid velocity*, to represent the trace of the solution on each face of Γ^i so that the local problem (4.3a) is completed by adding

$$\boldsymbol{u}_e^i = \hat{\boldsymbol{u}}^i \qquad \text{on } \Gamma^i \cap \Omega_e,$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$.

The *jump* $[\![\odot]\!]$ operator has been introduced following the definition by [98], such that, along each portion of Γ (Υ) it sums the values of a generic quantity $\odot$ from the left and from the right, say Ω_l and Ω_r (Ω_e^l and Ω_e^r), namely

$$[\![\odot]\!] = \odot_l + \odot_r.$$

Note that the above definition of the jump operator always involves the outward unit normal to a surface, say $[\![\odot\boldsymbol{n}]\!]$. Thus, for instance, at the edge between elements Ω_l and Ω_r, this definition implies $[\![\odot\boldsymbol{n}]\!] = \odot_l\boldsymbol{n}_l + \odot_r\boldsymbol{n}_r$ where $\boldsymbol{n}_l$ and $\boldsymbol{n}_r$ are the outward unit normals to $\partial\Omega_l$ and $\partial\Omega_r$, respectively. Moreover, recall that $\boldsymbol{n}_l = -\boldsymbol{n}_r$.

To simplify the presentation, we consider three types of elements: (i) the *standard HDG elements* not cut by Υ or $\partial\Omega$, i.e., $\Upsilon \cap \Omega_e = \emptyset$ and $\partial\Omega \cap \Omega_e = \emptyset$, (ii) the *immersed boundary elements* cut by $\partial\Omega$ but not by Υ, i.e., $\Upsilon \cap \Omega_e = \emptyset$ and $\partial\Omega \cap \Omega_e \neq \emptyset$, and (iii) the *interface elements* cut by Υ but not by $\partial\Omega$, i.e., $\Upsilon \cap \Omega_e \neq \emptyset$ and $\partial\Omega \cap \Omega_e = \emptyset$. Moreover, we suppose that the interface splits each element (at most) into two different regions. We refer to Figure 4.1 to illustrate such possible configurations. Finally, note that all the defined elements, (i), (ii), and (iii), also encompass the case in which the external boundary is aligned with the mesh skeleton, which means $\partial\Omega_e \cap \partial\Omega \neq \emptyset$.

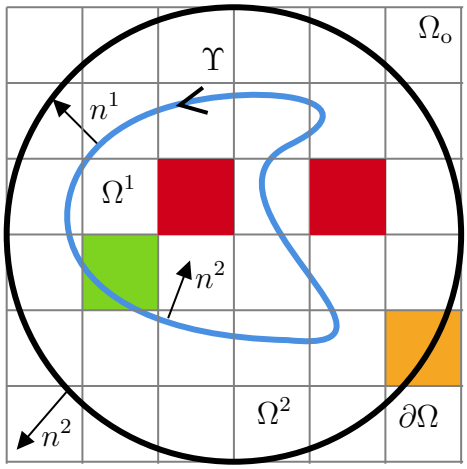
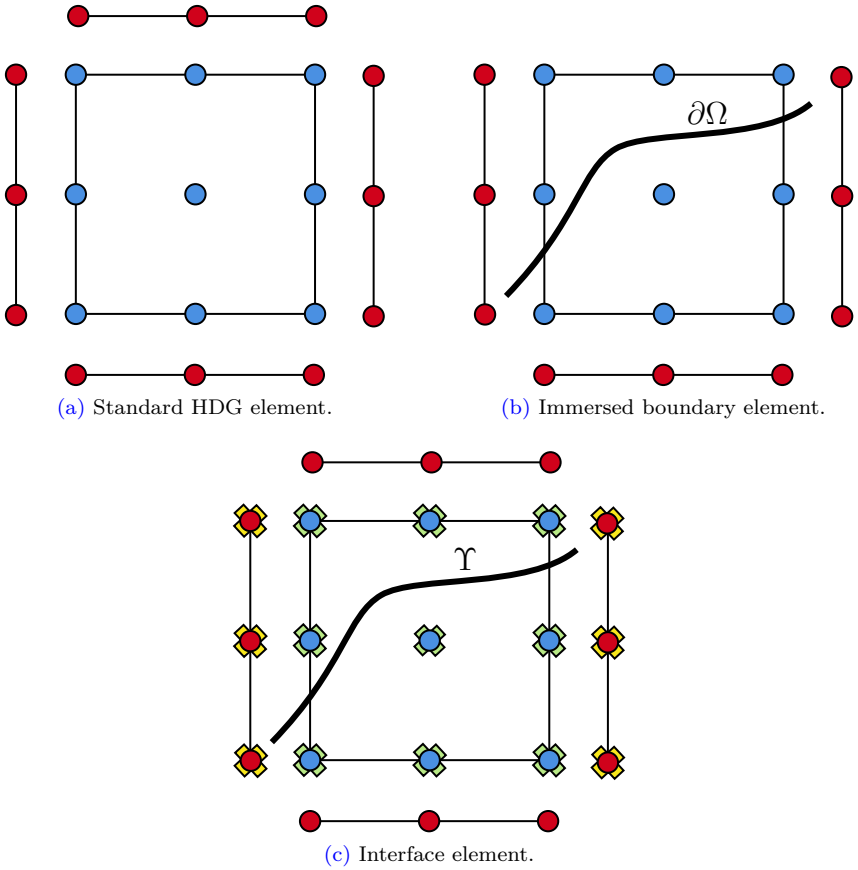


Figure 4.1: Domain Ω_o partitioned into 6×6 square elements. The external boundary $\partial\Omega$ and the interface Υ are not aligned with the mesh skeleton. The elements in red represent standard HDG elements contained in Ω^1 and Ω^2, the element in orange represents an immersed boundary element, cut by $\partial\Omega$, and the element in green represents an interface element, cut by Υ.

The HDG discretization employs element- and face-based unknowns to discretize (4.3). In the standard HDG elements, we use the standard HDG approximation, employing polynomial basis functions of order $k \geq 0$ for both element- and face-based unknowns. In the immersed boundary elements, the same approximation space is used for element-based unknowns and for the face-based unknowns located on faces $\Gamma_f \subset \Gamma$. Notably, the unknowns lying on faces Γ_f outside the domain Ω are removed from the problem. In the interface elements, the element-based unknowns are duplicated, as are the face-based unknowns for the faces intersected by the interface. Faces that remain uncut maintain the standard number of unknowns, as illustrated in Figure 4.2. Note that while the basis functions are defined across the entire element, Ω_e, or face, Γ_f, independently on the cut, the quadrature domain is restricted to Ω_e^i or Γ_f^i, as detailed in Section 4.4. Moreover, we define the face-based unknowns only in the interior mesh skeleton Γ, without introducing additional DoFs along the fluid interface or the external boundaries $\hat{\Gamma}_D$ and $\hat{\Gamma}_N$. When $\hat{\Gamma}_N$ fits the mesh, we resort to the standard HDG formulation (see for example [117]) with face-based unknowns defined on $\Gamma \cup \hat{\Gamma}_N$.

Henceforth, we consider rectangular domains Ω_o, partitioned in rectangular-shaped elements Ω_e. In this specific context, we use Lagrange basis functions, denoted as $\mathcal{P}^k_{\text{La}}$ (commonly used in fitted HDG methods [64]), for element-based unknowns. Owing to the rectangular shape of Ω_e, $\mathcal{P}^k_{\text{La}}$ is a two-dimensional tensor basis, thus respecting the M-decomposition assumption, as outlined in [34], [37]. For the face-based unknowns, we compare two possible bases: the

Lagrange basis, $\hat{\mathcal{P}}^k_{\mathrm{La}}$, and the Legendre basis, $\hat{\mathcal{P}}^k_{\mathrm{Le}}$. In Section 5.1, we will conduct a comparative analysis of both bases, showcasing the superior performance of Legendre face-based unknowns in terms of mitigating the conditioning of the global problem.



(a) Standard HDG element. (b) Immersed boundary element.

(c) Interface element.

Figure 4.2: Element- and face-based unknowns used for the different types of cut elements. Figure (4.2a) represents a standard HDG element, Figure (4.2b) represents an immersed boundary element, and Figure (4.2c) represents an interface element. Blue bullets represent element-based unknowns, red bullets face-based unknowns, green crosses the doubled element-based unknowns employed in interface elements, and yellow crosses the doubled face-based unknowns employed in interface cut faces.

In what follows, we use the standard HDG notation $(\cdot,\cdot)_S$ and $\langle\cdot,\cdot\rangle_S$ to denote the $\mathcal{L}^2$-product in a generic $\mathtt{n_{sd}}$- and $(\mathtt{n_{sd}}-1)$-dimensional subset $S \subset \Omega$, respectively.

4.3.1 Local problems

HDG defines, element-by-element, the so-called local problems. Consistently with the possible types of elements, three main cases are identified.

For the standard HDG elements, the standard local HDG problem is recovered in each fluid i equals either 1 or 2. Then, given the prescribed velocity $\boldsymbol{u}_{\mathrm{D}}$ on Γ_{D}, the pseudo-traction $\boldsymbol{t}$ on Γ_{N}, and the hybrid velocity $\hat{\boldsymbol{u}}^i$ on $\Gamma^i \cap \Omega_e$, we seek $(\boldsymbol{L}^i, \boldsymbol{u}^i, p^i) \in [\mathcal{H}(\mathrm{div}; \Omega_e^i)]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}} \times [\mathcal{H}^1(\Omega_e^i)]^{\mathtt{n_{sd}}} \times \mathcal{H}^1(\Omega_e^i)$ solving (4.3a). The problem is well-posed. We recall the standard numerical trace of

the diffusive flux, defined in [35], [64], such that

$$\widehat{(\sqrt{\mu}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e} := (\sqrt{\mu^i}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e + \begin{cases}\tau(\boldsymbol{u}_e^i-\boldsymbol{u}_{\mathrm{D}}), & \text{on } \hat{\Gamma}_{\mathrm{D}}\cap\partial\Omega_e^i,\\ \tau(\boldsymbol{u}_e^i-\hat{\boldsymbol{u}}^i), & \text{on } \Gamma\cap\partial\Omega_e^i,\end{cases}$$

with τ a stabilization parameter. Consistently with [63], we set $\tau := c\mu/\ell$ where $\mu := \max(\mu^1,\mu^2)$, ℓ is a characteristic length of the domain, and c is a scaling factor which henceforth is set equal to 3. We refer to [63] for more details.

4.3.1.1 Immersed boundary elements

For immersed boundary elements, we introduce a Nitsche's penalty term [103] to weakly impose the Dirichlet boundary conditions when $\hat{\Gamma}_{\mathrm{D}}$ cuts Ω_e, namely, the weak formulation of (4.3a) is as follows: for all $e\in\{1{:}\mathtt{n_{el}}\}$, such that $\Omega_e\cap\Upsilon=\emptyset$ and $\Omega_e\cap\partial\Omega\neq\emptyset$, given $\boldsymbol{u}_{\mathrm{D}}$ on Γ_{D}, $\boldsymbol{t}$ on Γ_{N}, $\hat{\boldsymbol{u}}^i$ on $\Gamma\cap\partial\Omega_e^i$, find $(\boldsymbol{L}_e^i,\boldsymbol{u}_e^i,p_e^i)\in[\mathcal{H}(\mathrm{div};\Omega_e^i)]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}\times[\mathcal{H}^1(\Omega_e^i)]^{\mathtt{n_{sd}}}\times\mathcal{H}^1(\Omega_e^i)$ that satisfy

$$\begin{cases}
-\big(\boldsymbol{G},\boldsymbol{L}_e^i\big)_{\Omega_e^i}+\big(\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{G}),\boldsymbol{u}_e^i\big)_{\Omega_e^i}-\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\boldsymbol{u}_e^i\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{N}}^i}\\
\qquad=\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\boldsymbol{u}_{\mathrm{D}}\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{D}}^i}+\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\hat{\boldsymbol{u}}^i\rangle_{\partial\Omega_e^i\setminus(\Gamma_{\mathrm{D}}^i\cup\Gamma_{\mathrm{N}}^i)},\\
\big(\boldsymbol{w},\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{L}_e^i)\big)_{\Omega_e^i}+\big(\boldsymbol{w},\boldsymbol{\nabla}p_e^i\big)_{\Omega_e^i}\\
\qquad-\big\langle\boldsymbol{w},(\sqrt{\mu}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e\big\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{N}}^i}+\tau_{\mathrm{N}}h_e^{-1}\langle\boldsymbol{w},\boldsymbol{u}_e^i\rangle_{\Omega_e\cap\Gamma_{\mathrm{D}}^i}\\
\qquad+\big\langle\boldsymbol{w},\widehat{(\sqrt{\mu}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e}-(\sqrt{\mu}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e\big\rangle_{\partial\Omega_e^i\setminus\Gamma_{\mathrm{N}}^i}\\
\qquad\qquad=\big(\boldsymbol{w},\boldsymbol{s}\big)_{\Omega_e^i}+\big\langle\boldsymbol{w},\boldsymbol{t}\big\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{N}}^i}+\tau_{\mathrm{N}}h_e^{-1}\langle\boldsymbol{w},\boldsymbol{u}_{\mathrm{D}}\rangle_{\Omega_e\cap\Gamma_{\mathrm{D}}^i},\\
\big(\boldsymbol{\nabla}q,\boldsymbol{u}_e^i\big)_{\Omega_e^i}-\langle q,\boldsymbol{u}_e^i\cdot\boldsymbol{n}_e\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{N}}^i}\\
\qquad\qquad=\langle q,\boldsymbol{u}_{\mathrm{D}}\cdot\boldsymbol{n}_e\rangle_{\overline{\Omega_e}\cap\Gamma_{\mathrm{D}}^i}+\langle q,\hat{\boldsymbol{u}}^i\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e^i\setminus(\Gamma_{\mathrm{D}}^i\cup\Gamma_{\mathrm{N}}^i)},
\end{cases}$$

for all $(\boldsymbol{G},\boldsymbol{w},q)\in[\mathcal{H}(\mathrm{div};\Omega_e^i)]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}\times[\mathcal{H}^1(\Omega_e^i)]^{\mathtt{n_{sd}}}\times\mathcal{H}^1(\Omega_e^i)$, where i is either 1 or 2. When $\overline{\Omega_e}\cap\hat{\Gamma}_{\mathrm{N}}=\emptyset$, the problem is closed by adding the pressure condition,

$$\frac{1}{|\Omega_e^i|}(1,p_e^i)_{\Omega_e^i}=\rho_e,$$

with ρ_e an independent global variable representing the mean value of the pressure in Ω_e. Moreover, the Nitsche's consistent penalty constant, τ_{N}, must be large enough. In our problems $\tau_{\mathrm{N}}:=10$ proved to be sufficient. We note that, as expected, our results are not sensitive to the choice of τ_{N}.

4.3.1.2 Interface elements

For the interface elements, occupied by both immiscible fluids and such that $\Omega_e\cap\partial\Omega=\emptyset$, the interface conditions couple the element-based variables of Ω_e^1 and Ω_e^2 so that the local problem is as follows: for all $e\in\{1{:}\mathtt{n_{el}}\}$, such that $\Omega_e\cap\Upsilon\neq\emptyset$ and $\Omega_e\cap\partial\Omega=\emptyset$, given $\boldsymbol{u}_{\mathrm{D}}$ on Γ_{D}, $\boldsymbol{t}$ on Γ_{N}, $\hat{\boldsymbol{u}}^1$ on $\Gamma\cap\partial\Omega_e^1$, and $\hat{\boldsymbol{u}}^2$ on $\Gamma\cap\partial\Omega_e^2$ (i.e., $\hat{\boldsymbol{u}}$ on $\Gamma\cap\partial\Omega_e$), find $(\boldsymbol{L}_e,\boldsymbol{u}_e,p_e)\in$

$[\mathcal{H}(\mathrm{div};\Omega_e\cap\hat\Omega)]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}\times[\mathcal{H}^1(\Omega_e\cap\hat\Omega)]^{\mathtt{n_{sd}}}\times\mathcal{H}^1(\Omega_e\cap\hat\Omega)$ that satisfy

$$
\begin{cases}
-\big(\boldsymbol{G},\boldsymbol{L}_e\big)_{\Omega_e}+\big(\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{G}),\boldsymbol{u}_e\big)_{\Omega_e}-\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\boldsymbol{u}_e\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{N}}}\\
\quad-\langle\{\sqrt{\mu}\boldsymbol{G}\},\underbrace{[\![\boldsymbol{u}_e\otimes\boldsymbol{n}_e]\!]}_{=0}\rangle_{\Upsilon\cap\Omega_e}-\langle[\![\sqrt{\mu}\boldsymbol{G}\boldsymbol{n}_e]\!],\{\boldsymbol{u}_e\}\rangle_{\Upsilon\cap\Omega_e}\\
\qquad=\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\boldsymbol{u}_{\mathrm{D}}\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{D}}}+\langle\boldsymbol{G}\,\boldsymbol{n}_e,\sqrt{\mu}\,\hat{\boldsymbol{u}}\rangle_{\partial\Omega_e\setminus(\hat\Gamma_{\mathrm{D}}\cup\hat\Gamma_{\mathrm{N}})},\\
\big(\boldsymbol{w},\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{L}_e)\big)_{\Omega_e}+\big(\boldsymbol{w},\boldsymbol{\nabla}p_e\big)_{\Omega_e}\\
\qquad+\big\langle\boldsymbol{w},\widehat{(\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e}-(\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e\big\rangle_{\partial\Omega_e\setminus\hat\Gamma_{\mathrm{N}}}\\
\qquad-\big\langle\boldsymbol{w},(\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e\big\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{N}}}\\
\qquad-\big\langle\{\boldsymbol{w}\},\gamma(\boldsymbol{\nabla}\cdot\boldsymbol{n}_e^i)\boldsymbol{n}_e^i+\boldsymbol{\nabla}\gamma+[\![(\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e]\!]\big\rangle_{\Upsilon\cap\Omega_e}\\
\qquad=\big(\boldsymbol{w},\boldsymbol{s}\big)_{\Omega_e}+\langle\boldsymbol{w},\boldsymbol{t}\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{N}}},\\
\big(\boldsymbol{\nabla}q,\boldsymbol{u}_e\big)_{\Omega_e}-\langle q,\boldsymbol{u}_e\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{N}}}\\
\quad-\langle[\![q\,\boldsymbol{n}_e]\!],\{\boldsymbol{u}_e\}\rangle_{\Upsilon\cap\Omega_e}-\langle\{q\},\overbrace{[\![\boldsymbol{u}_e\cdot\boldsymbol{n}_e]\!]}^{=0}\rangle_{\Upsilon\cap\Omega_e}\\
\qquad=\langle q,\boldsymbol{u}_{\mathrm{D}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\cap\hat\Gamma_{\mathrm{D}}}+\langle q,\hat{\boldsymbol{u}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\setminus(\hat\Gamma_{\mathrm{D}}\cup\hat\Gamma_{\mathrm{N}})},
\end{cases}
$$

for all $(\boldsymbol{G},\boldsymbol{w},q)\in[\mathcal{H}(\mathrm{div};\Omega_e\cap\hat\Omega)]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}\times[\mathcal{H}^1(\Omega_e\cap\hat\Omega)]^{\mathtt{n_{sd}}}\times\mathcal{H}^1(\Omega_e\cap\hat\Omega)$, where the interface conditions (4.3b) have been imposed such that symmetry is preserved. Moreover, the *mean* $\{\odot\}$ operator has been introduced following the definition by [98], such that, along each portion of the interface Υ, it sums the weighted values from Ω_e^1 and Ω_e^2, namely

$$
\{\odot\}=\frac{1}{2}(\odot|_{\Omega_e^1}+\odot|_{\Omega_e^2}),\tag{4.4}
$$

see for instance [75]. Also, the following three identities have been used:

$$
\begin{aligned}
\sum_{i\in\{1:2\}}\langle\boldsymbol{G}\,\boldsymbol{n}_e^i,\sqrt{\mu^i}\,\boldsymbol{u}^i\rangle_\Upsilon&=\int_\Upsilon\sqrt{\mu^1}u_i^1G_{ij}^1n_j^1+\sqrt{\mu^2}u_i^2G_{ij}^2n_j^2\,d\Upsilon\\
&=\int_\Upsilon\Big([\![\boldsymbol{u}\otimes\boldsymbol{n}_e]\!]:\{\sqrt{\mu}\boldsymbol{G}\}+\{\boldsymbol{u}\}\cdot[\![\sqrt{\mu}\boldsymbol{G}\boldsymbol{n}_e]\!]\Big)d\Upsilon\\
&=\langle[\![\boldsymbol{u}\otimes\boldsymbol{n}_e]\!],\{\sqrt{\mu}\boldsymbol{G}\}\rangle_\Upsilon+\langle\{\boldsymbol{u}\},[\![\sqrt{\mu}\boldsymbol{G}\boldsymbol{n}_e]\!]\rangle_\Upsilon,\\
\sum_{i\in\{1:2\}}\langle\boldsymbol{w}^i,(\sqrt{\mu^i}\boldsymbol{L}_e^i+p_e^i\boldsymbol{I}_{\mathtt{n_{sd}}})\,\boldsymbol{n}_e^i\rangle_\Upsilon&=\langle[\![\boldsymbol{w}\otimes\boldsymbol{n}_e]\!],\{\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}}\}\rangle_\Upsilon+\langle\{\boldsymbol{w}\},[\![(\sqrt{\mu}\boldsymbol{L}_e+p_e\boldsymbol{I}_{\mathtt{n_{sd}}})\boldsymbol{n}_e]\!]\rangle_\Upsilon,\\
\sum_{i\in\{1:2\}}\langle q^i,\boldsymbol{u}^i\cdot\boldsymbol{n}_e^i\rangle_\Upsilon&=\langle[\![q\,\boldsymbol{n}_e]\!],\{\boldsymbol{u}\}\rangle_\Upsilon+\langle\{q\},[\![\boldsymbol{u}\cdot\boldsymbol{n}_e]\!]\rangle_\Upsilon.
\end{aligned}
$$

Rearranging terms, the local weak problem becomes:

$$\begin{cases} -\big(\boldsymbol{G},\boldsymbol{L}_e\big)_{\Omega_e} \\ \quad +\big(\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{G}),\boldsymbol{u}_e\big)_{\Omega_e} - \langle \boldsymbol{G}\,\boldsymbol{n}_e, \sqrt{\mu}\,\boldsymbol{u}_e\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{N}}} - \langle [\![\sqrt{\mu}\boldsymbol{G}\boldsymbol{n}_e]\!], \{\boldsymbol{u}_e\}\rangle_{\Upsilon\cap\Omega_e} \\ \qquad = \langle \boldsymbol{G}\,\boldsymbol{n}_e, \sqrt{\mu}\,\boldsymbol{u}_{\mathrm{D}}\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{D}}} + \langle \boldsymbol{G}\,\boldsymbol{n}_e, \sqrt{\mu}\,\hat{\boldsymbol{u}}\rangle_{\partial\Omega_e\setminus(\hat{\Gamma}_{\mathrm{D}}\cup\hat{\Gamma}_{\mathrm{N}})}, \\ \\ \big\langle \boldsymbol{w}, \tau\,\boldsymbol{u}_e\big\rangle_{\partial\Omega_e\setminus\hat{\Gamma}_{\mathrm{N}}} \\ \quad +\big(\boldsymbol{w},\boldsymbol{\nabla}\cdot(\sqrt{\mu}\boldsymbol{L}_e)\big)_{\Omega_e} - \big\langle \boldsymbol{w}, \sqrt{\mu}\boldsymbol{L}_e\,\boldsymbol{n}_e\big\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{N}}} - \big\langle \{\boldsymbol{w}\}, [\![\sqrt{\mu}\boldsymbol{L}_e\,\boldsymbol{n}_e]\!]\big\rangle_{\Upsilon\cap\Omega_e} \\ \qquad +\big(\boldsymbol{w},\boldsymbol{\nabla}p_e\big)_{\Omega_e} - \big\langle \boldsymbol{w}, p_e\,\boldsymbol{n}_e\big\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{N}}} - \big\langle \{\boldsymbol{w}\}, [\![p_e\,\boldsymbol{n}_e]\!]\big\rangle_{\Upsilon\cap\Omega_e} \\ \qquad\quad = \big(\boldsymbol{w},\boldsymbol{s}\big)_{\Omega_e} + \big\langle \boldsymbol{w}, \boldsymbol{t}\big\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{N}}} + \big\langle \boldsymbol{w}, \tau\,\boldsymbol{u}_{\mathrm{D}}\big\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{D}}} \\ \qquad\qquad + \big\langle \boldsymbol{w}, \tau\,\hat{\boldsymbol{u}}\big\rangle_{\partial\Omega_e\setminus(\hat{\Gamma}_{\mathrm{N}}\cup\hat{\Gamma}_{\mathrm{D}})} + \big\langle \{\boldsymbol{w}\}, \gamma(\boldsymbol{\nabla}\cdot\boldsymbol{n}_e^1)\boldsymbol{n}_e^1 + \boldsymbol{\nabla}\gamma\big\rangle_{\Upsilon\cap\Omega_e} \\ \\ \big(\boldsymbol{\nabla}q,\boldsymbol{u}_e\big)_{\Omega_e} - \langle q, \boldsymbol{u}_e\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{N}}} - \langle [\![q\,\boldsymbol{n}_e]\!], \{\boldsymbol{u}_e\}\rangle_{\Upsilon\cap\Omega_e} \\ \qquad = \langle q, \boldsymbol{u}_{\mathrm{D}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\cap\hat{\Gamma}_{\mathrm{D}}} + \langle q, \hat{\boldsymbol{u}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e\setminus(\hat{\Gamma}_{\mathrm{D}}\cup\hat{\Gamma}_{\mathrm{N}})}, \end{cases} \tag{4.5}$$

which induces a symmetric problem. As before, when $\partial\Omega_e \cap \hat{\Gamma}_{\mathrm{N}} = \emptyset$, the problem is closed by adding the pressure condition,

$$\frac{1}{|\Omega_e|}\sum_{i\in\{1:2\}} (p_e^i, 1)_{\Omega_e\cap\Omega^i} = \rho_e.$$

Remark 4.4 (Viscosity-robust mean operator). The mean operator defined in (4.4) is not robust with respect to high-contrast viscosity coefficients (see [22], [77]). When μ^1 and μ^2 are such that $\mu^1 \ll \mu^2$, or vice versa, we propose the weighted mean

$$\{\odot\} = \omega_e^1 \odot |_{\Omega_e^1} + \omega_e^2 \odot |_{\Omega_e^2},$$

where the weights are defined as

$$\omega_e^1 := \frac{\mu_e^2}{\mu_e^1 + \mu_e^2}, \qquad \omega_e^2 := \frac{\mu_e^1}{\mu_e^1 + \mu_e^2}.$$

Remark 4.5 (General cut elements). Suppose an element Ω_e is cut into a variable number of regions. The local problems for immersed boundary and interface elements can be generalized by doubling the unknowns in each region occupied by a fluid and coupling the regions sharing an interface.

4.3.2 Global problem

In the previous section, for each element Ω_e, $e \in \{1{:}\mathtt{n_{el}}\}$, and all the fluids $i \in \{1,2\}$, the local variables $(\boldsymbol{L}_e^i, \boldsymbol{u}_e^i, p_e^i)$ are expressed in terms of the global unknowns $(\hat{\boldsymbol{u}}^i, \rho_e)$. The global problem is represented by the *transmission conditions* (4.3c). By definition, $\hat{\boldsymbol{u}}^i$ is uniquely defined on each face of Γ thus condition $[\![\boldsymbol{u}^i \otimes \boldsymbol{n}^i]\!] = \boldsymbol{0}$ is automatically satisfied. In addition to the transmission conditions, for all $e \in \{1{:}\mathtt{n_{el}}\}$ such that $\overline{\Omega_e} \cap \hat{\Gamma}_{\mathrm{N}} = \emptyset$, we consider the following

compatibility condition,

$$\sum_{i\in\{1:2\}} \left\{ \langle \hat{\boldsymbol{u}}^i \cdot \boldsymbol{n}_e, 1\rangle_{\partial\Omega_e^i \setminus \hat{\Gamma}_{\mathrm{D}}} + \langle \boldsymbol{u}_{\mathrm{D}} \cdot \boldsymbol{n}_e, 1\rangle_{\partial\Omega_e^i \cap \hat{\Gamma}_{\mathrm{D}}} \right\} = 0,$$

derived from the divergence-free condition, where we have used the equality $\langle \boldsymbol{u}^r \boldsymbol{n}_e^r - \boldsymbol{u}^l \boldsymbol{n}_e^l, 1\rangle_{\Upsilon\cap\Omega_e} = 0$. The weak formulation is as follows: given $\boldsymbol{u}_{\mathrm{D}}$ on Γ_{D}, $\boldsymbol{t}$ on Γ_{N}, $(\boldsymbol{L}_e^i, \boldsymbol{u}_e^i, p_e^i) \in [\mathcal{H}(\mathrm{div}; \Omega_e \cap \hat{\Omega})]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}} \times [\mathcal{H}^1(\Omega_e \cap \hat{\Omega})]^{\mathtt{n_{sd}}} \times \mathcal{H}^1(\Omega_e \cap \hat{\Omega})$, for all $i \in \{1,2\}$, find $(\hat{\boldsymbol{u}}^i, \rho_e) \in [\mathcal{H}^{\frac{1}{2}}(\Gamma \cap \partial\Omega_e^i)]^{\mathtt{n_{sd}}} \times \mathbb{R}^{\mathtt{n_{sd}}}$ that satisfy

$$\begin{aligned} \sum_{e\in\{1:\mathtt{n_{el}}\}} \sum_{i\in\{1:2\}} \Big\{ & \langle \hat{\boldsymbol{w}}, (\sqrt{\mu^i}\boldsymbol{L}_e^i + p_e^i \boldsymbol{I}_{\mathtt{n_{sd}}})\, \boldsymbol{n}_e \rangle_{\partial\Omega_e^i \setminus (\hat{\Gamma}_{\mathrm{D}} \cup \hat{\Gamma}_{\mathrm{N}})} \\ & + \langle \hat{\boldsymbol{w}}, \tau \boldsymbol{u}_e^i \rangle_{\partial\Omega_e^i \setminus (\hat{\Gamma}_{\mathrm{D}} \cup \hat{\Gamma}_{\mathrm{N}})} - \langle \hat{\boldsymbol{w}}, \tau \hat{\boldsymbol{u}}^i \rangle_{\partial\Omega_e^i \setminus (\hat{\Gamma}_{\mathrm{D}} \cup \hat{\Gamma}_{\mathrm{N}})} \Big\} = 0, \end{aligned} \tag{4.6}$$

for all $\hat{\boldsymbol{w}} \in [\mathcal{L}_2(\Gamma \cap \partial\Omega_e^i)]^{\mathtt{n_{sd}}}$, and

$$\sum_{i\in\{1:2\}} \left\{ \langle \hat{\boldsymbol{u}}^i \cdot \boldsymbol{n}_e, 1\rangle_{\partial\Omega_e^i \setminus \hat{\Gamma}_{\mathrm{D}}} + \langle \boldsymbol{u}_{\mathrm{D}} \cdot \boldsymbol{n}_e, 1\rangle_{\partial\Omega_e^i \cap \hat{\Gamma}_{\mathrm{D}}} \right\} = 0,$$

for all $e \in \{1:\mathtt{n_{el}}\}$ such that $\overline{\Omega_e} \cap \hat{\Gamma}_{\mathrm{N}} = \emptyset$. Note that the global problem is not directly affected by interface terms. Remind that when $\hat{\Gamma}_{\mathrm{N}} = \emptyset$, the additional pressure condition, presented in Remark 4.2, is required.

4.4 Domain definition and quadratures in the interface elements

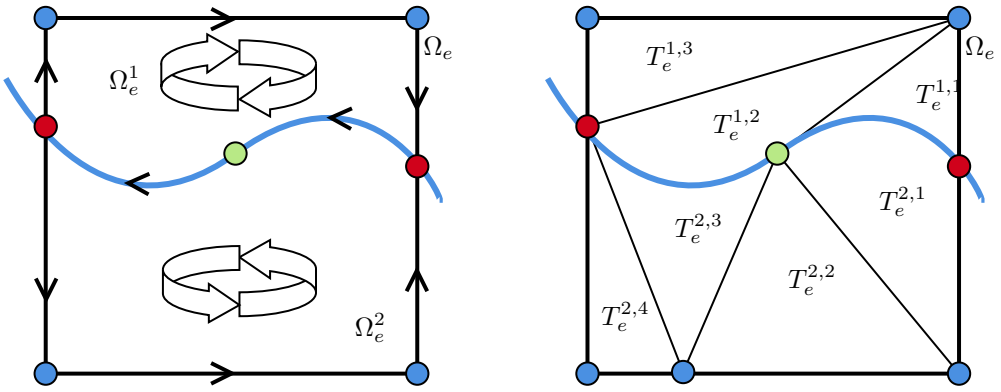
The knowledge of the NURBS interface (4.1) does not allow for a straightforward determination of the computational domain Ω^i, $i \in \{1,2\}$. Let us assume that the external boundary $\partial\Omega$ is aligned with the mesh so that we can focus only on the interface Υ. Moreover, let us assume that the interface Υ is closed, orientable, and does not touch the boundary $\partial\Omega$ so that we denote by Ω^1 the interior subdomain and by Ω^2 the exterior subdomain. The same procedure can be generalized for unfitted $\partial\Omega$. The following method takes inspiration from the work presented in [99].

To exactly define Ω^i, the idea is to decompose the problem element-wise by identifying for each element the regions Ω_e^i, such that $\Omega^i = \bigcup_{e\in\{1:\mathtt{n_{el}}\}} \Omega_e^i$, for all $i \in \{1,2\}$. Thus, we proceed by locating the intersection points between the internal skeleton Γ and Υ so that Υ_e is the portion of Υ inside an element Ω_e, i.e., $\Upsilon_e := \Upsilon \cap \Omega_e$. Note that by definition, Υ_e is a continuous curve; otherwise, Ω_e would be split into more than two regions. Moreover, note that Υ_e can be composed of several NURBS curves, that is

$$\Upsilon_e := \bigcup_{j\in\mathcal{J}_e} \Upsilon_e^j := \boldsymbol{C}_\Upsilon^j([\lambda_1^j, \lambda_2^j]), \tag{4.7}$$

with $\mathcal{J}_e$ the set of coefficients j of the NURBS curves $\boldsymbol{C}_\Upsilon^j$ inside Ω_e, and λ_1^j and λ_2^j the extremes of $\boldsymbol{C}_\Upsilon^j$ in Ω_e such that $\Upsilon_e^j = \boldsymbol{C}_\Upsilon^j([\lambda_1^j, \lambda_2^j])$. Owing to the assumption of orientability, Υ_e is a

directed curve. Then, a left and a right side of Υ_e exist, and depending on the orientation of Υ_e, we determine the fluid index i of the adjacent regions. For instance, when Υ is clockwise oriented, the right regions belong to Ω^1, the left to Ω^2, and vice versa when Υ is counter-clockwise oriented. We schematize the procedure in Figure 4.3a. The assignment of materials to the remaining uncut elements is determined iteratively, starting with the elements for which materials have already been defined and subsequently extending to elements that share faces, inheriting the material of the shared face.



(a) Construction of regions Ω_e^i, for all $i \in \{1, 2\}$.

(b) Quadrature triangulation of regions Ω_e^i, for all $i \in \{1, 2\}$.

Figure 4.3: Example of a square element Ω_e cut by the interface Υ into two regions. The interface is composed of two NURBS curves distinguished by the green bullet. Left figure: Decomposition of Ω_e in the regions Ω_e^1 and Ω_e^2. Right figure: Decomposition of Ω_e^i in quadrature triangles, for all $i \in \{1, 2\}$. The red bullets represent the intersection points between Υ and $\partial\Omega_e$, and the blue bullets represent the vertices of the element and of the quadrature triangles lying on the mesh skeleton.

Specific numerical techniques are used to integrate along cut faces, along NURBS curves, and in cut elements, whereas uncut faces and uncut elements employ the standard HDG quadrature.

The quadrature along cut faces does not present particular complications by mapping the Gauss–Legendre nodes onto the face region.

The quadrature along NURBS curves is treated element-wise. We define the quadrature along the portion of interface Υ_e as

$$\int_{\Upsilon_e} f \mathrm{d}l = \sum_{j \in \mathcal{J}_e} \int_{\lambda_1^j}^{\lambda_2^j} f(\boldsymbol{C}_\Upsilon^j(\lambda)) \| J_{\boldsymbol{C}_\Upsilon^j}(\lambda) \| \mathrm{d}\lambda,$$

with $\| J_{\boldsymbol{C}_\Upsilon^j} \|$ the norm of the differential of the NURBS curve $\boldsymbol{C}_\Upsilon^j$ which describes Υ_e^j, and f a generic scalar function defined along Υ_e. The integrals are approximated using the 1D Gauss–Legendre quadrature defined in the parametric space $[\lambda_1^j, \lambda_2^j]$ and mapped along $\boldsymbol{C}_\Upsilon^j$. The same procedure is realized for unfitted external boundaries.

The quadrature within a generic region Ω_e^i involves multiple steps. Firstly, the region Ω_e^i undergoes a partitioning process, guided by Lee's visibility algorithm [86], which is essential to ensure the accuracy of the quadrature, as emphasized in [118] for fitted triangular meshes. It is noteworthy that Lee's algorithm necessitates polygonal regions. Thus, we approximate Ω_e^i

linearly by introducing a sufficient number of points along Υ_e to describe the curve accurately (e.g., the number of points proportional to the inverse of the curvature of Υ_e). Once the visibility subregions are determined, we partition each subregion into triangles reconstructing the initial region Ω_e^i,

$$\Omega_e^i = \bigcup_{l \in \{1:\mathtt{n}_{\mathrm{tri,e}}^i\}} T_e^{i,l},$$

with $\mathtt{n}_{\mathrm{tri,e}}^i$ the total number of triangles composing Ω_e^i. The procedure is detailed in Appendix 5.A. The triangulation consists of two types of triangles: affine and curved. The curved triangles are constructed with, at maximum, one curved side described by at most one NURBS curve. Note that the exact NURBS curve description is used instead of linear approximations, which serves solely for visibility checks. Then, given f a generic scalar function defined in Ω_e^i, we calculate the quadrature in Ω_e^i as the sum of the quadrature in each triangle,

$$\int_{\Omega_e^i} f(x,y)\mathrm{d}x\mathrm{d}y = \sum_{l \in \{1:\mathtt{n}_{\mathrm{tri,e}}^i\}} \int_{T_e^{i,l}} f(x,y)\mathrm{d}x\mathrm{d}y.$$

When $T_e^{i,l}$ is affine, we employ the standard Gauss–Legendre quadrature, when $T_e^{i,l}$ is curved, as detailed in [115], we constructed the Gauss–Legendre quadrature in the special tensor space $R := [\lambda_1^j, \lambda_2^j] \times [0,1]$ so that is mapped into $T_e^{i,l}$ by means of the affine mapping $\boldsymbol{\psi}$,

$$\boldsymbol{\psi}(\lambda,\theta) = \boldsymbol{C}^j(\lambda)(1-\theta) + \theta \boldsymbol{x}_3, \qquad \forall \lambda \in [\lambda_1^j, \lambda_2^j], \qquad \forall \theta \in [0,1],$$

with $\boldsymbol{x}_3$ the triangle vertex that does not lie on the NURBS curve. Then,

$$\int_{T_e^{i,l}} f(x,y)\mathrm{d}x\mathrm{d}y = |J_{\boldsymbol{\psi}}| \int_R f(\psi(\lambda,\theta))\mathrm{d}\lambda\mathrm{d}\theta,$$

with $|J_{\boldsymbol{\psi}}|$ the Jacobian of $\boldsymbol{\psi}$, which is constant owing to the linearity of $\boldsymbol{\psi}$.

The algorithm of the aforementioned method and the generalization for elements divided into more than two regions are detailed in Appendix 5.B.

4.5 Computational details for high-fidelity simulations

In this section, we present a local, element-wise postprocessing technique that enables a superconvergent velocity approximation. We employ this approach to formulate a degree-adaptive strategy. Additionally, we outline an element extension method to ensure robustness with respect to badly cut elements, which are elements cut by the interface or boundary such that only a small portion of the volume lies within the computational domain.

4.5.1 Degree adaptivity

Following [37], for all $i \in \{1,2\}$, we introduce the discrete functional space

$$\mathcal{V}_\star^h(\hat{\Omega}) := \left\{ v \in \mathcal{L}_2(\hat{\Omega}) \mid \ v|_{\Omega_e^i} \in \mathcal{P}^{k+1}(\Omega_e^i),\ \forall e \in \{1:\mathtt{n}_{\mathtt{el}}\} \right\}, \tag{4.8}$$

composed of polynomial functions of order at most $(k+1)$. Then, the accuracy of the velocity field approximation can be enhanced by solving an additional, computationally efficient element-wise problem. For all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1,2\}$, find the superconvergent velocity field $\boldsymbol{u}_\star \in [\mathcal{V}_\star^h(\hat{\Omega})]^{\mathtt{n_{sd}}}$ such that

$$\begin{cases} -\boldsymbol{\nabla}\cdot(\mu^i\boldsymbol{\nabla}\boldsymbol{u}^i_{\star,e}) = \boldsymbol{\nabla}\cdot(\sqrt{\mu^i}\boldsymbol{L}^i_e) & \text{in } \Omega^i_e, \\ \mu^i\boldsymbol{\nabla}\boldsymbol{u}^i_{\star,e}\cdot\boldsymbol{n}_e = -\sqrt{\mu^i}\boldsymbol{L}^i_e\cdot\boldsymbol{n}_e & \text{on } \partial\Omega^i_e \cup (\Upsilon\cap\Omega_e), \end{cases} \tag{4.9}$$

and such that the mean superconvergent velocity satisfies

$$(\boldsymbol{u}^i_{\star,e},1)_{\Omega_e\cap\Omega^i} = (\boldsymbol{u}^i_e,1)_{\Omega^i_e}. \tag{4.10}$$

Hence, the standard HDG weak formulation is recovered in each fluid, namely

$$\begin{aligned} (\boldsymbol{\nabla}\boldsymbol{w}^i_\star,\mu^i\boldsymbol{\nabla}\boldsymbol{u}^i_{\star,e})_{\Omega^i_e} &= -(\boldsymbol{\nabla}\boldsymbol{w}^i_\star,\sqrt{\mu^i}\boldsymbol{L}^i_e)_{\Omega^i_e}, \\ (\boldsymbol{u}^i_{\star,e},1)_{\Omega^i_e} &= (\boldsymbol{u}^i_e,1)_{\Omega^i_e}, \end{aligned} \tag{4.11}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1,2\}$.

Consistently with [117], we employ the following element-wise measure of the error, that is

$$E_e := \max_{i\in\{1,2\}} \frac{1}{\|\boldsymbol{u}^i_{\star,e}\|_2}\Big((\boldsymbol{u}^i_{\star,e}-\boldsymbol{u}^i_e,\boldsymbol{u}^i_{\star,e}-\boldsymbol{u}^i_e)_{\Omega^i_e}\Big)^{1/2}, \qquad \text{for } e \in \{1{:}\mathtt{n_{el}}\}, \tag{4.12}$$

where $\|\odot\|_2$ denotes the $\mathcal{L}^2$-norm of a scalar-, vector-, or tensor- valued function defined in S.

By following the strategy depicted in [117], given an arbitrary desired accuracy ε, we ensure the error to be lower than such accuracy by locally adapting the degree of approximation. We solve iteratively problem (4.2) by increasing/decreasing, for each element Ω_e, the local degree of approximation of

$$\delta k_e = \Big\lceil \frac{\log(\varepsilon/E_e)}{\log(h_e)} \Big\rceil, \tag{4.13}$$

with $\lceil\cdot\rceil$ the ceiling function. The procedure ends when $\delta k_e = 0$, for all $e \in \{1{:}\mathtt{n_{el}}\}$.

4.5.2 Element extension

It is well-known that unfitted methods suffer in the presence of elements cut by the interface or boundary such that the portion of the computational volume is small with respect to the volume of the whole element. We define the parameter

$$\alpha^i_e := \frac{|\Omega^i_e|}{|\Omega_e|}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}, \qquad \forall i \in \{1,2\}, \tag{4.14}$$

as the ratio between the volume of the computational region Ω^i_e and the volume of whole element Ω_e in the domain. We denote by $\alpha := \min_{e,i}\alpha^i_e$ the smallest cut. An element is said badly cut when $\alpha^i_e < \alpha_{\min}$, with $\alpha_{\min}$ a user-specified variable set to $\alpha_{\min} := 0.3$, consistently with the literature, e.g. [20], [22].

To address badly cut elements, we use an element extension method inspired by [99]. The idea is to join badly cut elements with a neighbor element to eliminate the bad cut. Given a

badly cut element, the method erases the element-based unknowns of such element and extends the quadrature of the selected neighbor element onto the badly cut element. Additionally, the extended element inherits the face-based unknowns of the badly cut element. If a face is shared between the two elements, its face-based unknowns are erased, e.g., see Figure 4.4. For each

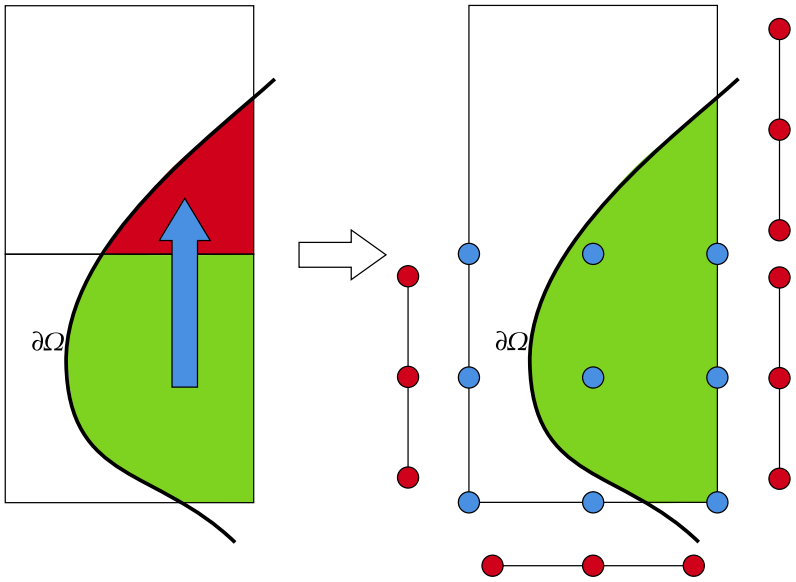


Figure 4.4: Illustration of the element extension strategy for an element badly cut by the external boundary. Left figure: initial configuration with badly cut region (red) and well-cut region (green) selected for the expansion. Right figure: the result of the element extension. The two elements are joined in one unique element where the computational region is highlighted in green. The blue bullets represent the element-based unknowns, and the red bullets represent the face-based unknowns of the resulting element extension.

badly cut element, we assess potential extensions based on three factors: (i) the combined area, (ii) the Euclidean distance between element centers, and (iii) the number of previously extended elements connected to it. The element with the highest combined score of these factors is extended onto the badly cut element. In two-fluid problems, we apply a stronger penalty to discourage multiple extensions per element owing to inter-region coupling, thus minimizing the local problem size.

Chapter 5

HDG for microfluid dynamics problems

In this chapter, we test the unfitted HDG-NEFEM method for different one- and two-fluid problems. In Section 5.1, we validate the method, and we obtain optimal error decay and condition number growth. Moreover, we study the effects of badly cut faces and badly cut cells. Then, in Sections 5.2 and 5.3, we solve two verification test cases for one- and two-fluid problems, respectively. In the former, the external boundary, and in the latter, the interface does not fit the mesh. Finally, in Sections 5.4 and 5.5, we solve two physical meaningful problems where the external boundary and the interface assume more complex configurations.

5.1 Numerical validation

Hybrid methods pose unknowns on both elements and faces. Thus, the unfitted mesh approach can generate, as well as the classical badly cut elements, badly cut faces. Following the HDG decomposition into local and global problems, we consider a one-fluid problem, and we analyze the effects of bad cuts on the error and on the conditioning of local and global matrices. The condition number is computed using the Euclidean norm, and we name κ^{loc} and κ^{glob}, the conditioning of the HDG local and global matrices, respectively. Consistently with the element ratio α_e^i defined in (4.14), we define for each face $\Gamma_f \subset \Gamma$,

$$\beta_f^i := \frac{|\Gamma_f \cap \partial\Omega^i|}{|\Gamma_f|}, \qquad f \in \{1{:}\mathtt{n_{fc}}\}, \quad i \in \{1,2\}, \tag{5.1}$$

the ratio between the area of the computational face Γ_f^i and the area of the whole face Γ_f. We denote by $\beta := \min_{f,i} \beta_f^i$ the smallest cut.

Let us consider the one-fluid case problem with only Dirichlet boundary conditions, such that $\Omega^1 := \Omega$, $\Omega^2 := \emptyset$, $\Gamma_{\text{D}} := \partial\Omega$, $\Gamma_{\text{N}} := \emptyset$. Additionally, we set $\mu := 1$,

$$\boldsymbol{s} := \boldsymbol{\nabla} p - \boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} \boldsymbol{u}),$$

and $\boldsymbol{u}_{\mathrm{D}} := \boldsymbol{u}^{\mathrm{ref}}$, where the exact analytical solutions are

$$\boldsymbol{u}^{\mathrm{ref}} := \begin{pmatrix} x^2(1-x)^2(2y-6y^2+4y^3) \\ -y^2(1-y)^2(2x-6x^2+4x^3) \end{pmatrix}, \qquad p^{\mathrm{ref}} := x(1-x). \tag{5.2}$$

5.1.1 Badly cut faces

Let us consider the M-shaped domain Ω in Figure 5.1 using the mesh composed of 4×4 square elements partitioning exactly the square $\Omega_{\mathrm{o}} = (0,1)^2$. The M-shaped domain is defined such that all the elements and faces are well-cut, but the top vertical face of length ε. We study the influence of the badly cut face on the global conditioning for various $\varepsilon \in \{0.005, 0.05, 0.1, 0.15\}$ so that we obtain $\beta \in \{2\%, 20\%, 40\%, 60\%\}$ of the whole face while maintaining well-cut cells with $\alpha > 50\%$. Figure 5.2 displays a comparison between Lagrange (using Fekete nodes) and Legendre face basis functions in the reference face $\hat{\Gamma} := [-1,1]$. Lagrange basis functions belong to the category of nodal bases, which means that they are composed of polynomial functions, denoted as $\{b_j(\boldsymbol{x})\}_{j\in\{0:k\}}$, satisfying $b_j(\boldsymbol{x}_i) = \delta_{ij}$, with $\{\boldsymbol{x}_i\}_{i\in\{0:k\}}$ a set of nodes, k the degree of approximation, and δ_{ij} the Kronecker symbol. Consequently, a polynomial function centered at $\boldsymbol{x}_i$ (i.e., $b_j(\boldsymbol{x}_i) = 1$) is equal to 0 at all nodes except $\boldsymbol{x}_i$. When we integrate a Lagrange interpolating function over $\hat{\Gamma}$, we observe that as the polynomial order k increases, the contribution of each polynomial function to the quadrature becomes concentrated in the vicinity of its center. As a result, when a node on a cut face is cut out of the computational domain, the polynomial function centered at such node provides small contributions, which tend towards zero as we increase the polynomial order k of the basis. This phenomenon can lead to ill-conditioning. On the contrary, this phenomenon is less pronounced for modal basis functions, such as Legendre bases, where the contribution of each polynomial function is more evenly distributed throughout the entire domain. In our view, this distinction is the primary reason for the enhanced robustness exhibited by Legendre face basis functions when confronted with badly cut faces.

Figure 5.3 shows the global condition number κ^{glob} solving problem (5.2) comparing Lagrange and Legendre face basis functions for different polynomial orders $k \in \{1{:}4\}$. We use 2D Lagrange tensor basis functions, $\mathcal{P}^k_{\mathrm{La}}$, with Fekete nodes for the element-based unknowns. Note that this choice leads to optimal conditioning of standard HDG (uncut) elements. Using Legendre face basis functions, κ^{glob} appears independent of β. In Section 5.2, we prove numerically that the global condition number κ^{glob} grows like $\mathcal{O}(h^{-2})$ for $h \to 0$ achieving the optimal rate for second-order elliptic differential operators. Henceforth, we set the Legendre polynomials as default face basis functions.

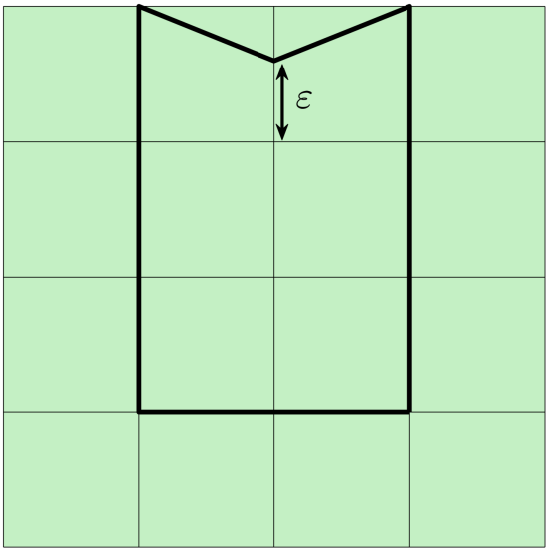
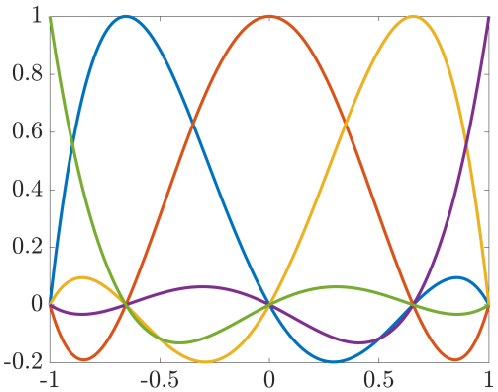
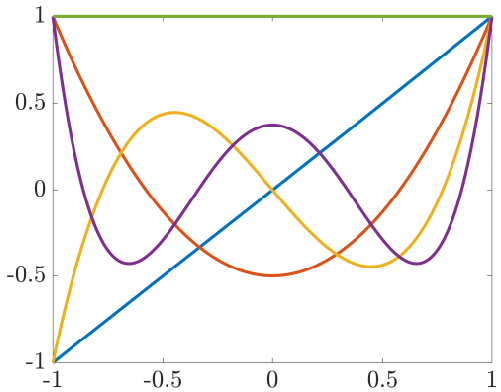


Figure 5.1: M-shaped domain with $\varepsilon = 0.15$ ($\beta = 60\%$ of the whole face). The domain is a rectangle $(0.25, 0.75) \times (0.25, 1)$ where the top side has been divided into two segments connected in the point $(0.5, 0.75 + \varepsilon)$. The mesh is composed of 4×4 square elements partitioning exactly the square $\Omega_o = (0, 1)^2$.



(a) 1D Lagrange basis functions $\hat{\mathcal{P}}^4_{\text{La}}$.



(b) 1D Legendre basis functions $\hat{\mathcal{P}}^4_{\text{Le}}$.

Figure 5.2: Example of nodal (Lagrange with Fekete nodes) and modal (Legendre) face basis functions in the reference face $\hat{\Gamma} = [-1, 1]$ of order $k = 4$.

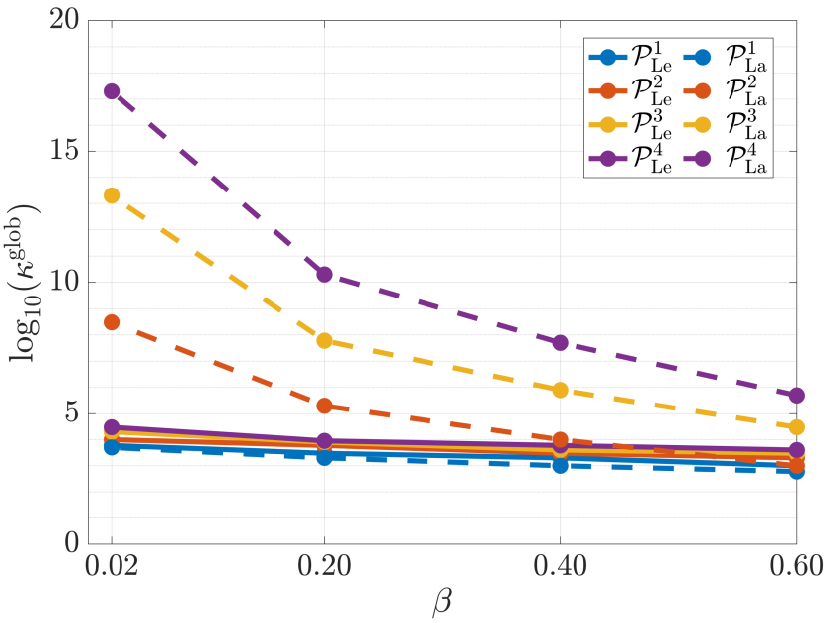
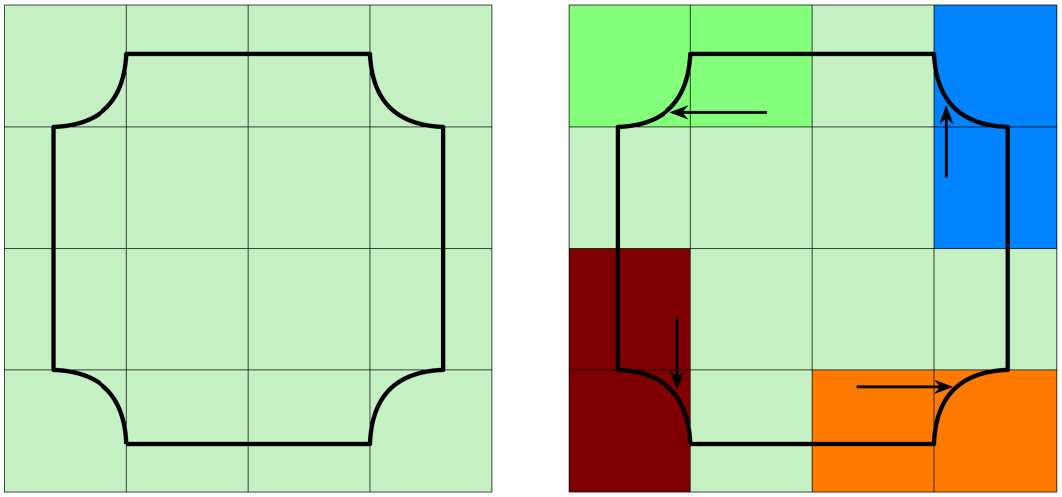


Figure 5.3: M-shaped domain with $\varepsilon \in \{0.15, 0.1, 0.05, 0.005\}$ ($\beta \in \{0.60, 0.40, 0.20, 0.02\}$). Comparison of $\log_{10}(\kappa^{\mathrm{glob}})$ with respect to β using Legendre $\hat{\mathcal{P}}^k_{\mathrm{Le}}$ (continuous lines) and Lagrange $\hat{\mathcal{P}}^k_{\mathrm{La}}$ (dotted lines) face basis functions. The basis orders are $k = 1$ (blue lines), $k = 2$ (red lines), $k = 3$ (yellow lines), and $k = 4$ (purple lines). The mesh is composed of 4×4 square elements.

5.1.2 Badly cut cells

Let us check the influence of the badly cut cells on the HDG local formulation. To counter the adverse effects of the badly cut cells, we use the element extension strategy, presented in Section 4.5.2. We consider the smoothed square domain Ω in Figure 5.4, using the mesh composed of 4×4 square elements exactly partitioning the square $\Omega_o = (0,1)^2$. In this configuration, all the faces are well-cut, $\beta > 60\%$, and the elements at the corners are badly-cut, $\alpha = 6\%$. Note that the well-cut elements have an element ratio equal to 60%).

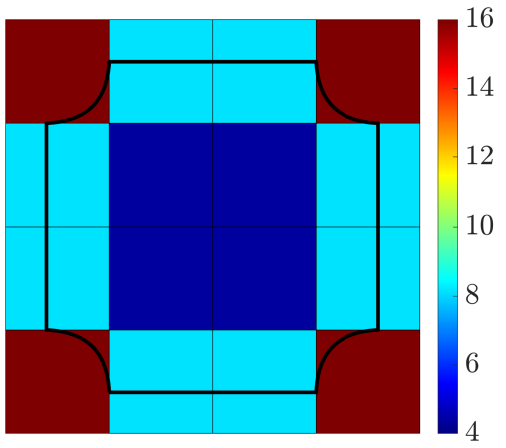


(a) Smoothed square domain.

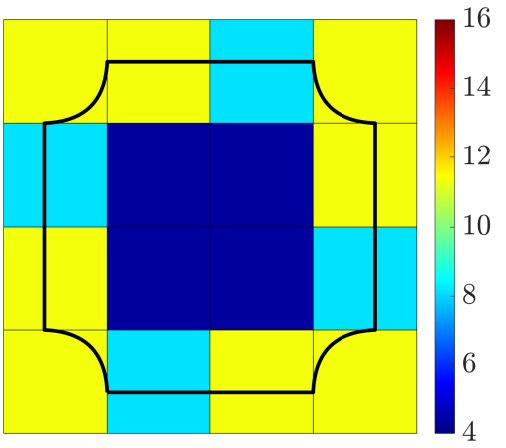
(b) The element extension agglomeration procedure.

Figure 5.4: Smoothed square domain. The domain is a square $(0.1, 0.9)^2$ where the angles have been substituted with an arc of circumference of radius 0.15. The mesh is composed of 4×4 square elements partitioning exactly the square $\Omega_o = (0,1)^2$.

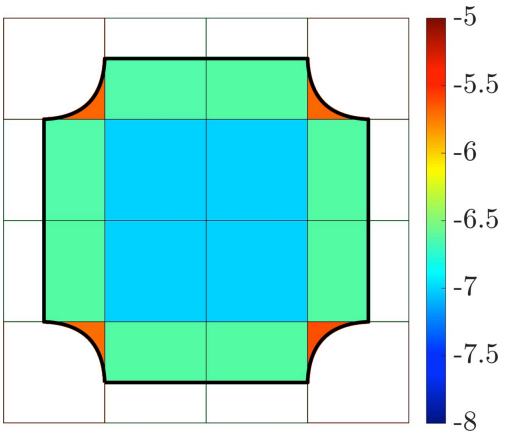
Figures 5.5a and 5.5b display the local conditioning κ^{loc} for the HDG formulation without and with element extension. Consistently, Figure 5.5c and 5.5d display the element-wise L^2-velocity error. We employ the Lagrange basis $\mathcal{P}^4_{\text{La}}$ and Legendre basis $\hat{\mathcal{P}}^4_{\text{Le}}$ for the element- and face-based unknowns, respectively. We notice that element extension improves numerical error by countering the effects of the badly cut cells, even though it is insufficient to mitigate local conditioning. Indeed, extrapolation has some drawbacks on the local condition number of the extended element: comparing Figures 5.5a and 5.5b, one can notice that the 6% extrapolation worsens κ^{loc} of about 2 orders of magnitude compared to its initial value. Different basis functions, such as the tensor Legendre basis, have been investigated as alternatives. We observed an enhancement in the local conditioning, but not significant enough to opt for such Legendre basis.



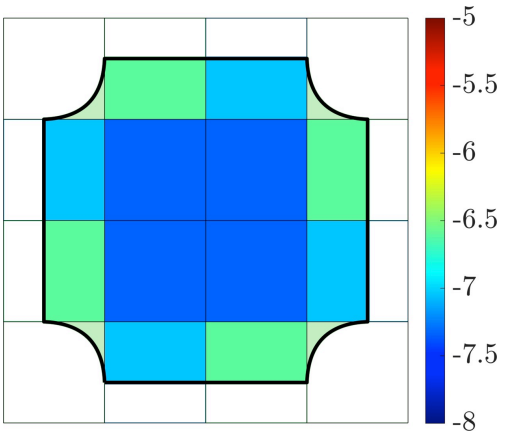
(a) Conditioning $\log_{10}(\kappa^{\text{loc}})$ without robustness strategies.



(b) Conditioning $\log_{10}(\kappa^{\text{loc}})$ applying the element extension strategy.



(c) Error $\|\boldsymbol{u}-\boldsymbol{u}^{\text{ref}}\|_2$ without robustness strategies.



(d) Error $\|\boldsymbol{u}-\boldsymbol{u}^{\text{ref}}\|_2$ applying the element extension strategy.

Figure 5.5: Smoothed square domain. Comparison on the local condition number $\log_{10}(\kappa^{\text{loc}})$ (top row) and of the element L^2-velocity error (bottom row) without (left) and with (right) element extension. The mesh is composed of 4×4 square elements.

5.2 Taylor–Couette with immersed boundary

In this section, we utilize the tools designed for addressing badly cut cells and faces, employing the element extension strategy and Legendre basis functions. The primary objective is to validate the optimality of the unfitted HDG-NEFEM method for one-fluid problem. Additionally, we implement the degree-adaptive strategy, as introduced in Section 4.5.1, to further enhance the computational approach.

Let us consider the 2D Taylor–Couette flow problem. The domain consists of two coaxial circles, centered in $\boldsymbol{x}_{\text{c}} := (0.5, 0.5)$ of radii $R_1 := 1/6$, $R_2 := 1/3$, rotating with different angular velocities $\omega_1 := 0$, $\omega_2 := 1$. The domain is immersed onto a mesh composed of $N_x \times N_y$ square elements, exactly matching $\Omega_{\text{o}} := (0,1)^2$. Moreover, we impose $\mu := 1$, $\boldsymbol{s} := \boldsymbol{0}$, and $\boldsymbol{u}_{\text{D}} = \boldsymbol{u}^{\text{ref}}$

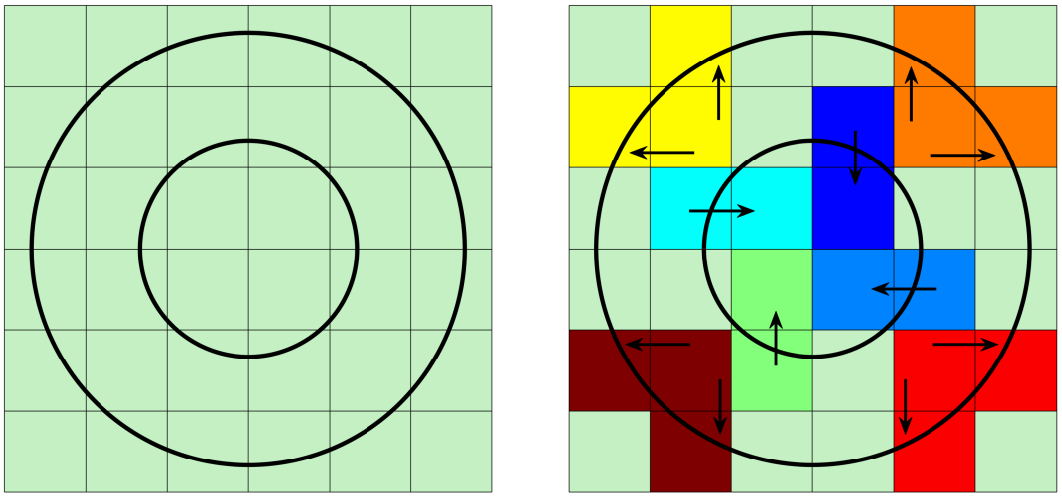
on the whole boundary $\Gamma_{\mathrm{D}} = \partial\Omega$ such that the analytical solution is

$$\boldsymbol{u}^{\mathrm{ref}} = \begin{pmatrix} -Ay - By/(x^2+y^2) \\ Ax + Bx/(x^2+y^2) \end{pmatrix}, \qquad p^{\mathrm{ref}} = 1, \tag{5.3}$$

with

$$A = \frac{\omega_2 R_2^2 - \omega_1 R_1^2}{R_2^2 - R_1^2}, \quad B = \frac{(\omega_1\omega_2)R_2^2 R_1^2}{R_2^2 - R_1^2}.$$

Figure 5.6 displays the element extension strategy using a mesh composed of 8×8 square elements. In Figures 5.7 and 5.8, we show the module of the approximated solution and its error, respectively, for $k = 4$ and using a mesh composed of 8×8 square elements.



(a) Taylor–Couette domain. (b) The element extension agglomeration procedure.

Figure 5.6: Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Example of the element extension strategy applied to a mesh composed of 8×8 square elements partitioning exactly the square $\Omega_{\mathrm{o}} = (0,1)^2$.

Figures 5.9 and 5.10 display the convergence for each variable of the HDG formulation. Note that optimal convergence h^{k+1} is achieved, and by means of the local postprocess, we obtain h^{k+2} superconvergence for $\boldsymbol{u}_\star$.

Figure 5.11 displays the maximum local and global condition numbers as a function of the polynomial approximation order k. We observe linear growth rate $\mathcal{O}(k)$. Figure 5.12 displays the global condition number κ^{glob} as a function of the mesh-size h. The expected growth rate of $\mathcal{O}(h^{-2})$ is achieved.

Finally, in Figure 5.13, we apply the degree adaptation to ensure 4 significant digits.

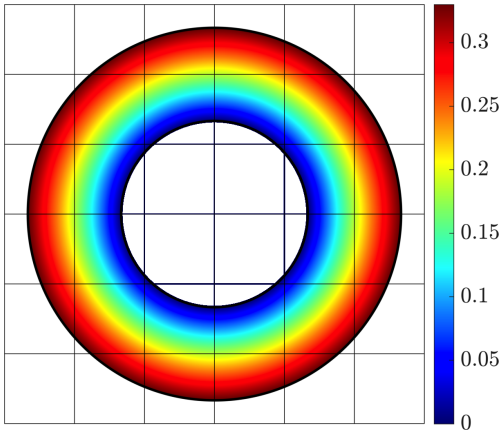
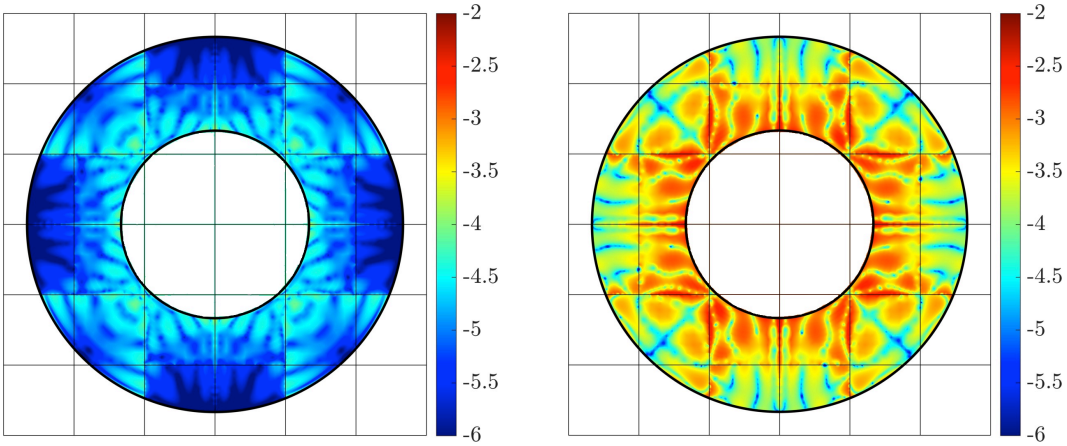


Figure 5.7: Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Absolute value of the approximated velocity for $k = 4$. The mesh is composed of 8×8 square elements. The element extension strategy is applied.



(a) Absolute value of the error of the approximated velocity, $\log_{10}(\|\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^{\mathrm{ref}}(\boldsymbol{x})\|)$.

(b) Absolute value of the error of the approximated pressure $\log_{10}(|p(\boldsymbol{x}) - p^{\mathrm{ref}}(\boldsymbol{x})|)$.

Figure 5.8: Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Absolute value of the velocity and pressure errors for $k = 4$. The mesh is composed of 8×8 square elements. The element extension strategy is applied.

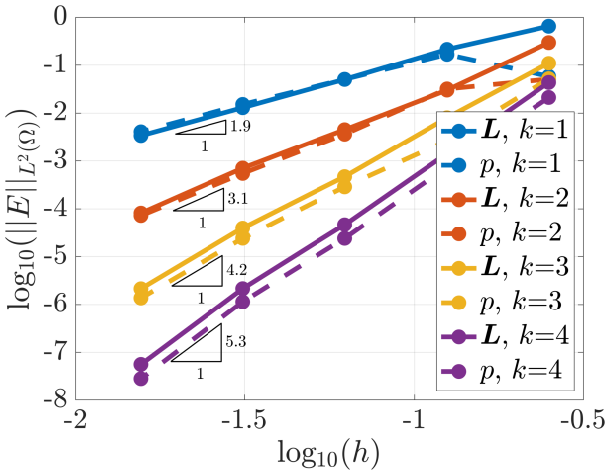
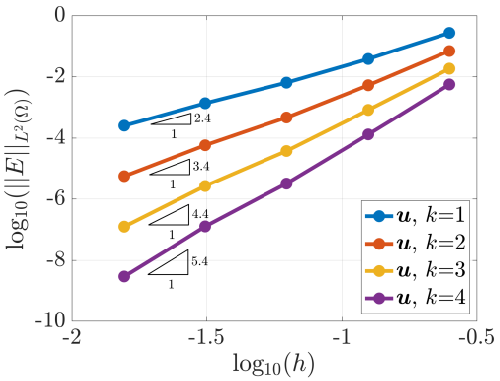
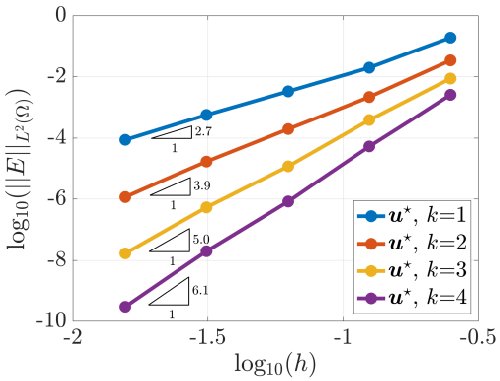


Figure 5.9: Taylor–Couette domain. The L^2-error of the mixed variable (continuous lines) and pressure (dashed lines) using element extension. The meshes are composed of 2×2, 4×4, 8×8, 16×16, and 32×32 square elements. The basis orders are $k = 1$ (blue lines), $k = 2$ (red lines), $k = 3$ (yellow lines), and $k = 4$ (purple lines).

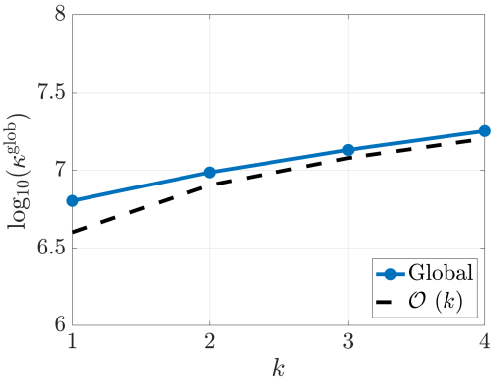


(a) $L^2(\boldsymbol{u} - \boldsymbol{u}^{\text{ref}})$ error of the velocity varying the mesh-size h.

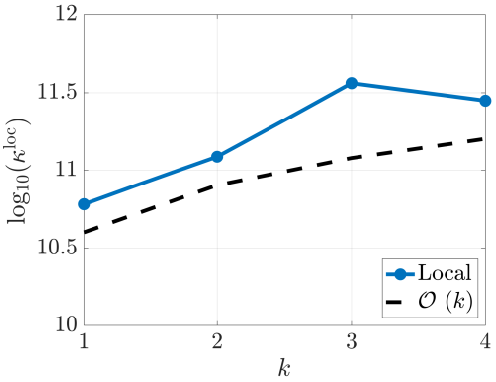


(b) $L^2(\boldsymbol{u}_\star - \boldsymbol{u}^{\text{ref}})$ error of the superconvergent velocity varying the mesh-size h.

Figure 5.10: Taylor–Couette domain. In Figure 5.10a, the L^2-velocity error, and in Figure 5.10b, L^2-superconvergent velocity error using element extension. The meshes are composed of 2×2, 4×4, 8×8, 16×16, and 32×32 square elements. The basis orders are $k = 1$ (blue lines), $k = 2$ (red lines), $k = 3$ (yellow lines), and $k = 4$ (purple lines).



(a) $\max_h(\kappa^{\mathrm{glob}})$ with respect to the polynomial order k of the HDG approximation.



(b) $\max_h(\kappa^{\mathrm{loc}})$ with respect to the polynomial order k of the HDG approximation.

Figure 5.11: Taylor–Couette domain. In Figure 5.11a, the maximum value of the global condition number as a function of the polynomial order k for the meshes composed of 2×2, 4×4, 8×8, 16×16, and 32×32 square elements. Similarly, in Figure 5.11b, the local condition number. The element extension strategy is applied.

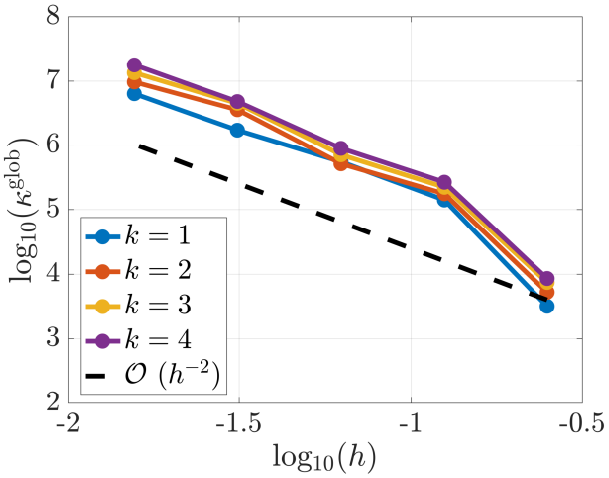
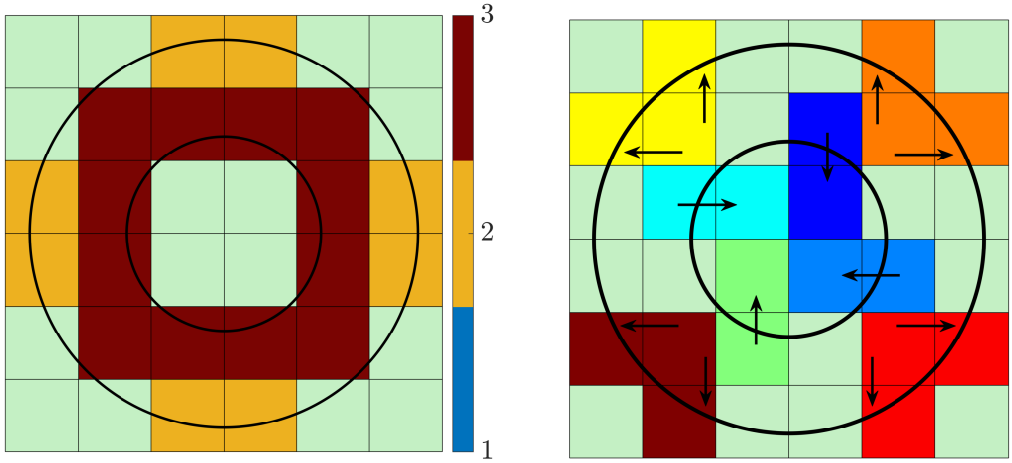


Figure 5.12: Taylor–Couette domain. Global condition number as a function of the mesh-size h for the basis orders $k = 1$ (blue lines), $k = 2$ (red lines), $k = 3$ (yellow lines), and $k = 4$ (purple lines). The element extension strategy is applied. The meshes are composed of 2×2, 4×4, 8×8, 16×16, and 32×32.



(a) Degree-adaptive approximation space.

(b) The element extension agglomeration procedure.

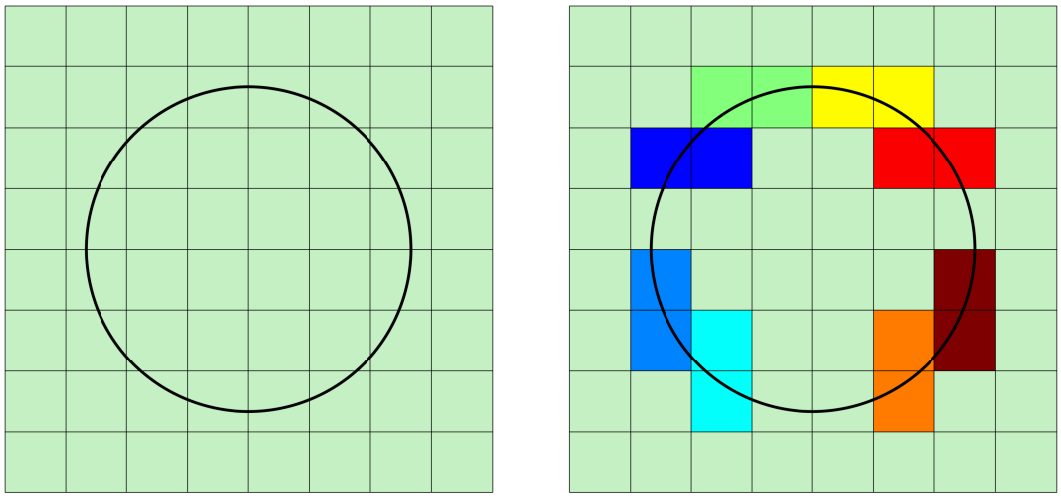
Figure 5.13: Taylor–Couette domain, zoom in $(0.125, 0.875)^2$. Degree-adaptive approximation space ensuring 4 significant digits. The mesh is composed of 8×8 square elements.

5.3 Circular bubble at equilibrium

We consider a two-fluid interface problem with a circular bubble of radius $R := 1/3$ centered in $\boldsymbol{x}_c := (0.5, 0.5)$ (domain Ω^1) having viscosity $\mu^1 := 10$ immersed into an external fluid (domain Ω^2) confined into the box $\Omega := (0,1)^2$ with viscosity $\mu^2 := 1$. Moreover, we set $\boldsymbol{s} := \boldsymbol{0}$, $\boldsymbol{u}_D := \boldsymbol{0}$, $\gamma := 1$. Under these assumptions, it is known that the problem is at equilibrium, and its analytical solution is

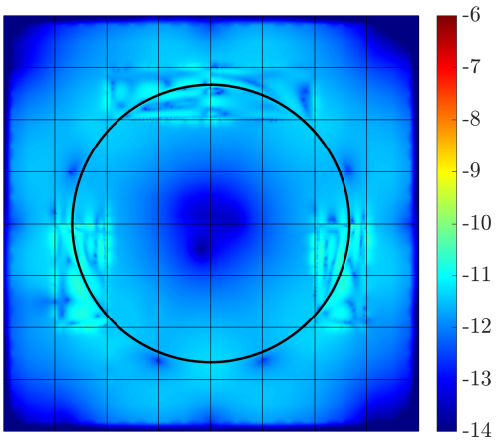
$$\boldsymbol{u}^{\text{ref}} = \boldsymbol{0},$$
$$p^{\text{ref}} = \begin{cases} \pi R\gamma - \dfrac{\gamma}{R}, & \text{if } r < R, \\ \pi R\gamma, & \text{elsewhere.} \end{cases}$$

Figure 5.14 displays the mesh domain $\Omega_o = (0,1)^2$ exactly discretized by the mesh composed of 8×8 square elements using element extension. Employing a linear HDG approximation, we achieve machine error, as shown in Figure 5.15, for all the variables: the L^2-velocity error is $\sim 10^{-12}$, the L^2-pressure error $\sim 10^{-9}$, and L^2-mixed variable error $\sim 10^{-10}$.

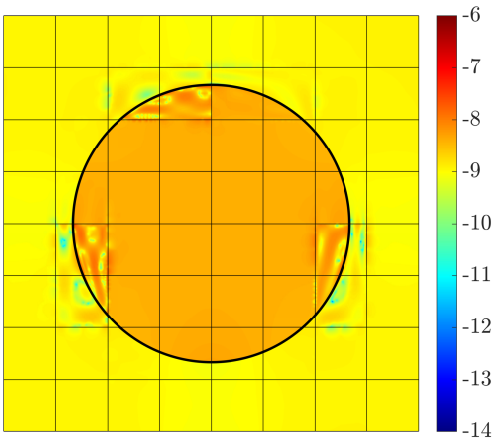


(a) Equilibrium bubble domain. (b) The element extension agglomeration procedure.

Figure 5.14: Equilibrium bubble. The mesh is composed of 8×8 square elements partitioning exactly the square $\Omega_o = (0,1)^2$. The element extension strategy is applied.



(a) Absolute value of the error of the approximated velocity, $\log_{10}(\|\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^{\text{ref}}(\boldsymbol{x})\|)$.



(b) Absolute value of the error of the approximated pressure, $\log_{10}(|p(\boldsymbol{x}) - p^{\text{ref}}(\boldsymbol{x})|)$.

Figure 5.15: Equilibrium bubble. The mesh is composed of 8×8 square elements. The basis order is $k = 1$.

5.4 Passive microfluidic mixer with immersed boundary

Let us consider the one-fluid microchannel inspired by the realistic design of a passive microfluidic mixer [126], see Figure 5.16. The flow enters from the top, where we set Dirichlet boundary conditions $\boldsymbol{u}_{\text{D}} = (0, -1)$, and exits from the bottom, where we set zero Neumann boundary conditions. No-slip boundary conditions are imposed on the remaining boundaries and on the inner circular obstacles. These inclusions are represented by 8 circles of different radii used to create additional structures to deviate the flow and summarized in Table 5.1. Moreover, we set

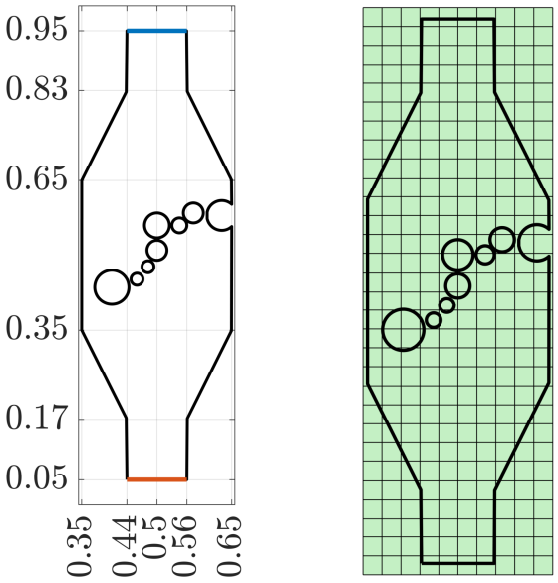
Table 5.1: Mixer: circular inclusions data (ordered from left to right).

Radii (R_i)	Centers ($\boldsymbol{x}_i^c$)	Radii (R_i)	Centers ($\boldsymbol{x}_i^c$)
$\boldsymbol{x}_c^1 := (0.410, 0.438)$	$R^1 := 0.034$	$\boldsymbol{x}_c^2 := (0.460, 0.452)$	$R^2 := 0.012$
$\boldsymbol{x}_c^3 := (0.482, 0.477)$	$R^3 := 0.012$	$\boldsymbol{x}_c^4 := (0.500, 0.510)$	$R^4 := 0.020$
$\boldsymbol{x}_c^5 := (0.500, 0.560)$	$R^5 := 0.025$	$\boldsymbol{x}_c^6 := (0.545, 0.560)$	$R^6 := 0.015$
$\boldsymbol{x}_c^7 := (0.573, 0.585)$	$R^7 := 0.020$	$\boldsymbol{x}_c^8 := (0.630, 0.580)$	$R^8 := 0.030$

$\mu := 1$, $\boldsymbol{s} := \boldsymbol{0}$. The domain is immersed onto a mesh composed of 10×32 square elements, exactly matching the rectangle $\Omega_{\text{o}} := (0.34375, 0.65625) \times (0, 1)$.

In Figure 5.17, we show the approximated solution using the degree-adaptive approximation space as depicted in Figure 5.18b, ensuring a precision of 2 significant digits. Figure 5.18a displays the employed element extension strategy. The velocity field undergoes significant modifications in regions proximate to the circular inclusions, as outlined in [126] to enhance the mixing process. Additionally, we accurately capture the pressure drop between the inlet and outlet locations. Our approach employs degree adaptation, where high-order polynomials, reaching up to order 6, are selectively used in regions near the circular inclusions owing to the more complex flow behavior. In contrast, a first-order polynomial approximation is employed

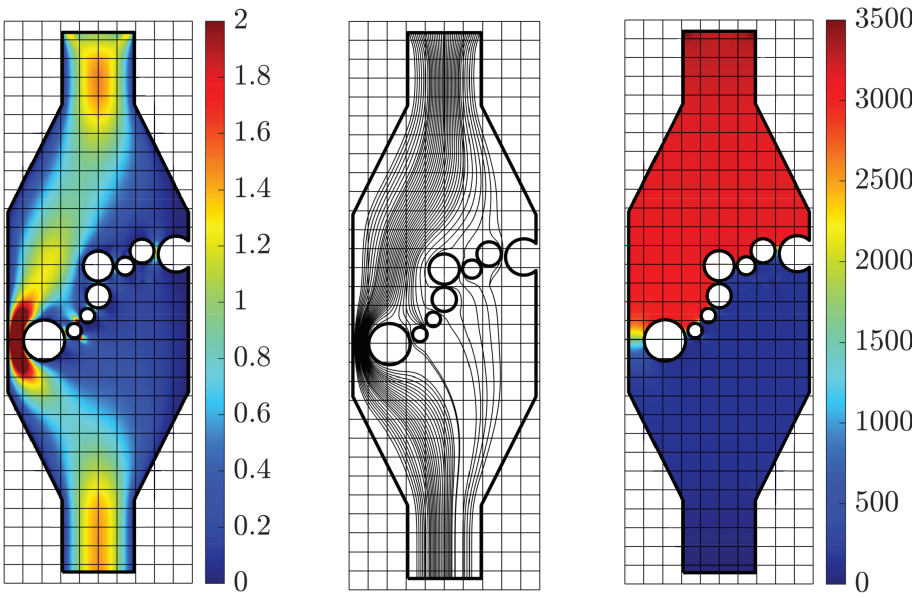
elsewhere. This strategy effectively delivers accurate and efficient solutions, even on relatively coarse meshes, as exemplified in this study.



(a) Computational domain.

(b) Unfitted mesh.

Figure 5.16: Mixer. The inlet boundary (in blue) and the outlet boundary (in red). The channel (without the inclusions) is symmetric with respect to the axis $x = 0.5$. The mesh is composed of 10×32 square elements partitioning exactly the rectangle $\Omega_o = (0.34375, 0.65625) \times (0, 1)$.

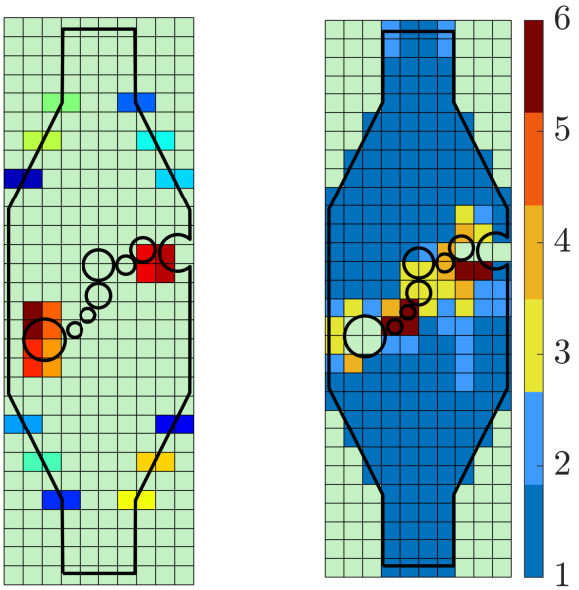


(a) Absolute value of the approximated velocity.

(b) Approximated streamlines.

(c) Approximated pressure p.

Figure 5.17: Microchannel domain. The employed degree-adaptive approximation space is displayed in Figure 5.18b and ensures 2 significant digits. The mesh is composed of 10×32 square elements. The element extension strategy is applied.



(a) The element extension agglomeration procedure.

(b) Degree-adaptive approximation space to ensure 2 significant digits.

Figure 5.18: Mixer. The mesh is composed of 10×32 square elements. Degree-adaptive approximation space ensuring 2 significant digits. The element extension strategy is applied.

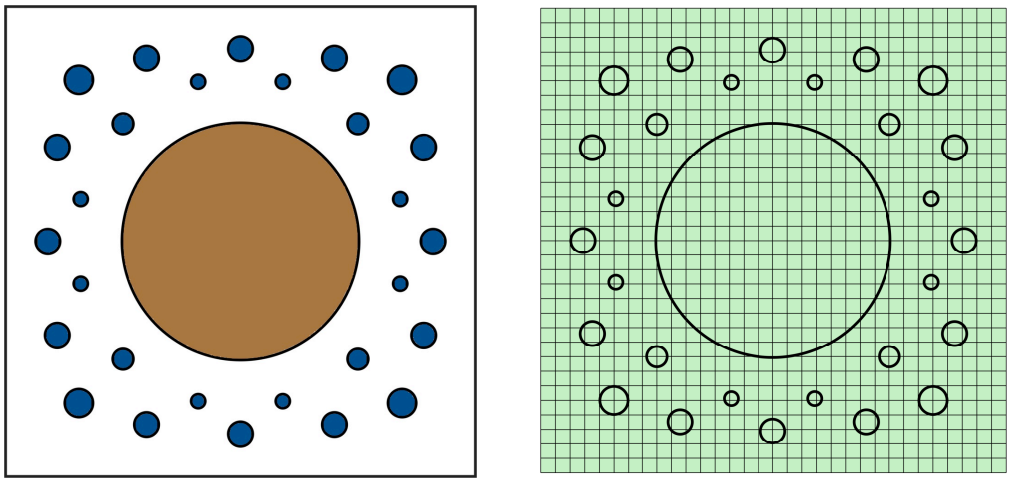
5.5 Emulsion flows in a porous medium

Consider the emulsion flow problem within porous media, as described in [7], [84], [93], where the gravitational force influences the fluids, leading to a downward flow. In this context, we have two fluids in contact: a dispersed phase (referred to as fluid 1, e.g., oil bubbles) immersed within a continuous fluid phase (fluid 2, e.g., water) that permeates through the pores of a homogeneous porous medium. Figure 5.19 illustrates a periodic microscopic unit cell of this porous medium. The solid matrix is represented by a circular region centered at $\boldsymbol{x}_{\mathrm{c}} := (0.5, 0.5)$ with a radius of $R := 0.25231$ and is assumed to be rigid and impermeable. Material 2 is subdivided into 28 bubbles, as detailed in Table 5.2, each defined by closed NURBS curves. This material exhibits a porosity of approximately 0.8, and the flow is such that the saturation of material 2 remains at 80%. Regarding the boundary conditions, no-slip boundary conditions $\boldsymbol{u}_{\mathrm{D}} := \mathbf{0}$ are set on the porous material and periodic boundary conditions on the external square box. It is worth noting that the hybrid variable $\hat{\boldsymbol{u}}$ naturally incorporates the periodic boundary conditions. Indeed, $\hat{\boldsymbol{u}}$ enables the coupling of each pair of external faces associated, which exhibit periodicity, treating them as a unified inner face while eliminating the unknowns related to one of the two original periodic faces. Additional problem data includes the surface tension of $\gamma := 2.4 \cdot 10^5$, viscosities of $\mu^1 := 40$ and $\mu^2 := 4$, such that the viscosity ratio is $\mu^1/\mu^2 = 10$, and a body force term $\boldsymbol{s} := (0, -613.125)$. These values are the adimensionalization of the physically meaningful data from [7], [84], [93].

Figure 5.21 displays the degree-adaptive approximation space ensuring 2 significant digits. Figure 5.20 illustrates the velocity and pressure approximations using such degree-adaptive approach. The approximated velocity field is in accordance with the results in [93]. Moreover, the pressure jump between fluids 1 and 2 is consistent with the surface tension condition at the interface, which depends on the curvature of the dispersed bubble phase. Indeed, we observe that the pressure jump increases proportionally to the inverse of the radius of the circular bubbles.

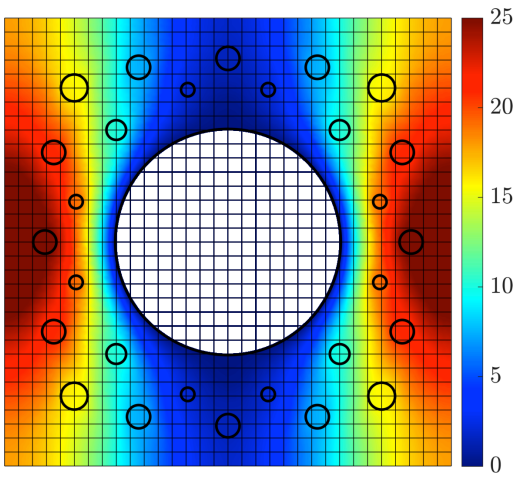
Table 5.2: Porous media emulsion: circular bubbles data.

Bubble Index (i)	Radii (R_i)	Centers ($\boldsymbol{x}_i^c$)
1-4	0.03	(0.156, 0.156), (0.156, 0.844), (0.844, 0.844), (0.844, 0.156)
5-8	0.026	(0.5, 0.1), (0.1, 0.5), (0.5, 0.9), (0.9, 0.5)
9-12	0.026	(0.7, 0.11), (0.3, 0.11), (0.11, 0.3), (0.11, 0.7)
13-16	0.026	(0.3, 0.89), (0.7, 0.89), (0.89, 0.7), (0.89, 0.3)
17-20	0.022	(0.75, 0.25), (0.25, 0.25), (0.25, 0.75), (0.75, 0.75)
21-24	0.015	(0.59, 0.16), (0.41, 0.16), (0.16, 0.41), (0.16, 0.59)
25-28	0.015	(0.41, 0.84), (0.59, 0.84), (0.84, 0.41), (0.84, 0.59)

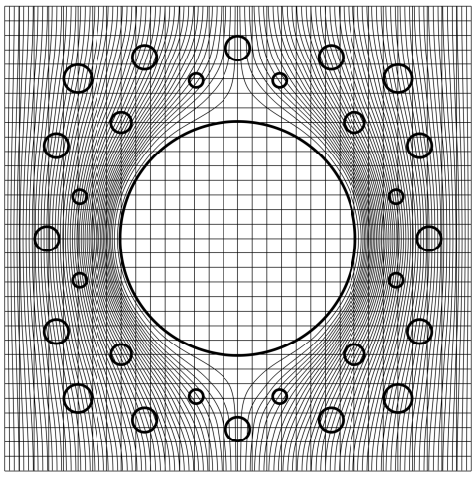


(a) Computational domain. (b) Unfitted mesh.

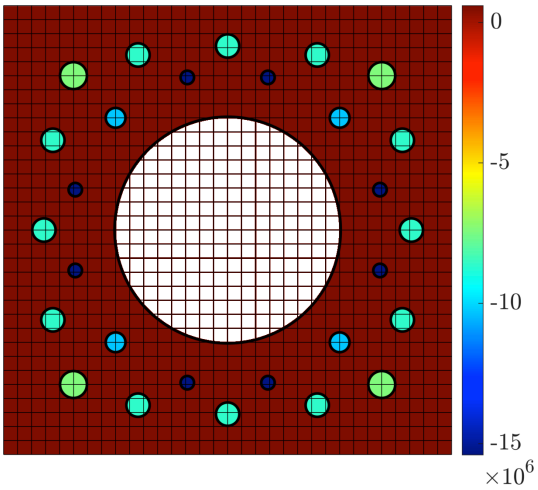
Figure 5.19: Porous media emulsion. The no-slip boundary conditions are set on the boundary of the brown circular pore, periodic boundary conditions on the external boundary of the box $(0,1)^2$, and interface boundary conditions between the continuous fluid phase (white) and the bubbles (blue). The mesh is composed of 32×32 square elements partitioning exactly the square $\Omega_o = (0,1)^2$.



(a) Absolute value of the approximated velocity.

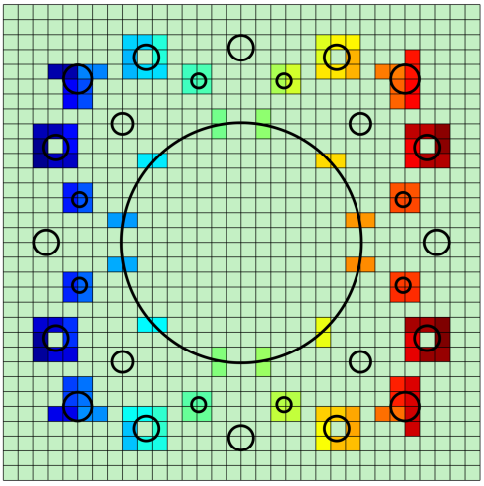


(b) Approximated streamlines.

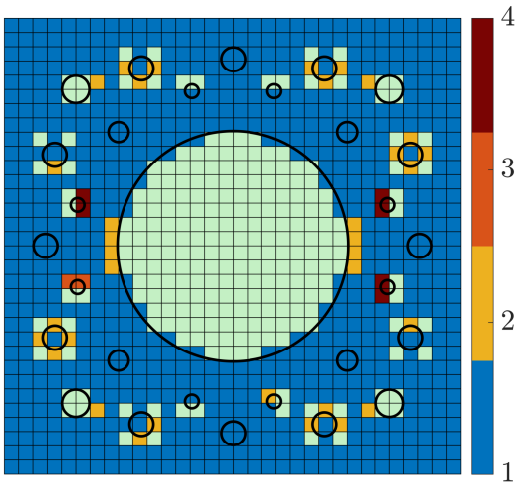


(c) Approximated pressure p.

Figure 5.20: Porous media emulsion. Figure 5.21b displays the degree-adaptive approximation space. The mesh is composed of 32×32 square elements. The element extension strategy is applied.



(a) The element extension agglomeration procedure.



(b) Degree-adaptive approximation space ensuring 2 significant digits.

Figure 5.21: Porous media emulsion. The mesh is composed of 32×32 square elements. Degree-adaptive approximation space ensuring 2 significant digits. The element extension strategy is applied.

5.6 Conclusions

In this chapter, we have extended the degree-adaptive approach, combining the advantages of the HDG-NEFEM formulation with unfitted schemes. The proposed methodology ensures an exact geometric discretization by the combination of the unfitted framework with the NURBS representation of boundaries and interfaces. Additionally, the proposed element-wise error estimator enhances the computational efficiency while ensuring the accuracy of the results. Notably, we avoided introducing additional DoFs along interfaces and boundaries.

Our investigations revealed that the unfitted HDG-NEFEM method exhibits optimal error decay and condition number growth when applied to one-fluid problems. While similar convergence results are expected for two-fluid problems, this point will be investigated in future work.

Furthermore, this Thesis opens up new areas for future research. Interesting directions are moving interfaces, unsteady flows, and more complex governing equations like the Navier-Stokes equations.

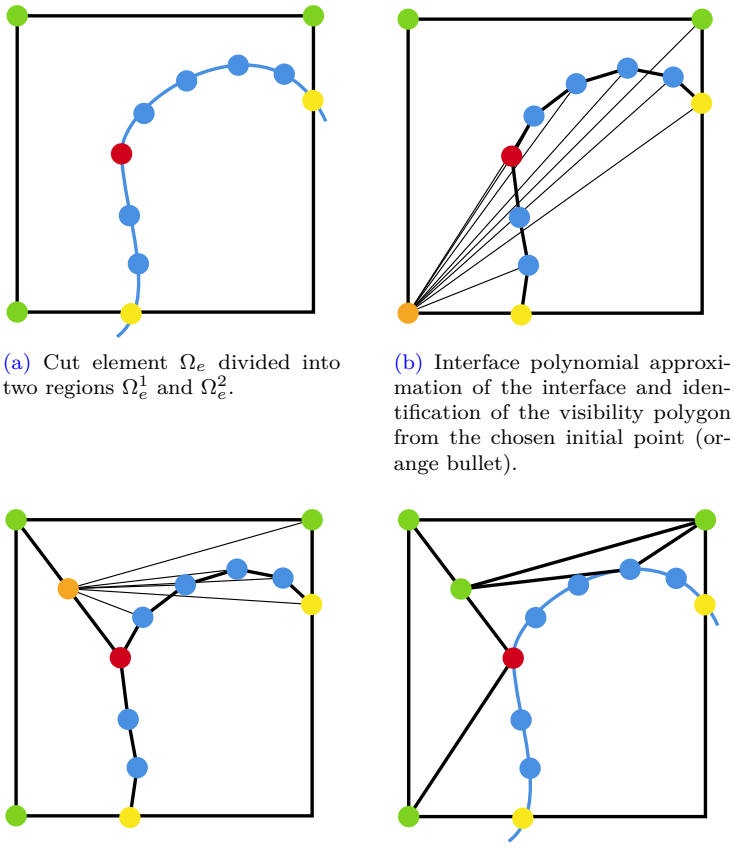
5.A Quadrature triangulation

Given a cut element Ω_e, for each region Ω_e^i, the procedure is as follows:

1. The region Ω_e^i is linearly approximated as a polygon defined by the vertices of Ω_e lying in Ω_e^i, the extremes of Υ_e^j, with $j \in \{1{:}\mathtt{n}_\Upsilon\}$, and a large enough number of points along Υ_e^j such that Υ_e^j is well described (e.g., the number of points proportional to the inverse of the curvature of Υ_e).

2. Given an initial point $\boldsymbol{x}_0$ within Ω_e^i (e.g., a vertex or the barycenter of a face belonging to $\partial\Omega_e^i$), Lee's visibility algorithm [86] defines the subregions of Ω_e^i visible from $\boldsymbol{x}_0$, i.e., the polygon composed of the points such that the segment connecting to $\boldsymbol{x}_0$ does not cut Υ_e.

3. The NURBS description of Υ_e is restored, and each subregion is divided into triangles with at maximum one curved side described by at most one NURBS curve.



(a) Cut element Ω_e divided into two regions Ω_e^1 and Ω_e^2.

(b) Interface polynomial approximation of the interface and identification of the visibility polygon from the chosen initial point (orange bullet).

(c) First visible region. Identification of other visibility polygons.

(d) Resulting triangulation of a region Ω_e^i.

Figure 5.22: Triangulation procedure for a region Ω_e^i, $i \in \{1, 2\}$. In Figure 5.22a, a generic cut element Ω_e is cut by the interface Υ_e. The two yellow bullets represent the intersection points $\Upsilon \cap \partial\Omega_e$. The interface Υ_e is composed of two NURBS curves, separated by the red bullet, and is sampled by 6 blue bullets. In Figure 5.22b, the interface Υ_e is approximated by segments. Using Lee's visibility algorithm, we check the visibility polygon of the orange bullet. In Figure 5.22c, the points visible from the orange bullet form the first visible subregion. We repeat the procedure until Ω_e^i is completely partitioned in visible subregions. In Figure 5.22d, the triangulation is finalized, and each subregion is subdivided into quadrature triangles. Note that the quadrature triangles use the exact description of Υ_e.

5.B Domain definition

The knowledge of a NURBS interface dividing Ω into two subdomains does not enable a straightforward determination of the subdomains Ω^i, $i \in \{1, 2\}$. We assume that the interface Υ is closed, orientable, and does not touch the boundary $\partial\Omega$ so that we denote by Ω^1 the interior subdomain and by Ω^2 the exterior subdomain. Moreover, we assume the external boundary $\partial\Omega$ aligned with the mesh interface so that to focus only on the interface Υ, see Figure 5.23. The same procedure can be generalized for unfitted $\partial\Omega$.

To determine Ω^i and the local regions Ω_e^i, for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all fluid $i \in \{1, 2\}$, we inspired from the work [99], and we proceed as follows:

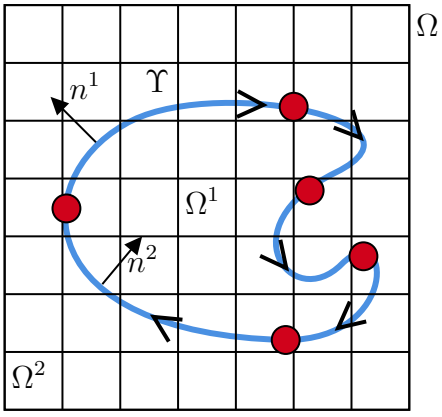


Figure 5.23: Computational domain Ω, divided into the subdomains Ω^1 and Ω^2, with fitted external boundary $\partial\Omega$. The interface Υ is clockwise oriented and is composed of 5 NURBS curves (separated by red bullets). The outward unit normal to Ω^i is denoted by $\boldsymbol{n}^i$, for all $i \in \{1, 2\}$. The mesh is composed of 7×7 square elements.

1. Construction of the points $\{\boldsymbol{x}_k^j\}_{k\in\{1:n_j\}}$ for each NURBS curve $\boldsymbol{C}_\Upsilon^j$, $j \in \{1{:}\mathtt{n}_\Upsilon\}$, composing Υ. Given n_j large enough (e.g., $n_j := 100$), the points $\boldsymbol{x}_k^j$ are set equispaced in the parametric NURBS domain $[0, 1]$, such that

$$\boldsymbol{x}_k^j = \boldsymbol{C}_\Upsilon^j(\lambda_k^j), \qquad \lambda_k^j = k/n_j, \qquad k \in \{0{:}n_j\}.$$

2. For all $k \in \{0{:}n_j\}$ and all $j \in \{1{:}\mathtt{n}_\Upsilon\}$, we find the mesh element Ω_e to which $\boldsymbol{x}_k^j$ belongs, i.e., $\boldsymbol{x}_k^j \in \Omega_e$. These elements are called cut elements. Parallelization or optimized sequential algorithms, which search for cut elements only in the neighborhood of already defined cut elements, can be used to fasten the research of such cut elements.

3. For each cut element Ω_e and all $k \in \{1{:}n_j - 1\}$, we search for neighbor points $\boldsymbol{x}_m^j$ of $\boldsymbol{x}_k^j$, with $m \in \{k - 1{:}k + 1\}$, that lie outside Ω_e. When a point $\boldsymbol{x}_m^j$ is outside Ω_e, owing to the continuity of $\boldsymbol{C}^j$, we find the intersection point between Υ and $\partial\Omega_e$, $\overline{\boldsymbol{x}}_{e,n}^j = \boldsymbol{C}_\Upsilon^j(\lambda_n^j) \in \boldsymbol{C}_\Upsilon^j([\lambda_{k-1}^j, \lambda_{k+1}^j])$ by employing a dichotomy algorithm. We denote by n_e^j the number of intersections $\Omega_e \cap \Upsilon$, $\overline{\boldsymbol{x}}_{e,n}^j$, with $n \in \{1{:}n_e^j\}$, and by $\mathcal{J}_e$ the set of coefficients j of the NURBS curves $\boldsymbol{C}_\Upsilon^j$ inside Ω_e, i.e.,

$$\mathcal{J}_e := \{j \in \{1{:}\mathtt{n}_\Upsilon\} \mid \boldsymbol{C}_\Upsilon^j \cap \Omega_e \neq \emptyset\}.$$

   Notice that we assumed interface elements to be divided at maximum into two regions, i.e., $n_e^j := 2$. This hypothesis is relaxed in Remark 5.1 for elements divided into multiple regions.

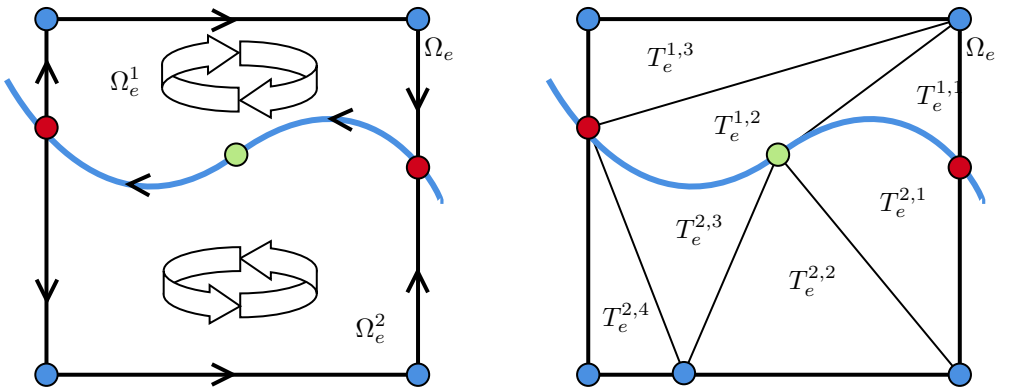
4. Check on the cut elements. Given a cut element Ω_i, for all the intersection points in Ω_i, $\overline{\boldsymbol{x}}_{i,n}^j$, with $j \in \{1{:}\mathtt{n}_\Upsilon\}$ and $n \in \{1, 2\}$, it has to exist a corresponding intersection point $\overline{\boldsymbol{x}}_{k,n}^j$, lying in a neighbor element Ω_k sharing one face, such that $\overline{\boldsymbol{x}}_{i,n}^j = \overline{\boldsymbol{x}}_{k,n}^j$. If this condition is not satisfied, a local refinement is performed to look for missing cut elements. We assume that the interface does not intersect an element through a vertex.

5. For each cut element Ω_e, the intersection points $\overline{\boldsymbol{x}}^j_{e,n}$, with $n \in \{1,2\}$, are ordered according to the orientation of Υ (first in, last out). We denote by Υ_e the portion of Υ inside Ω_e, $\Upsilon_e = \Upsilon \cap \Omega_e$, such that Υ_e is a continuous curve defined as

$$\Upsilon_e := \bigcup_{j \in \mathcal{J}_e} \Upsilon^j_e := \boldsymbol{C}^j_\Upsilon([\lambda^j_1, \lambda^j_2]), \tag{5.4}$$

with λ^j_1 and λ^j_2 the extremes of $\boldsymbol{C}^j_\Upsilon$ in Ω_e such that $\Upsilon^j_e = \boldsymbol{C}^j_\Upsilon([\lambda^j_1, \lambda^j_2])$.

6. For each cut element Ω_e, the two regions, Ω^1_e and Ω^2_e, are constructed connecting the mesh vertices and the ordered intersection points, as schematized by Figure 5.24a. The algorithm starts from the first intersection point and follows two possible paths: (i) always taking the first edge to the right (first region), or (ii) always taking the first edge to the left (second region). A superscript i denotes each region according to its fluid type, Ω^i_e, that is defined according to its position with respect to Υ (left or right side) and the orientation of Υ (counter-clockwise or clockwise oriented). When Υ is clockwise oriented, the right regions belong to Ω^1, the left to Ω^2, and vice versa when Υ is counter-clockwise oriented.



(a) Construction of regions Ω^i_e, for all $i \in \{1,2\}$.

(b) Quadrature triangulation of regions Ω^i_e, for all $i \in \{1,2\}$.

Figure 5.24: Example of a square element Ω_e cut by the interface Υ into two regions. The interface is composed of two NURBS curves, Υ^j_e and Υ^k_e, distinguished by the green bullet. Left figure: Decomposition of Ω_e in the regions Ω^1_e and Ω^2_e. Right figure: Decomposition of Ω^i_e in quadrature triangles of regions Ω^i_e, for all $i \in \{1,2\}$. The red bullets represent the intersection points $\boldsymbol{x}^j_{e,1}$ and $\boldsymbol{x}^k_{e,2}$, the blue bullets represent the vertices of the element and of the quadrature triangles lying on the mesh skeleton.

Remark 5.1 (Interface elements cut in more than two regions). Let us investigate the generalization to the case with a generic number of intersection points n^j_e splitting Ω_e into $n^j_e/2 + 1$ disjoint regions. Then, Υ_e is composed of $n^j_e - 1$ disjoint curves $\Upsilon_{e,n}$, i.e.,

$$\Upsilon_e := \bigcup_{n \in \{1:n^j_e - 1\}} \Upsilon_{e,n},$$

so that each curve $\Upsilon_{e,n}$ is continuous and can be composed of different NURBS curves $\boldsymbol{C}^j_\Upsilon$, i.e.,

$$\Upsilon_{e,n} = \sum_{j \in \mathcal{J}_{e,n}} \Upsilon^j_{e,n} := \Upsilon_{e,n} \cap \boldsymbol{C}^j_\Upsilon(\lambda), \qquad \forall \lambda \in [0,1].$$

with

$$\mathcal{J}_{e,n} := \{j \in \{1{:}\mathtt{n}_\Upsilon\} \mid \boldsymbol{C}^j_\Upsilon \cap \Upsilon_{e,n} \neq \emptyset\}.$$

Then, the different regions are identified by repeating step 6 for all the oriented edges. The generalization of the quadrature is straightforward.

Part III

Bridging HHO and HDG

Chapter 6

Bridging HHO and HDG for one- and two-fluid Stokes problems

In this chapter, we compare the HHO and HDG formulations to discretize the one-fluid and the two-fluid Stokes problems, using the so-called full gradient formulation. We consider meshes fitted to the external boundary and unfitted to the fluid interface. At the interface, surface tension effects are considered.

The chapter is structured as follows: In Section 6.1, we consider the one-fluid Stokes problem and we prove that the HHO and HDG methods can differ only in the choice of the approximation spaces and of the stabilization operator. In Section 6.2, we consider the two-fluid Stokes problem and we compare the unfitted HHO formulation, presented in Chapter 2, with the unfitted HDG formulation, presented in Chapter 4. We show that the unfitted formulations can additionally differ in the choice of the penalty method.

6.1 One-fluid Stokes problem

In this section, we compare the HHO and HDG formulations to discretize the one-fluid Stokes problem posed in a domain with a simple shape so that a fitted mesh can be used. We prove that the HHO and HDG methods can differ only in the choice of the approximation spaces and of the stabilization operator. For the reader's convenience and the self-completeness of this chapter, some definitions and statements from the previous chapters are re-iterated in this chapter.

6.1.1 Model problem

Let $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$, $\mathtt{n_{sd}} \in \{2,3\}$, be the computational domain (connected, bounded, open Lipschitz set) occupied by a single incompressible Stokes fluid. On the boundary $\partial\Omega$ of Ω, Dirichlet and Neumann boundary conditions are prescribed, on the (relatively open) subsets Γ_{D} and Γ_{N}, respectively, with $\overline{\Gamma_{\mathrm{D}}} \cup \overline{\Gamma_{\mathrm{N}}} = \partial\Omega$ and $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$. We impose the velocity $\boldsymbol{u}_{\mathrm{D}} \in \mathcal{H}^{1/2}(\Gamma_{\mathrm{D}}; \mathbb{R}^{\mathtt{n_{sd}}})$ on the Dirichlet boundary, and the pseudo-traction $\boldsymbol{t} \in \mathcal{L}^2(\Gamma_{\mathrm{N}}; \mathbb{R}^{\mathtt{n_{sd}}})$ on the Neumann boundary. Additionally, we impose the body force $\boldsymbol{f} \in \mathcal{L}^2(\Omega; \mathbb{R}^{\mathtt{n_{sd}}})$ in the domain. For simplicity, we assume

that Ω is a polyhedron so that it can be meshed exactly.

We consider the so-called *full gradient formulation* of the Stokes problem such that we seek the velocity and pressure fields $(\boldsymbol{u}, p) \in \mathcal{H}^1(\Omega;\mathbb{R}^{\mathtt{n_{sd}}}) \times \mathcal{L}^2(\Omega)$ that satisfy

$$-\nabla\cdot(\mu\nabla\boldsymbol{u} - p\boldsymbol{I}_{\mathtt{n_{sd}}}) = \boldsymbol{f} \qquad \text{in } \Omega, \tag{6.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{6.1b}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}}, \tag{6.1c}$$

$$(\mu\nabla\boldsymbol{u} - p\boldsymbol{I}_{\mathtt{n_{sd}}})\boldsymbol{n}_\Omega = \boldsymbol{t} \qquad \text{on } \Gamma_{\mathrm{N}}, \tag{6.1d}$$

where μ is the viscosity, $\boldsymbol{n}_\Omega$ is the unit outward normal to Ω, and $\boldsymbol{I}_{\mathtt{n_{sd}}}$ is the identity tensor of dimension $\mathtt{n_{sd}}$. We assume that μ is piecewise constant on a polyhedral partition of Ω. We assume that the interface between Γ_{D} and Γ_{N} is Lipschitz, so that there is a bounded extension operator $\mathcal{H}^{1/2}(\Gamma_{\mathrm{D}};\mathbb{R}^{\mathtt{n_{sd}}}) \to \mathcal{H}^{1/2}(\partial\Omega;\mathbb{R}^{\mathtt{n_{sd}}})$. Moreover, we assume that the equations are written in non-dimensional form. When $\Gamma_{\mathrm{N}} = \emptyset$, the Stokes problem is solvable only if

$$\int_{\partial\Omega} \boldsymbol{u}_{\mathrm{D}}\cdot\boldsymbol{n}_\Omega = 0 \tag{6.2}$$

and, in this case, up to a global additive constant on the pressure, which we fix by imposing

$$\int_\Omega p = 0. \tag{6.3}$$

Moreover, when $\Gamma_{\mathrm{D}} = \emptyset$, we assume the compatibility condition

$$\int_\Omega \boldsymbol{f} + \int_{\partial\Omega} \boldsymbol{t} = \boldsymbol{0}. \tag{6.4}$$

6.1.2 Discrete setting

We consider a shape-regular mesh composed of $\mathtt{n_{el}}$ disjoint (open) subdomains Ω_e such that Ω is exactly covered, i.e.,

$$\overline{\Omega} = \bigcup_{e\in\{1:\mathtt{n_{el}}\}} \overline{\Omega_e}. \tag{6.5}$$

Here and in what follows, we employ the shorthand notation $\{m{:}n\} := \{m,\dots,n\} = \{p \in \mathbb{N},\ m \le p \le n\}$ for integer numbers $m \le n$. We denote by h_e the diameter of a generic mesh cell Ω_e and by $\boldsymbol{n}_e$ its unit outward normal. We assume that the mesh is fitted to the partition on which μ is piecewise constant and to the partition of $\partial\Omega$ related to the prescribed boundary conditions.

The collection of mesh faces, Γ, is called *mesh skeleton* and is defined as

$$\Gamma := \bigcup_{e\in\{1:\mathtt{n_{el}}\}} \partial\Omega_e = \bigcup_{f\in\{1:\mathtt{n_{fc}}\}} \Gamma_f, \tag{6.6}$$

where $\partial\Omega_e$ is the boundary of Ω_e for all $e \in \{1{:}\mathtt{n_{el}}\}$, Γ_f is a generic mesh face belonging to Γ, and $\mathtt{n_{fc}}$ is the total number of mesh faces. The set Γ is partitioned as $\Gamma = \Gamma^\circ \cup \Gamma^\partial$, where $\Gamma^\circ := \Gamma \setminus \partial\Omega$ is the collection of mesh interfaces and Γ^∂ is the collection of mesh boundary

faces. The faces in Γ° compose the internal skeleton. These faces are conventionally numbered from 1 to $\mathtt{n}_{\mathtt{fc}}^\circ$.

To avoid distracting technicalities, HHO and HDG methods are first compared in the case of homogeneous Dirichlet boundary conditions, i.e., we set $\Gamma_{\mathrm{N}} = \emptyset$, $\Gamma_{\mathrm{D}} = \partial\Omega$ and $\boldsymbol{u}_{\mathrm{D}} = \boldsymbol{0}$. The treatment of more general boundary conditions is done in Section 6.1.6. To solve problem (6.1), HHO methods utilize a primal formulation and approximate the triple $(\boldsymbol{u}, p, \hat{\boldsymbol{u}})$, formed by velocity, pressure, and velocity trace on the mesh skeleton. In contrast, HDG methods approximate a quadruple by introducing an extra dual variable, $\boldsymbol{L}$, inspired from the mixed formulation and which represents the gradient of the velocity scaled by $-\sqrt{\mu}$. Notice that the assumption of piecewise constant viscosity is fundamental to preserve the polynomial nature of the HDG dual variable $\boldsymbol{L} = -\sqrt{\mu}\,\boldsymbol{\nabla}\boldsymbol{u}$. A simple alternative is to define $\boldsymbol{L}$ as $-\boldsymbol{\nabla}\boldsymbol{u}$.

Henceforth, to distinguish between the HHO and HDG notation, we employ the superscript HHO to denote HHO-related quantities and the superscript HDG to denote HDG-related quantities. Both HHO and HDG methods consider the following approximation spaces:

$$\boldsymbol{\mathcal{L}} := \{\boldsymbol{G} := (\boldsymbol{G}_e)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}});\ \boldsymbol{G}_e \in \boldsymbol{\mathcal{L}}_e,\ \forall e \in \{1:\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.7a}$$

$$\boldsymbol{\mathcal{U}} := \{\boldsymbol{v} := (\boldsymbol{v}_e)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}});\ \boldsymbol{v}_e \in \boldsymbol{\mathcal{U}}_e,\ \forall e \in \{1:\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.7b}$$

$$\mathcal{P} := \{q := (q_e)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega;\mathbb{R});\ q_e \in \mathcal{P}_e,\ \forall e \in \{1:\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.7c}$$

$$\hat{\boldsymbol{\mathcal{U}}} := \{\hat{\boldsymbol{v}} := (\hat{\boldsymbol{v}}_f)_{f\in\{1:\mathtt{n}_{\mathtt{fc}}\}} \in \mathcal{L}^2(\Gamma;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}});\ \hat{\boldsymbol{v}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f,\ \forall f \in \{1:\mathtt{n}_{\mathtt{fc}}\}\}, \tag{6.7d}$$

where $\boldsymbol{\mathcal{L}}_e, \boldsymbol{\mathcal{U}}_e, \mathcal{P}_e, \hat{\boldsymbol{\mathcal{U}}}_f$ are finite-dimensional spaces composed of polynomials from Ω_e to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}}$, from Ω_e to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}$, from Ω_e to $\mathbb{R}$, and from Γ_f to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}$, respectively. Letting $k \geq 0$ be the polynomial degree, the prototypical example we consider henceforth is

$$\boldsymbol{\mathcal{L}}_e := \mathbb{P}^k(\Omega_e;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}}), \qquad \mathcal{P}_e := \mathbb{P}^k(\Omega_e;\mathbb{R}), \qquad \hat{\boldsymbol{\mathcal{U}}}_f := \mathbb{P}^k(\Gamma_f;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), \tag{6.8}$$

and

$$\boldsymbol{\mathcal{U}}_e := \begin{cases} \mathbb{P}^k(\Omega_e;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), & \text{equal-order setting,} \\ \mathbb{P}^{k+1}(\Omega_e;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), & \text{mixed-order setting.} \end{cases} \tag{6.9}$$

For all $e \in \{1:\mathtt{n}_{\mathtt{el}}\}$, we denote by $\hat{\boldsymbol{v}}_e$ the collection of $(\hat{\boldsymbol{v}}_f)_{\Gamma_f\subset\partial\Omega_e}$, so that

$$\hat{\boldsymbol{v}}_e := (\hat{\boldsymbol{v}}_f)_{\Gamma_f\subset\partial\Omega_e} \in \hat{\boldsymbol{\mathcal{U}}}_e := \bigtimes_{\Gamma_f\subset\partial\Omega_e} \hat{\boldsymbol{\mathcal{U}}}_f. \tag{6.10}$$

In both HHO and HDG methods, the difference of the trace of $\boldsymbol{v}_e$ on $\partial\Omega_e$ and $\hat{\boldsymbol{v}}_e$ plays an important role. We write

$$\hat{\boldsymbol{\mathcal{U}}}_e^* := \{\boldsymbol{v}_e|_{\partial\Omega_e} - \hat{\boldsymbol{v}}_e;\ \forall \boldsymbol{v}_e \in \boldsymbol{\mathcal{U}}_e,\ \forall \hat{\boldsymbol{v}}_e \in \hat{\boldsymbol{\mathcal{U}}}_e\}. \tag{6.11}$$

Recalling (6.9), there are two situations: (i) the equal-order case where the polynomials composing $\boldsymbol{\mathcal{U}}_e$ and $\hat{\boldsymbol{\mathcal{U}}}_f$ are of the same order, in which case we have $\hat{\boldsymbol{\mathcal{U}}}_e^* = \hat{\boldsymbol{\mathcal{U}}}_e$; (ii) the mixed-order case where the polynomials composing $\boldsymbol{\mathcal{U}}_e$ are one order higher than those composing $\hat{\boldsymbol{\mathcal{U}}}_e$, in which case $\hat{\boldsymbol{\mathcal{U}}}_e \subsetneq \hat{\boldsymbol{\mathcal{U}}}_e^*$. Notice that, in the mixed-order case, a projection of $\boldsymbol{v}_e$ onto the space of

polynomials of degree k is needed before computing the trace. Then, it is convenient to define the $\mathcal{L}^2$-orthogonal projections $\Pi_e : \mathcal{L}^2(\Omega_e;\mathbb{R}^{\mathtt{n_{sd}}}) \to \boldsymbol{\mathcal{U}}_e$ and $\hat{\Pi}_e : \mathcal{L}^2(\partial\Omega_e;\mathbb{R}^{\mathtt{n_{sd}}}) \to \hat{\boldsymbol{\mathcal{U}}}_e$.

Different strategies can be used to impose the homogeneous version of the Dirichlet boundary condition (6.1c). For simplicity, we resort here to the strong imposition by employing the space

$$\hat{\boldsymbol{\mathcal{U}}}_0 := \{\hat{\boldsymbol{v}} \in \hat{\boldsymbol{\mathcal{U}}};\ \hat{\boldsymbol{v}} = \boldsymbol{0} \text{ on } \partial\Omega\}. \tag{6.12}$$

Other possibilities, common in the context of dG-FEM, are the weak imposition of the Dirichlet boundary conditions by means of Nitsche's method [29] or the use of Lagrange multipliers.

We denote by $(\cdot,\cdot)_S$ the $\mathcal{L}^2$-inner product in a generic $\mathtt{n_{sd}}$-dimensional subset $S \subset \Omega$ (e.g., mesh cell) defined as

$$(u,v)_S := \int_S uv, \qquad (\boldsymbol{u},\boldsymbol{v})_S := \int_S \boldsymbol{u}\cdot\boldsymbol{v}, \qquad (\boldsymbol{G},\boldsymbol{L})_S := \int_S \boldsymbol{G}:\boldsymbol{L}, \tag{6.13}$$

for scalar-, vector-, and tensor-valued functions, respectively. Analogously, we denote by $\langle\cdot,\cdot\rangle_S$ the corresponding $\mathcal{L}^2$-product in a generic $(\mathtt{n_{sd}}-1)$-dimensional subset $S \subset \Omega$ (e.g., mesh face).

We define two jump operators across a smooth $(\mathtt{n_{sd}}-1)$-dimensional manifold $\Sigma \subset \Omega$ (typically, a mesh face or a physical interface). The manifold Σ separates two subsets Ω_l and Ω_r of Ω, with unit outward normal $\boldsymbol{n}_l$ and $\boldsymbol{n}_r$, respectively. In the HHO notation, Σ is oriented by the unit normal $\boldsymbol{n}_\Sigma$ which conventionally points from Ω_l to Ω_r. Thus,

$$\boldsymbol{n}_\Sigma = \boldsymbol{n}_l = -\boldsymbol{n}_r. \tag{6.14}$$

Then, for a generic piecewise smooth function χ that can be scalar-, vector- or tensor-valued, we set

$$[\![\chi]\!]_\Sigma^{\mathrm{HHO}} := (\chi|_{\Omega_l})|_\Sigma - (\chi|_{\Omega_r})|_\Sigma. \tag{6.15}$$

We notice that Ω_l and Ω_r do not play symmetric roles, but the advantage of the definition (6.15) is that $[\![\chi]\!]_\Sigma^{\mathrm{HHO}}$ is of the same nature as χ. In the HDG notation, Ω_l and Ω_r play symmetric roles, but the jump operator acts on quantities that depend on the outward normal. Specifically, we set

$$[\![q\boldsymbol{n}]\!]_\Sigma^{\mathrm{HDG}} = q_l\boldsymbol{n}_l + q_r\boldsymbol{n}_r = (q_l - q_r)\boldsymbol{n}_l = [\![q]\!]_\Sigma^{\mathrm{HHO}}\boldsymbol{n}_\Sigma, \tag{6.16a}$$

$$[\![\boldsymbol{v}\cdot\boldsymbol{n}]\!]_\Sigma^{\mathrm{HDG}} = \boldsymbol{v}_l\cdot\boldsymbol{n}_l + \boldsymbol{v}_r\cdot\boldsymbol{n}_r = (\boldsymbol{v}_l - \boldsymbol{v}_r)\cdot\boldsymbol{n}_l = [\![\boldsymbol{v}]\!]_\Sigma^{\mathrm{HHO}}\cdot\boldsymbol{n}_\Sigma, \tag{6.16b}$$

$$[\![\boldsymbol{v}\otimes\boldsymbol{n}]\!]_\Sigma^{\mathrm{HDG}} = \boldsymbol{v}_l\otimes\boldsymbol{n}_l + \boldsymbol{v}_r\otimes\boldsymbol{n}_r = (\boldsymbol{v}_l - \boldsymbol{v}_r)\otimes\boldsymbol{n}_l = [\![\boldsymbol{v}]\!]_\Sigma^{\mathrm{HHO}}\otimes\boldsymbol{n}_\Sigma, \tag{6.16c}$$

$$[\![\boldsymbol{G}\boldsymbol{n}]\!]_\Sigma^{\mathrm{HDG}} = \boldsymbol{G}_l\boldsymbol{n}_l + \boldsymbol{G}_r\boldsymbol{n}_r = (\boldsymbol{G}_l - \boldsymbol{G}_r)\boldsymbol{n}_l = [\![\boldsymbol{G}]\!]_\Sigma^{\mathrm{HHO}}\boldsymbol{n}_\Sigma, \tag{6.16d}$$

with q, $\boldsymbol{v}$, and $\boldsymbol{G}$ being a scalar-, vector-, and tensor-valued piecewise smooth functions defined in Ω_l and Ω_r, respectively.

Remark 6.1 (Pressure mean value). Owing to the prescribed Dirichlet boundary conditions on the entire boundary $\partial\Omega$, the discrete pressure is actually sought in the global space

$$\mathcal{P}^0 := \{q \in \mathcal{P};\ \int_\Omega q = 0\}.$$

The corresponding local spaces denoted by $\mathcal{P}_e^0$, for all $e \in \{1{:}\mathtt{n_{el}}\}$. One can also work with

discrete pressures in $\mathcal{P}$ and enforce the condition $\int_\Omega p = 0$ in the resulting global space by means of a Lagrange multiplier.

6.1.3 HDG formulation

In this section, we present the HDG formulation of the Stokes problem (6.1) considering only homogeneous Dirichlet boundary conditions, i.e., $\Gamma_\mathrm{N} = \emptyset$, $\Gamma_\mathrm{D} = \partial\Omega$, and $\boldsymbol{u}_\mathrm{D} = \boldsymbol{0}$.

6.1.3.1 Global formulation

It is convenient to define the HDG viscous stress tensor such that

$$\boldsymbol{\Phi}_e^\mathrm{HDG} := \sqrt{\mu_e}\boldsymbol{L}_e^\mathrm{HDG} + p_e^\mathrm{HDG}\boldsymbol{I}_{\mathtt{n_{sd}}}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}. \tag{6.17}$$

To discretize problem (6.1), HDG methods seek the quadruple $(\boldsymbol{L}^\mathrm{HDG}, \boldsymbol{u}^\mathrm{HDG}, p^\mathrm{HDG}, \hat{\boldsymbol{u}}^\mathrm{HDG}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ such that the following holds true:

$$-(\boldsymbol{G}_e, \boldsymbol{L}_e^\mathrm{HDG})_{\Omega_e} + (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\,\boldsymbol{G}_e), \boldsymbol{u}_e^\mathrm{HDG})_{\Omega_e} - \langle \sqrt{\mu_e}\,\boldsymbol{G}_e\,\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^\mathrm{HDG}\rangle_{\partial\Omega_e} = 0, \tag{6.18a}$$

$$-(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{\Phi}_e^\mathrm{HDG})_{\Omega_e} + \langle \boldsymbol{w}_e, \widehat{\boldsymbol{\Phi}}_e^\mathrm{HDG}\boldsymbol{n}_e\rangle_{\partial\Omega_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e}, \tag{6.18b}$$

$$(\boldsymbol{\nabla}q_e, \boldsymbol{u}_e^\mathrm{HDG})_{\Omega_e} - \langle q_e, \hat{\boldsymbol{u}}_e^\mathrm{HDG}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} = 0, \tag{6.18c}$$

$$\langle \hat{\boldsymbol{w}}_f, [\![\widehat{\boldsymbol{\Phi}}^\mathrm{HDG}\boldsymbol{n}]\!]_{\Gamma_f}^\mathrm{HDG}\rangle_{\Gamma_f} = 0, \tag{6.18d}$$

for all $(\boldsymbol{G}_e, \boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{L}}_e \times \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_e^0$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, and for all $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$ and all $f \in \{1{:}\mathtt{n_{fc}^\circ}\}$. The numerical trace $\widehat{\boldsymbol{\Phi}}^\mathrm{HDG}$ is defined such that, for all $e \in \{1{:}\mathtt{n_{el}}\}$,

$$\widehat{\boldsymbol{\Phi}}_e^\mathrm{HDG}\boldsymbol{n}_e := \boldsymbol{\Phi}_e^\mathrm{HDG}\boldsymbol{n}_e + \boldsymbol{s}_e^\mathrm{HDG}(\boldsymbol{u}_e^\mathrm{HDG}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^\mathrm{HDG}), \tag{6.19}$$

where $\boldsymbol{s}_e^\mathrm{HDG} : \hat{\boldsymbol{\mathcal{U}}}_e^* \to \hat{\boldsymbol{\mathcal{U}}}_e$ is a linear stabilization operator which is generally taken to be of the form

$$\boldsymbol{s}_e^\mathrm{HDG}(\hat{\boldsymbol{\delta}}) := \tau_e^\mathrm{HDG}\begin{cases}\hat{\boldsymbol{\delta}}, & \text{equal-order setting,}\\ \hat{\Pi}_e(\hat{\boldsymbol{\delta}}), & \text{mixed-order setting,}\end{cases} \tag{6.20}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^*$, with a positive weight $\tau_e^\mathrm{HDG} > 0$. Notice that (6.18b) can be rewritten as

$$(\boldsymbol{w}_e, \boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_e^\mathrm{HDG})_{\Omega_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^\mathrm{HDG}(\boldsymbol{u}_e^\mathrm{HDG}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^\mathrm{HDG})\right\rangle_{\partial\Omega_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e}. \tag{6.21}$$

In problem (6.18), the equation (6.18a) is the discrete counterpart of the relation $\boldsymbol{L}^\mathrm{HDG} = -\sqrt{\mu}\,\boldsymbol{\nabla}\boldsymbol{u}^\mathrm{HDG}$, the equation (6.18b) that of the momentum balance (6.1a), and the equation (6.18c) that of the divergence condition (6.1b). Moreover, the equation (6.18d) is the transmission condition which weakly enforces the continuity of the normal component of the numerical flux across the faces composing the internal skeleton. Notice that, by definition, $\hat{\boldsymbol{u}}^\mathrm{HDG}$ is uniquely defined on each face of Γ° and vanishes on each face of Γ^∂.

Proposition 1 (Algebraic formulation). The algebraic formulation of problem (6.18) is

$$\left[\begin{array}{c|c|c|c} A_{LL}^{\mathrm{HDG}} & A_{Lu}^{\mathrm{HDG}} & 0 & A_{L\hat{u}}^{\mathrm{HDG}} \\ \hline A_{uL}^{\mathrm{HDG}} & A_{uu}^{\mathrm{HDG}} & A_{up}^{\mathrm{HDG}} & A_{u\hat{u}}^{\mathrm{HDG}} \\ \hline 0 & A_{pu}^{\mathrm{HDG}} & 0 & A_{p\hat{u}}^{\mathrm{HDG}} \\ \hline A_{\hat{u}L}^{\mathrm{HDG}} & A_{\hat{u}u}^{\mathrm{HDG}} & A_{\hat{u}p}^{\mathrm{HDG}} & A_{\hat{u}\hat{u}}^{\mathrm{HDG}} \end{array}\right] \left[\begin{array}{c} \boldsymbol{L}^{\mathrm{HDG}} \\ \hline \boldsymbol{u}^{\mathrm{HDG}} \\ \hline p^{\mathrm{HDG}} \\ \hline \hat{\boldsymbol{u}}^{\mathrm{HDG}} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline F_u \\ \hline 0 \\ \hline 0 \end{array}\right], \tag{6.22}$$

and the matrix is symmetric, i.e., $A_{uL}^{\mathrm{HDG}} = (A_{Lu}^{\mathrm{HDG}})^{\mathrm{T}}$, $A_{up}^{\mathrm{HDG}} = (A_{pu}^{\mathrm{HDG}})^{\mathrm{T}}$, $A_{\hat{u}L}^{\mathrm{HDG}} = (A_{L\hat{u}}^{\mathrm{HDG}})^{\mathrm{T}}$, $A_{\hat{u}u}^{\mathrm{HDG}} = (A_{u\hat{u}}^{\mathrm{HDG}})^{\mathrm{T}}$, and $A_{\hat{u}p}^{\mathrm{HDG}} = (A_{p\hat{u}}^{\mathrm{HDG}})^{\mathrm{T}}$.

6.1.3.2 Static condensation

Static condensation is a procedure that allows one to eliminate locally (in every mesh cell) the discrete unknowns $\boldsymbol{L}_e$ and $\boldsymbol{u}_e$, as well as p_e up to one constant. Thus, the only globally coupled unknowns are the skeleton unknowns $\hat{\boldsymbol{u}}_f$ and one constant pressure per mesh cell. To handle the pressure decomposition, we write

$$\mathcal{P}^0 = \mathcal{P}_0^0 \oplus \mathcal{P}_\perp^0, \tag{6.23}$$

where

$$\mathcal{P}_0^0 := \{q \in \mathcal{P}^0;\ q|_{\Omega_e} \in \mathbb{P}^0(\Omega_e;\mathbb{R}),\ \forall e \in \{1{:}\mathtt{n_{el}}\}\}, \tag{6.24a}$$
$$\mathcal{P}_\perp^0 := \{q \in \mathcal{P}^0;\ (q,1)_{\Omega_e} = 0,\ \forall e \in \{1{:}\mathtt{n_{el}}\}\}, \tag{6.24b}$$

and we write $\mathcal{P}_{0,e}^0$ and $\mathcal{P}_{\perp,e}^0$ for the corresponding local spaces. Consistently, for all $q \in \mathcal{P}^0$, we write $q = q_0 + q_\perp$ and $q_e = q_{0,e} + q_{\perp,e}$, for all $e \in \{1{:}\mathtt{n_{el}}\}$.

Proposition 2 (Local and global problems). The quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.18) if and only if:

(i) The local variables $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p_\perp^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}_\perp^0$ solve the following local problems as a function of the global variables $(\hat{\boldsymbol{u}}^{\mathrm{HDG}}, p_0^{\mathrm{HDG}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$:

$$-(\boldsymbol{G}_e, \boldsymbol{L}_e^{\mathrm{HDG}})_{\Omega_e} + (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\,\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} = \langle \sqrt{\mu_e}\,\boldsymbol{G}_e\,\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\rangle_{\partial\Omega_e}, \tag{6.25a}$$

$$(\boldsymbol{w}_e, \boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_{\perp,e}^{\mathrm{HDG}})_{\Omega_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^{\mathrm{HDG}}(\boldsymbol{u}_e^{\mathrm{HDG}}|_{\partial\Omega_e})\right\rangle_{\partial\Omega_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^{\mathrm{HDG}}(\hat{\boldsymbol{u}}_e^{\mathrm{HDG}})\right\rangle_{\partial\Omega_e}, \tag{6.25b}$$

$$(\boldsymbol{\nabla} q_e, \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} = \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e}, \tag{6.25c}$$

for all $(\boldsymbol{G}_e, \boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{L}}_e \times \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_{\perp,e}^0$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, where

$$\boldsymbol{\Phi}_{\perp,e}^{\mathrm{HDG}} := \sqrt{\mu_e}\boldsymbol{L}_e^{\mathrm{HDG}} + p_{\perp,e}^{\mathrm{HDG}}\boldsymbol{I}_{\mathtt{n_{sd}}} = \boldsymbol{\Phi}_e^{\mathrm{HDG}} - p_{0,e}^{\mathrm{HDG}}\boldsymbol{I}_{\mathtt{n_{sd}}}; \tag{6.26}$$

(ii) The global variables $(\hat{\boldsymbol{u}}^{\mathrm{HDG}}, p_0^{\mathrm{HDG}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$ solve the following global problems:

$$\langle \hat{\boldsymbol{w}}_f, [\![p_0^{\mathrm{HDG}}\boldsymbol{n} - \boldsymbol{s}^{\mathrm{HDG}}(\hat{\boldsymbol{u}}^{\mathrm{HDG}})]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f} = -\langle \hat{\boldsymbol{w}}_f, [\![\boldsymbol{\Phi}_\perp^{\mathrm{HDG}}\boldsymbol{n} + \boldsymbol{s}^{\mathrm{HDG}}(\boldsymbol{u}^{\mathrm{HDG}})]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f}, \tag{6.27a}$$

$$\langle 1, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} = 0, \tag{6.27b}$$

where (6.27a) holds for all $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$ and all $f \in \{1{:}\mathtt{n}_{\mathtt{fc}}^{\circ}\}$, (6.27b) holds for all $e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}$, and the local variables $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p_{\perp}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}_{\perp}^{0}$ result from (6.25).

6.1.4 HHO formulation

In this section, we present the HHO formulation of the Stokes problem (6.1) considering only homogeneous Dirichlet boundary conditions, i.e., $\Gamma_{\mathrm{N}} = \emptyset$, $\Gamma_{\mathrm{D}} = \partial\Omega$, and $\boldsymbol{u}_{\mathrm{D}} = \boldsymbol{0}$. For a generic pair $(\boldsymbol{v}, \hat{\boldsymbol{v}}) \in \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_0$, we introduce the HHO notation $\underline{\boldsymbol{v}} := (\boldsymbol{v}, \hat{\boldsymbol{v}}) \in \underline{\boldsymbol{\mathcal{U}}} := \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_0$. Correspondingly, for the local space, we define $\underline{\boldsymbol{\mathcal{U}}}_e := \boldsymbol{\mathcal{U}}_e \times \hat{\boldsymbol{\mathcal{U}}}_e$, for all $e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}$.

6.1.4.1 Global formulation

To discretize problem (6.1), HHO methods seek the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ such that the following holds true:

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{u}}^{\mathrm{HHO}}) - b^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, p^{\mathrm{HHO}}) = F(\underline{\boldsymbol{w}}), \tag{6.28a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{u}}^{\mathrm{HHO}}, q) = 0, \tag{6.28b}$$

for all $(\boldsymbol{w}, q, \hat{\boldsymbol{w}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$. The global problems are assembled cellwise as

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{v}}) := \sum_{e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}} a_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e, \underline{\boldsymbol{v}}_e), \tag{6.29a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{v}}, q) := \sum_{e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}} b_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e, q_e), \tag{6.29b}$$

$$F(\underline{\boldsymbol{w}}) := \sum_{e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}} (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e}, \tag{6.29c}$$

where the local bilinear forms are

$$\begin{aligned} a_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e, \underline{\boldsymbol{v}}_e) := {} & \mu_e \Big(\boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e), \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e) \Big)_{\Omega_e} \\ & + \tau_e^{\mathrm{HHO}} \Big\langle \boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{w}_e|_{\partial\Omega_e} - \hat{\boldsymbol{w}}_e), \boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{v}_e|_{\partial\Omega_e} - \hat{\boldsymbol{v}}_e) \Big\rangle_{\partial\Omega_e}, \end{aligned} \tag{6.30a}$$

$$b_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e, q_e) := (D_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e), q_e)_{\Omega_e}, \tag{6.30b}$$

for all $e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}$, with a positive weight $\tau_e^{\mathrm{HHO}} > 0$ (typically, $\tau_e^{\mathrm{HHO}} := \mu_e h_e^{-1}$). The following operators have been used: The gradient reconstruction operator $\boldsymbol{L}_e^{\mathrm{HHO}} : \underline{\boldsymbol{\mathcal{U}}}_e \to \boldsymbol{\mathcal{L}}_e$ such that

$$(\boldsymbol{G}, \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e))_{\Omega_e} := (\boldsymbol{G}, \boldsymbol{\nabla}\boldsymbol{v}_e)_{\Omega_e} + \langle \boldsymbol{G}\boldsymbol{n}_e, \hat{\boldsymbol{v}}_e - \boldsymbol{v}_e|_{\partial\Omega_e} \rangle_{\partial\Omega_e}, \tag{6.31}$$

for all $\underline{\boldsymbol{v}}_e \in \underline{\boldsymbol{\mathcal{U}}}_e$, and all $\boldsymbol{G} \in \boldsymbol{\mathcal{L}}_e$, the divergence reconstruction operator such that

$$D_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e) := \boldsymbol{I}_{\mathtt{n}_{\mathtt{sd}}} : \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e), \tag{6.32}$$

and the stabilization operator $\boldsymbol{s}_e^{\mathrm{HHO}} : \hat{\boldsymbol{\mathcal{U}}}_e^* \to \hat{\boldsymbol{\mathcal{U}}}_e$ such that

$$\boldsymbol{s}_e^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}}) := \begin{cases} \hat{\Pi}_e \Big(\hat{\boldsymbol{\delta}} + \big((\boldsymbol{I}_{\mathtt{n}_{\mathtt{sd}}} - \Pi_e) \boldsymbol{R}_e^{\mathrm{HHO}}(\boldsymbol{0}, \hat{\boldsymbol{\delta}}) \big)|_{\partial\Omega_e} \Big), & \text{equal-order setting,} \\ \hat{\Pi}_e(\hat{\boldsymbol{\delta}}), & \text{mixed-order setting,} \end{cases} \tag{6.33}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^*$. In the equal-order setting, the stabilization operator involves the local reconstruction operator $\boldsymbol{R}_e^{\mathrm{HHO}} : \underline{\boldsymbol{\mathcal{U}}}_e \to \boldsymbol{\mathcal{U}}_e^+ := \mathbb{P}^{k+1}(\Omega_e; \mathbb{R}^{\mathtt{n_{sd}}})$ such that

$$(\boldsymbol{\nabla w}, \boldsymbol{\nabla R}_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e))_{\Omega_e} := (\boldsymbol{\nabla w}, \boldsymbol{\nabla v}_e)_{\Omega_e} + \langle \boldsymbol{\nabla w n}_e, \hat{\boldsymbol{v}}_e - \boldsymbol{v}_e|_{\partial\Omega_e} \rangle_{\partial\Omega_e}, \tag{6.34}$$

for all $\underline{\boldsymbol{v}}_e \in \underline{\boldsymbol{\mathcal{U}}}_e$ and all $\boldsymbol{w} \in \boldsymbol{\mathcal{U}}_e^+$. Finally, it is convenient to define the HHO viscous stress tensor such that

$$\boldsymbol{\Phi}_e^{\mathrm{HHO}} := -\mu_e \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) + p_e^{\mathrm{HHO}} \boldsymbol{I}_{\mathtt{n_{sd}}}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}. \tag{6.35}$$

Proposition 3 (Algebraic formulation). The algebraic formulation of problem (6.28) is

$$\left[\begin{array}{cc:c} A_{uu}^{\mathrm{HHO}} & A_{up}^{\mathrm{HHO}} & A_{u\hat{u}}^{\mathrm{HHO}} \\ A_{pu}^{\mathrm{HHO}} & 0 & A_{p\hat{u}}^{\mathrm{HHO}} \\ \hdashline A_{\hat{u}u}^{\mathrm{HHO}} & A_{\hat{u}p}^{\mathrm{HHO}} & A_{\hat{u}\hat{u}}^{\mathrm{HHO}} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}^{\mathrm{HHO}} \\ p^{\mathrm{HHO}} \\ \hdashline \hat{\boldsymbol{u}}^{\mathrm{HHO}} \end{array}\right] = \left[\begin{array}{c} F_u \\ 0 \\ \hdashline 0 \end{array}\right], \tag{6.36}$$

and the matrix is symmetric, i.e., $A_{up}^{\mathrm{HHO}} = (A_{pu}^{\mathrm{HHO}})^{\mathrm{T}}$, $A_{\hat{u}u}^{\mathrm{HHO}} = (A_{u\hat{u}}^{\mathrm{HHO}})^{\mathrm{T}}$, and $A_{\hat{u}p}^{\mathrm{HHO}} = (A_{p\hat{u}}^{\mathrm{HHO}})^{\mathrm{T}}$. The vector F_u on the right-hand side of (6.36) is the same as in (6.22).

6.1.4.2 Static condensation

Recall the decomposition (6.23) of the pressure space, i.e., $\mathcal{P}^0 = \mathcal{P}_0^0 \oplus \mathcal{P}_\perp^0$. Notice that $b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_{0,e}^{\mathrm{HHO}}) = (\boldsymbol{L}^{\mathrm{HHO}}(\boldsymbol{w}_e, \boldsymbol{0}), p_{0,e}^{\mathrm{HHO}} \boldsymbol{I}_{\mathtt{n_{sd}}})_{\Omega_e} = 0$ for all $\boldsymbol{w}_e \in \boldsymbol{\mathcal{U}}_e$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, as a consequence of (6.31) after integration by parts.

Proposition 4 (Local and global problems). The triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.28) if and only if:

(i) The local variables $(\boldsymbol{u}^{\mathrm{HHO}}, p_\perp^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}_\perp^0$ solve the following local problems as a function of the global variables $(\hat{\boldsymbol{u}}^{\mathrm{HHO}}, p_0^{\mathrm{HHO}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$:

$$a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), (\boldsymbol{u}_e^{\mathrm{HHO}}, \boldsymbol{0})) - b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_{\perp,e}^{\mathrm{HHO}}) = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} - a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), (\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})), \tag{6.37a}$$

$$b_e^{\mathrm{HHO}}((\boldsymbol{u}_e^{\mathrm{HHO}}, \boldsymbol{0}), q_e) = -b_e^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}), q_e), \tag{6.37b}$$

for all $(\boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_{\perp,e}^0$ and all $e \in \{1{:}\mathtt{n_{el}}\}$;

(ii) The global variables $(\hat{\boldsymbol{u}}^{\mathrm{HHO}}, p_0^{\mathrm{HHO}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$ solve the following global problems:

$$\begin{aligned} &a^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), (\boldsymbol{0}, \hat{\boldsymbol{u}}^{\mathrm{HHO}})) - b^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), p_0^{\mathrm{HHO}}) \\ &\qquad = -a^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), (\boldsymbol{u}^{\mathrm{HHO}}, \boldsymbol{0})) + b^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), p_\perp^{\mathrm{HHO}}), \end{aligned} \tag{6.38a}$$

$$b_e^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}), 1) = 0, \tag{6.38b}$$

where (6.38a) holds for all $\hat{\boldsymbol{w}} \in \hat{\boldsymbol{\mathcal{U}}}_0$, (6.38b) holds for all $e \in \{1{:}\mathtt{n_{el}}\}$, and the local variables $(\boldsymbol{u}^{\mathrm{HHO}}, p_\perp^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}_\perp^0$ result from (6.37).

6.1.5 Bridging HHO and HDG

Within the above setting, HDG and HHO methods depend on the choice of the local spaces $\boldsymbol{\mathcal{L}}_e, \boldsymbol{\mathcal{U}}_e, \mathcal{P}_e, \hat{\boldsymbol{\mathcal{U}}}_f$, and of the stabilization operators $\boldsymbol{s}_e^{\mathrm{HDG}}$ and $\boldsymbol{s}_e^{\mathrm{HHO}}$, respectively. Recall that in the equal-order case for HHO, we have $\hat{\boldsymbol{\mathcal{U}}}_e^* = \hat{\boldsymbol{\mathcal{U}}}_e$, so that $\boldsymbol{s}_e^{\mathrm{HHO}} : \hat{\boldsymbol{\mathcal{U}}}_e \to \hat{\boldsymbol{\mathcal{U}}}_e$. We then define the adjoint operator with respect to the $\mathcal{L}^2(\partial\Omega_e;\mathbb{R}^{\mathtt{n_{sd}}})$-inner product by setting

$$\left\langle (\boldsymbol{s}_e^{\mathrm{HHO}})^*(\hat{\boldsymbol{\delta}}^*), \hat{\boldsymbol{\delta}} \right\rangle_{\partial\Omega_e} := \left\langle \hat{\boldsymbol{\delta}}^*, \boldsymbol{s}_e^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}}) \right\rangle_{\partial\Omega_e}, \qquad \forall \hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}^* \in \hat{\boldsymbol{\mathcal{U}}}_e. \tag{6.39}$$

Proposition 5 (From HHO to HDG). Define the local polynomial spaces

$$\boldsymbol{\mathcal{L}}_e = \mathbb{P}^k(\Omega_e;\mathbb{R}^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}), \quad \boldsymbol{\mathcal{U}}_e = \mathbb{P}^\ell(\Omega_e;\mathbb{R}^{\mathtt{n_{sd}};\mathbb{R}}), \quad \mathcal{P}_e = \mathbb{P}^k(\Omega_e), \quad \hat{\boldsymbol{\mathcal{U}}}_f = \mathbb{P}^k(\Gamma_f;\mathbb{R}^{\mathtt{n_{sd}}}), \tag{6.40}$$

with either $\ell = k$ (equal-order) or $\ell = k+1$ (mixed-order). If the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.28) with the stabilization operator (6.33), then setting

$$\boldsymbol{Z}_e^{\mathrm{HHO}} := -\sqrt{\mu_e}\boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}), \tag{6.41}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$, the quadruple $(\boldsymbol{Z}^{\mathrm{HHO}}, \boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.18) with the stabilization operator

$$\boldsymbol{s}_e^{\mathrm{HDG}}(\hat{\boldsymbol{\delta}}) = \tau_e^{\mathrm{HHO}} \begin{cases} \big((\boldsymbol{s}_e^{\mathrm{HHO}})^* \circ \boldsymbol{s}_e^{\mathrm{HHO}}\big)(\hat{\boldsymbol{\delta}}), & \text{equal-order setting,} \\ \hat{\Pi}_e(\hat{\boldsymbol{\delta}}), & \text{mixed-order setting,} \end{cases} \tag{6.42}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^*$, and with the numerical flux trace such that

$$\widehat{\boldsymbol{\Phi}}_e^{\mathrm{HDG}}\boldsymbol{n}_e := \boldsymbol{\Phi}_e^{\mathrm{HHO}}\boldsymbol{n}_e + \boldsymbol{s}_e^{\mathrm{HDG}}(\boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}), \tag{6.43}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$, where $\boldsymbol{\Phi}_e^{\mathrm{HHO}} := \sqrt{\mu_e}\boldsymbol{Z}_e^{\mathrm{HHO}} + p_e^{\mathrm{HHO}}\boldsymbol{I}_{\mathtt{n_{sd}}}$.

Proposition 6 (From HDG to HHO). Define the local polynomial spaces as in (6.40). If the quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.18) with the stabilization operator (6.20), then

$$\boldsymbol{L}_e^{\mathrm{HDG}} = -\sqrt{\mu_e}\,\boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}), \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}, \tag{6.44}$$

with the operator $\boldsymbol{L}_e^{\mathrm{HHO}}$ defined in (6.31), and the triple $(\boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.28) with the weight $\tau_e^{\mathrm{HHO}} = \tau_e^{\mathrm{HDG}}$ and stabilization operator

$$\boldsymbol{s}_e^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}}) = \begin{cases} \hat{\boldsymbol{\delta}}, & \text{equal-order setting,} \\ \hat{\Pi}_e(\hat{\boldsymbol{\delta}}), & \text{mixed-order setting,} \end{cases} \tag{6.45}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^*$.

Proof. We only prove Proposition 5 since the proof of Proposition 6 is similar. Moreover, we only detail the equal-order setting. Assume that the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.28). Let $\boldsymbol{Z}_e^{\mathrm{HHO}}$ be defined by (6.41) for all $e \in \{1{:}\mathtt{n_{el}}\}$. We need

to prove that the quadruple $(\boldsymbol{Z}^{\mathrm{HHO}}, \boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ satisfies (6.18). Notice that $\boldsymbol{Z}^{\mathrm{HHO}} \in \boldsymbol{\mathcal{L}}$ since μ is piecewise constant.

(i) Since μ_e is constant, we infer that for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $\boldsymbol{G}_e \in \boldsymbol{\mathcal{L}}_e$,

$$\begin{aligned}(\boldsymbol{G}_e, \boldsymbol{Z}_e^{\mathrm{HHO}})_{\Omega_e} &= -\sqrt{\mu_e}\Big(\boldsymbol{G}_e, \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big)_{\Omega_e}\\ &= -\sqrt{\mu_e}\Big\{(\boldsymbol{G}_e, \boldsymbol{\nabla}\boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} + \langle \boldsymbol{G}_e\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}} - \boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e}\rangle_{\partial\Omega_e}\Big\}\\ &= (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} - \langle \sqrt{\mu_e}\boldsymbol{G}_e\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\rangle_{\partial\Omega_e}.\end{aligned}$$

Hence, (6.18a) holds.

(ii) We notice that

$$\begin{aligned}\boldsymbol{\Phi}_e^{\mathrm{HHO}} &= \sqrt{\mu_e}\boldsymbol{Z}_e^{\mathrm{HHO}} + p_e^{\mathrm{HHO}}\boldsymbol{I}_{\mathtt{n_{sd}}}\\ &= -\mu_e\boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) + p_e^{\mathrm{HHO}}\boldsymbol{I}_{\mathtt{n_{sd}}}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}.\end{aligned}$$

Hence, for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $\boldsymbol{w}_e \in \boldsymbol{\mathcal{U}}_e$,

$$\begin{aligned}-(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HHO}})_{\Omega_e} &= \mu_e\Big(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big)_{\Omega_e} - (\boldsymbol{\nabla}\boldsymbol{w}_e, p_e^{\mathrm{HHO}}\boldsymbol{I}_{\mathtt{n_{sd}}})_{\Omega_e}\\ &= -\langle \boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HHO}}\boldsymbol{n}_e\rangle_{\partial\Omega_e} + \Big\langle \tau_e^{\mathrm{HHO}}\boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{w}_e|_{\partial\Omega_e}), \boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big\rangle_{\partial\Omega_e}\\ &\quad + a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), \underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) - b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_e^{\mathrm{HHO}})\\ &= -\Big\langle \boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HHO}}\boldsymbol{n}_e + \tau_e^{\mathrm{HHO}}\Big((\boldsymbol{s}_e^{\mathrm{HHO}})^* \circ \boldsymbol{s}_e^{\mathrm{HHO}}\Big)(\boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big\rangle_{\partial\Omega_e}\\ &\quad + (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e}\\ &= -\langle \boldsymbol{w}_e, \widehat{\boldsymbol{\Phi}}_e^{\mathrm{HHO}}\boldsymbol{n}_e\rangle_{\partial\Omega_e} + (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e},\end{aligned}$$

where we used (6.37a) tested with $(\boldsymbol{w}_e, \boldsymbol{0})$. Hence, (6.18b) holds.

(iii) For all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $q_e \in \mathcal{P}_e^0$, we have

$$\begin{aligned}(\boldsymbol{\nabla}q_e, \boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} &= -b_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}, q_e) + \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e}\\ &= \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e},\end{aligned}$$

where we used (6.37b) tested q_e. Hence, (6.18c) holds.

(iv) Let $f \in \{1{:}\mathtt{n_{fc}^\circ}\}$ and $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$. We test (6.28a) with $\underline{\boldsymbol{w}} := (\boldsymbol{0}, \tilde{\boldsymbol{w}}_f)$ where $\tilde{\boldsymbol{w}}_f := (\delta_{ff'}\hat{\boldsymbol{w}}_f)_{f' \in \{1:\mathtt{n_{fc}^\circ}\}} \in \hat{\boldsymbol{\mathcal{U}}}_0$, and $\delta_{ff'}$ is the Kronecker symbol. Thus, the only nonzero component of $\tilde{\boldsymbol{w}}_f$ is attached to the face Γ_f. We obtain

$$\begin{aligned}0 &= a^{\mathrm{HHO}}((\boldsymbol{0}, \tilde{\boldsymbol{w}}_f), \underline{\boldsymbol{u}}^{\mathrm{HHO}}) - b^{\mathrm{HHO}}((\boldsymbol{0}, \tilde{\boldsymbol{w}}_f), p^{\mathrm{HHO}})\\ &= \sum_{\substack{e \in \{1:\mathtt{n_{el}}\}\\ \Gamma_f \subset \partial\Omega_e}} \Big\{a_e^{\mathrm{HHO}}((\boldsymbol{0}, \tilde{\boldsymbol{w}}_e), \underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) - b_e^{\mathrm{HHO}}((\boldsymbol{0}, \tilde{\boldsymbol{w}}_e), p_e^{\mathrm{HHO}})\Big\}.\end{aligned}$$

We have

$$\begin{aligned}\alpha_e &:= a_e^{\mathrm{HHO}}((\boldsymbol{0}, \tilde{\boldsymbol{w}}_e), \underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\\ &= \mu_e\langle \hat{\boldsymbol{w}}_f, \boldsymbol{L}_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\boldsymbol{n}_e\rangle_{\Gamma_f} - \Big\langle \hat{\boldsymbol{w}}_f, \Big((\boldsymbol{s}_e^{\mathrm{HHO}})^* \circ \boldsymbol{s}_e^{\mathrm{HHO}}\Big)(\boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big\rangle_{\Gamma_f},\end{aligned}$$

and

$$\begin{aligned}\beta_e &:= b_e^{\mathrm{HHO}}((\boldsymbol{0},\tilde{\boldsymbol{w}}_e),p_e^{\mathrm{HHO}})\\ &= \langle\hat{\boldsymbol{w}}_f,p_e^{\mathrm{HHO}}\boldsymbol{n}_e\rangle_{\Gamma_f}.\end{aligned}$$

Hence, recalling the definition (6.43) of the numerical flux trace, we obtain

$$\alpha_e-\beta_e=-\widehat{\boldsymbol{\Phi}}_e^{\mathrm{HDG}}\boldsymbol{n}_e.$$

We infer that

$$\langle\hat{\boldsymbol{w}}_f,[\![\widehat{\boldsymbol{\Phi}}^{\mathrm{HDG}}\boldsymbol{n}]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f}=0.$$

Hence, (6.18d) holds and the proof is complete. □

Proposition 7 (Algebraic comparison). Following the choices in Proposition 5, the algebraic HHO formulation in (6.36) can be restated to align with the HDG algebraic formulation (6.22) by writing

$$\begin{cases}A_{LL}^{\mathrm{HHO}}\boldsymbol{L}^{\mathrm{HHO}}=-(A_{Lu}^{\mathrm{HHO}}\boldsymbol{u}^{\mathrm{HHO}}+A_{L\hat{u}}^{\mathrm{HHO}}\hat{\boldsymbol{u}}^{\mathrm{HHO}}),&\text{(6.46a)}\\ \left[\begin{array}{cc:c}A_{uu}^{\mathrm{HHO}}&A_{up}^{\mathrm{HHO}}&A_{u\hat{u}}^{\mathrm{HHO}}\\A_{pu}^{\mathrm{HHO}}&0&A_{p\hat{u}}^{\mathrm{HHO}}\\\hdashline A_{\hat{u}u}^{\mathrm{HHO}}&A_{\hat{u}p}^{\mathrm{HHO}}&A_{\hat{u}\hat{u}}^{\mathrm{HHO}}\end{array}\right]\left[\begin{array}{c}\boldsymbol{u}^{\mathrm{HHO}}\\p^{\mathrm{HHO}}\\\hdashline\hat{\boldsymbol{u}}^{\mathrm{HHO}}\end{array}\right]=\left[\begin{array}{c}F_u\\0\\\hdashline 0\end{array}\right],&\text{(6.46b)}\end{cases}$$

where (6.46a) is the algebraic counterpart of (6.31) (so that A_{LL}^{HHO} is block-diagonal) and (6.46b) is nothing but (6.36).

Remark 6.2 (Equal-order stabilization). The HHO stabilization operator defined in equation (6.33) relies on the reconstruction operator. This means that, for any point $\boldsymbol{x}\in\Gamma_f$, the value of $\boldsymbol{s}_e^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}})(\boldsymbol{x})$ depends on the values taken by $\hat{\boldsymbol{\delta}}$ over the whole boundary $\partial\Omega_e$. In contrast, the HDG stabilization operator acts pointwise. In other words, $\boldsymbol{s}_e^{\mathrm{HDG}}(\hat{\boldsymbol{\delta}})(\boldsymbol{x})$ depends solely on the value of $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$. This distinction leads to HHO methods providing optimal error estimates for all polynomial orders $k\geq 0$ and for general polyhedral meshes. For HDG methods using classical pointwise stabilization, optimality can be achieved on meshes with specific cell shapes (simplices, quads, hexs, etc.) for $k\geq 1$.

6.1.6 More general boundary conditions

Let us now consider the Stokes problem defined in (6.1) with Neumann and non-homogeneous Dirichlet boundary conditions, i.e., $\Gamma_{\mathrm{N}}\neq\emptyset$, where a generic pseudo-traction $\boldsymbol{t}\in\mathcal{L}^2(\Gamma_{\mathrm{N}};\mathbb{R}^{\mathtt{n_{sd}}})$ is applied, and $\Gamma_{\mathrm{D}}\neq\emptyset$, where a generic velocity $\boldsymbol{u}_{\mathrm{D}}\in\mathcal{H}^{1/2}(\Gamma_{\mathrm{D}};\mathbb{R}^{\mathtt{n_{sd}}})$ is imposed.

It is convenient to define the $\mathcal{L}^2$-orthogonal projection $\hat{\Pi}_f:\mathcal{L}^2(\Gamma_f;\mathbb{R}^{\mathtt{n_{sd}}})\to\hat{\boldsymbol{\mathcal{U}}}_f$, for all $f\in\{1{:}\mathtt{n_{fc}}\}$. With obvious notation, the mesh skeleton is decomposed as $\Gamma=\Gamma^{\circ}\cup\Gamma^{\mathrm{N}}\cup\Gamma^{\mathrm{D}}$, and we enumerate conventionally the faces in $\Gamma^{\circ}\cup\Gamma^{\mathrm{N}}$ from 1 to $\mathtt{n}_{\mathtt{fc}}^{\circ\mathrm{N}}$. Moreover, for all $\Gamma_f\subset\Gamma^N$, we define the subscript e_f as the unique index $e\in\{1{:}\mathtt{n_{el}}\}$ so that $\Gamma_f\subset\partial\Omega_e$. Additionally, we set

$$\hat{\boldsymbol{\mathcal{U}}}_{\mathrm{D}}=\{\hat{\boldsymbol{v}}\in\hat{\boldsymbol{\mathcal{U}}};\ \hat{\boldsymbol{v}}_f=\hat{\Pi}_f(\boldsymbol{u}_{\mathrm{D}}|_{\Gamma_f}),\ \forall\Gamma_f\subset\Gamma^{\mathrm{D}}\},\tag{6.47}$$

and we replace (6.12) by

$$\hat{\boldsymbol{\mathcal{U}}}_0=\{\hat{\boldsymbol{v}}\in\hat{\boldsymbol{\mathcal{U}}};\ \hat{\boldsymbol{v}}_f=\boldsymbol{0},\ \forall\Gamma_f\subset\Gamma^{\mathrm{D}}\}.\tag{6.48}$$

To discretize problem (6.1), HDG methods seek the quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P} \times \hat{\boldsymbol{\mathcal{U}}}_{\mathrm{D}}$ such that the following holds true:

$$-(\boldsymbol{G}_e, \boldsymbol{L}_e^{\mathrm{HDG}})_{\Omega_e} + (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\,\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle \sqrt{\mu_e}\,\boldsymbol{G}_e\,\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\rangle_{\partial\Omega_e} = 0, \tag{6.49a}$$

$$-(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HDG}})_{\Omega_e} + \langle \boldsymbol{w}_e, \widehat{\boldsymbol{\Phi}}_e^{\mathrm{HDG}}\,\boldsymbol{n}_e\rangle_{\partial\Omega_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e}, \tag{6.49b}$$

$$(\boldsymbol{\nabla} q_e, \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} = 0, \tag{6.49c}$$

$$\begin{cases} \langle \hat{\boldsymbol{w}}_f, [\![\widehat{\boldsymbol{\Phi}}^{\mathrm{HDG}}\,\boldsymbol{n}]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f} = 0, & \text{if } \Gamma_f \subset \Gamma^{\circ}, \\ \langle \hat{\boldsymbol{w}}_f, \widehat{\boldsymbol{\Phi}}_{e_f}^{\mathrm{HDG}}\,\boldsymbol{n}_{e_f}\rangle_{\Gamma_f} = -\langle \hat{\boldsymbol{w}}_f, \boldsymbol{t}\rangle_{\Gamma_f}, & \text{if } \Gamma_f \subset \Gamma^{\mathrm{N}}, \end{cases} \tag{6.49d}$$

for all $(\boldsymbol{G}_e, \boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{L}}_e \times \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_e$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, and for all $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$ and all $f \in \{1{:}\mathtt{n_{fc}^{\circ N}}\}$. The numerical trace $\widehat{\boldsymbol{\Phi}}^{\mathrm{HDG}}$ is defined such that, for all $e \in \{1{:}\mathtt{n_{el}}\}$,

$$\widehat{\boldsymbol{\Phi}}_e^{\mathrm{HDG}}\,\boldsymbol{n}_e := \boldsymbol{\Phi}_e^{\mathrm{HDG}}\,\boldsymbol{n}_e + \boldsymbol{s}_e^{\mathrm{HDG}}(\boldsymbol{u}_e^{\mathrm{HDG}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}), \tag{6.50}$$

with $\boldsymbol{\Phi}_e^{\mathrm{HDG}}$ defined in (6.17), and $\boldsymbol{s}_e^{\mathrm{HDG}}$ defined in (6.20).

To discretize problem (6.1), HHO methods seek the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P} \times \hat{\boldsymbol{\mathcal{U}}}_{\mathrm{D}}$ such that the following holds true:

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{u}}^{\mathrm{HHO}}) - b^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, p^{\mathrm{HHO}}) = F(\underline{\boldsymbol{w}}), \tag{6.51a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{u}}^{\mathrm{HHO}}, q) = 0, \tag{6.51b}$$

for all $(\boldsymbol{w}, q, \hat{\boldsymbol{w}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P} \times \hat{\boldsymbol{\mathcal{U}}}_0$, where we have used the HHO notation $\underline{\boldsymbol{v}} := (\boldsymbol{v}, \hat{\boldsymbol{v}}) \in \underline{\boldsymbol{\mathcal{U}}} := \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_0$ or $\underline{\boldsymbol{\mathcal{U}}} := \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_{\mathrm{D}}$ depending on the context. The global problems are assembled cellwise by using the following expressions:

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{v}}) := \sum_{e\in\{1:\mathtt{n_{el}}\}} a_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e, \underline{\boldsymbol{v}}_e), \tag{6.52a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{v}}, q) := \sum_{e\in\{1:\mathtt{n_{el}}\}} b_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e, q_e), \tag{6.52b}$$

$$F(\underline{\boldsymbol{w}}) := \sum_{e\in\{1:\mathtt{n_{el}}\}} (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \sum_{\Gamma_f\subset\Gamma^{\mathrm{N}}} \langle \hat{\boldsymbol{w}}_f, \boldsymbol{t}\rangle_{\Gamma_f}, \tag{6.52c}$$

and the local bilinear forms and operators are defined as in (6.30).

6.2 Two-fluid Stokes problem

In this section, we compare the HHO and HDG formulations to discretize the two-fluid Stokes problem with surface tension at their interface, posed in a domain with a simple shape so that a fitted mesh can be used at the domain boundary. The mesh is unfitted with respect to the interface between the two fluids. Compared to the one-fluid case addressed in Section 6.1, we show that the unfitted HHO and HDG methods can additionally differ in the choice of the penalty method.

6.2.1 Model problem

As in Section 6.1, the computational domain $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$, $\mathtt{n_{sd}} \in \{2,3\}$, is a polyhedron. The two fluid setting is the same as in Chapters 2 and 4, namely the domain Ω is occupied by two immiscible, incompressible Stokes fluids separated by a single interface. Let $\Omega^i \subset \Omega$ the domain occupied by the fluid indexed by $i \in \{1,2\}$ such that $\overline{\Omega} = \overline{\Omega^1} \cup \overline{\Omega^2}$ and $\Omega^1 \cap \Omega^2 = \emptyset$. The interface is $\Upsilon := \partial\Omega^1 \cap \partial\Omega^2$. We assume that Υ is closed and does not touch the boundary $\partial\Omega$ of Ω. By convention, the index 1 refers to the interior subdomain Ω^1 such that $\partial\Omega^1 = \Upsilon$, and the index 2 refers to the exterior subdomain Ω^2 such that $\partial\Omega^2 = \partial\Omega \cup \Upsilon$. The interface Υ is assumed to be orientable, and such that a unit normal, $\boldsymbol{n}_\Upsilon$, can be defined everywhere on Υ. By convection, $\boldsymbol{n}_\Upsilon$ points from Ω^1 to Ω^2. Moreover, for all $i \in \{1,2\}$, we define $\boldsymbol{n}^i$ the unit outward normal to the domain Ω^i. On the boundary $\partial\Omega$ of Ω, we assume for simplicity that homogeneous Dirichlet boundary conditions are prescribed. Additionally, we impose the body force $\boldsymbol{f} \in \mathcal{L}^2(\Omega;\mathbb{R}^{\mathtt{n_{sd}}})$ in the domain Ω and the surface force $\boldsymbol{g}_\Upsilon \in \mathcal{L}^2(\Upsilon;\mathbb{R}^{\mathtt{n_{sd}}})$ at the interface Υ (resulting, e.g., from Laplace's law for surface tension).

As in Section 6.1, we consider the so-called *full gradient formulation* of the Stokes interface problem (the Cauchy stress formulation was considered in Chapter 2) such that, given the interface Υ, we seek the velocity and pressure fields $(\boldsymbol{u}^i, p^i) \in \mathcal{H}^1(\Omega^i;\mathbb{R}^{\mathtt{n_{sd}}}) \times \mathcal{L}^2(\Omega^i)$, $i \in \{1,2\}$, that satisfy

$$-\nabla\cdot(\mu^i\nabla\boldsymbol{u}^i - p^i\boldsymbol{I}_{\mathtt{n_{sd}}}) = \boldsymbol{f}^i \quad \text{in } \Omega^i, \tag{6.53a}$$

$$\nabla\cdot\boldsymbol{u}^i = 0 \quad \text{in } \Omega^i, \tag{6.53b}$$

$$\boldsymbol{u}^2 = \boldsymbol{0} \quad \text{on } \partial\Omega, \tag{6.53c}$$

$$[\![\boldsymbol{u}]\!]_\Upsilon^{\mathrm{HHO}} = \boldsymbol{0}, \quad [\![\mu^i\nabla\boldsymbol{u}^i - p^i\boldsymbol{I}_{\mathtt{n_{sd}}}]\!]_\Upsilon^{\mathrm{HHO}}\boldsymbol{n}_\Upsilon = \boldsymbol{g}_\Upsilon \quad \text{on } \Upsilon, \tag{6.53d}$$

where $\boldsymbol{f}^i := \boldsymbol{f}|_{\Omega^i} \in \mathcal{L}^2(\Omega^i;\mathbb{R}^{\mathtt{n_{sd}}})$ denotes the restriction of $\boldsymbol{f}$ to Ω^i, and μ^i is the viscosity of the fluid i. Moreover, $[\![\odot]\!]_\Upsilon^{\mathrm{HHO}}$ is the HHO jump operator defined in (6.15). The Stokes interface problem is uniquely solvable up to a global additive constant on the pressure, which we fix by imposing

$$\sum_{i\in\{1:2\}}\int_{\Omega^i} p^i = 0. \tag{6.54}$$

6.2.2 Discrete setting

We consider a shape-regular mesh composed of $\mathtt{n_{el}}$ disjoint (open) subdomains Ω_e such that Ω is exactly covered, i.e.,

$$\overline{\Omega} = \bigcup_{e\in\{1:\mathtt{n_{el}}\}} \overline{\Omega_e}. \tag{6.55}$$

Since the mesh is unfitted, the interface Υ can cut arbitrarily through some of the mesh cells. Thus, for all $e \in \{1{:}\mathtt{n_{el}}\}$, we define Ω_e^i and $\partial\Omega_e^i$ the portion of Ω_e and $\partial\Omega_e$ that belongs to the fluid indexed by i, that is,

$$\Omega_e^i := \Omega_e \cap \Omega^i, \qquad \partial\Omega_e^i := \partial\Omega_e \cap (\overline{\Omega^i} \setminus \Upsilon), \qquad \forall i \in \{1,2\}. \tag{6.56}$$

Notice that $\partial\Omega_e^i$ is contained in the boundary of Ω_e^i but differs from it if the cell Ω_e is cut by the interface Υ. Similarly, for all $i \in \{1,2\}$, we set $\Gamma^i := \Gamma \cap (\Omega^i \setminus \Upsilon)$, and $\Gamma_f^i := \Gamma_f \cap (\Omega^i \setminus \Upsilon)$,

for all $f \in \{1{:}\mathtt{n}_{\mathtt{fc}}^{\circ}\}$. In general, the interface intersects every face at most at one point so that $\Gamma^i = \Gamma \cap \Omega^i$ and $\Gamma_f^i = \Gamma_f \cap \Omega^i$. Moreover, let $\Upsilon_e := \Upsilon \cap \Omega_e$ be the portion of Υ inside Ω_e. In what follows, we assume that the mesh is such that $\mu^i|_{\Omega_e^i}$ is constant for all $i \in \{1,2\}$ and all $e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}$.

Consistently with the one-fluid case, both HHO and HDG methods consider the following approximation spaces in each subdomain: For all $i \in \{1,2\}$,

$$\boldsymbol{\mathcal{L}}^i := \{\boldsymbol{G}^i := (\boldsymbol{G}_e^i)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}});\ \boldsymbol{G}_e^i \in \boldsymbol{\mathcal{L}}_e^i,\ \forall e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.57a}$$

$$\boldsymbol{\mathcal{U}}^i := \{\boldsymbol{v}^i := (\boldsymbol{v}_e^i)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}});\ \boldsymbol{v}_e^i \in \boldsymbol{\mathcal{U}}_e^i,\ \forall e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.57b}$$

$$\mathcal{P}^i := \{q^i := (q_e^i)_{e\in\{1:\mathtt{n}_{\mathtt{el}}\}} \in \mathcal{L}^2(\Omega^i;\mathbb{R});\ q_e^i \in \mathcal{P}_e^i,\ \forall e \in \{1{:}\mathtt{n}_{\mathtt{el}}\}\}, \tag{6.57c}$$

$$\widehat{\boldsymbol{\mathcal{U}}}^i := \{\widehat{\boldsymbol{v}}^i := (\widehat{\boldsymbol{v}}_f^i)_{f\in\{1:\mathtt{n}_{\mathtt{fc}}\}} \in \mathcal{L}^2(\Gamma^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}});\ \widehat{\boldsymbol{v}}_f^i \in \widehat{\boldsymbol{\mathcal{U}}}_f^i,\ \forall f \in \{1{:}\mathtt{n}_{\mathtt{fc}}\}\}, \tag{6.57d}$$

where $\boldsymbol{\mathcal{L}}_e^i, \boldsymbol{\mathcal{U}}_e^i, \mathcal{P}_e^i, \widehat{\boldsymbol{\mathcal{U}}}_f^i$ are finite-dimensional spaces composed of polynomials from Ω_e^i to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}}$, from Ω_e^i to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}$, from Ω_e^i to $\mathbb{R}$, and from Γ_f^i to $\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}$, respectively. Letting $k \geq 0$ be the polynomial degree, the prototypical example we consider henceforth is

$$\boldsymbol{\mathcal{L}}_e^i := \mathbb{P}^k(\Omega_e^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}\times\mathtt{n}_{\mathtt{sd}}}), \qquad \mathcal{P}_e^i := \mathbb{P}^k(\Omega_e^i;\mathbb{R}), \qquad \widehat{\boldsymbol{\mathcal{U}}}_f^i := \mathbb{P}^k(\Gamma_f^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), \tag{6.58}$$

and

$$\boldsymbol{\mathcal{U}}_e^i := \begin{cases} \mathbb{P}^k(\Omega_e^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), & \text{equal-order setting,} \\ \mathbb{P}^{k+1}(\Omega_e^i;\mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}), & \text{mixed-order setting.} \end{cases} \tag{6.59}$$

Notice that there is no discrete unknown attached to the interface Υ. We denote by $\widehat{\boldsymbol{v}}_e^i$ the collection of $(\widehat{\boldsymbol{v}}_f^i)_{\Gamma_f^i\subset\partial\Omega_e^i}$, so that

$$\widehat{\boldsymbol{v}}_e^i := (\widehat{\boldsymbol{v}}_f^i)_{\Gamma_f^i\subset\partial\Omega_e^i} \in \widehat{\boldsymbol{\mathcal{U}}}_e^i := \bigtimes_{\Gamma_f^i\subset\partial\Omega_e^i} \widehat{\boldsymbol{\mathcal{U}}}_f^i, \tag{6.60}$$

and we write

$$\widehat{\boldsymbol{\mathcal{U}}}_e^{*,i} := \{\boldsymbol{v}_e^i|_{\partial\Omega_e^i} - \widehat{\boldsymbol{v}}_e^i;\ \forall \boldsymbol{v}_e^i \in \boldsymbol{\mathcal{U}}_e^i,\ \forall \widehat{\boldsymbol{v}}_e^i \in \widehat{\boldsymbol{\mathcal{U}}}_e^i\}. \tag{6.61}$$

To handle uncut and cut cells in a single formalism, we set

$$\boldsymbol{\mathcal{L}}_e := \boldsymbol{\mathcal{L}}_e^i, \qquad \boldsymbol{\mathcal{U}}_e := \boldsymbol{\mathcal{U}}_e^i, \qquad \mathcal{P}_e := \mathcal{P}_e^i, \qquad \widehat{\boldsymbol{\mathcal{U}}}_f := \widehat{\boldsymbol{\mathcal{U}}}_f^i, \tag{6.62}$$

for every uncut cell Ω_e inside Ω^i, and every uncut face Γ_f inside Ω^i, with $i \in \{1,2\}$, and

$$\boldsymbol{\mathcal{L}}_e := \boldsymbol{\mathcal{L}}_e^1 \times \boldsymbol{\mathcal{L}}_e^2, \qquad \boldsymbol{\mathcal{U}}_e := \boldsymbol{\mathcal{U}}_e^1 \times \boldsymbol{\mathcal{U}}_e^2, \qquad \mathcal{P}_e := \mathcal{P}_e^1 \times \mathcal{P}_e^2, \qquad \widehat{\boldsymbol{\mathcal{U}}}_f := \widehat{\boldsymbol{\mathcal{U}}}_f^1 \times \widehat{\boldsymbol{\mathcal{U}}}_f^2, \tag{6.63}$$

for every cut cell Ω_e and every cut face Γ_f. The same notation is adopted for the variables, e.g., for a generic cut cell Ω_e, we write

$$\boldsymbol{v}_e = (\boldsymbol{v}_{e,1}, \boldsymbol{v}_{e,2}), \qquad \boldsymbol{v}_{e,i} \in \boldsymbol{\mathcal{U}}_e^i, \qquad \forall i \in \{1,2\}. \tag{6.64}$$

For brevity, we only consider the strong enforcement of the homogeneous Dirichlet boundary condition, and we set

$$\hat{\boldsymbol{U}}_0 = \hat{\boldsymbol{U}}^1 \times \hat{\boldsymbol{U}}_0^2, \qquad \hat{\boldsymbol{U}}_0^2 := \{\hat{\boldsymbol{v}}^2 \in \hat{\boldsymbol{U}}^2;\ \hat{\boldsymbol{v}}^2 = \boldsymbol{0} \text{ on } \partial\Omega\}. \tag{6.65}$$

Finally, it is convenient to introduce the *average operator* at any portion of interface Υ_e. For a generic piecewise smooth function χ that can be scalar-, vector, or tensor-valued, the average operator is defined as

$$\{\chi\}_{\Upsilon_e} = \frac{1}{2}\Big((\chi|_{\Omega_e^1})|_{\Upsilon_e} + (\chi|_{\Omega_e^2})|_{\Upsilon_e}\Big). \tag{6.66}$$

Remark 6.3 (Viscosity-robust average operator). The average operator defined in (6.66) does not ensure robust error estimates for high-contrast viscosity coefficients (see [22], [54]). When μ^1 and μ^2 are such that $\mu^1 \ll \mu^2$, or vice versa, it is preferable to use the weighted average

$$\{\chi\}_{\Upsilon_e} = \alpha_e^1(\chi|_{\Omega_e^1})|_{\Upsilon_e} + \alpha_e^2(\chi|_{\Omega_e^2})|_{\Upsilon_e},$$

where the weights depend on the viscosity ratio as follows:

$$\alpha_e^1 := \frac{\mu_e^2}{\mu_e^1 + \mu_e^2}, \qquad \alpha_e^2 := \frac{\mu_e^1}{\mu_e^1 + \mu_e^2},$$

consistently with the HHO method presented in Chapter 2.

Remark 6.4 (Pressure mean value). Owing to the prescribed Dirichlet boundary conditions on the entire boundary $\partial\Omega$, the discrete pressure is actually sought in the global space

$$\mathcal{P}^0 := \{q \in \mathcal{P};\ \sum_{i\in\{1:2\}} \int_{\Omega^i} q^i = 0\}.$$

The corresponding local spaces are denoted by $\mathcal{P}_e^0$, for all $e \in \{1{:}\mathtt{n_{el}}\}$.

6.2.3 HDG formulation

In this section, we present the HDG formulation of the Stokes problem (6.53).

6.2.3.1 Global formulation

It is convenient to define the HDG viscous stress tensor such that

$$\boldsymbol{\Phi}_{e,i}^{\mathrm{HDG}} := \sqrt{\mu_e^i}\boldsymbol{L}_{e,i}^{\mathrm{HDG}} + p_{e,i}^{\mathrm{HDG}}\boldsymbol{I}_{\mathtt{n_{sd}}}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}, \qquad \forall i \in \{1,2\}. \tag{6.67}$$

To discretize problem (6.53), HDG methods seek the quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ such that the following holds true:

$$\begin{aligned} &-(\boldsymbol{G}_e, \boldsymbol{L}_e^{\mathrm{HDG}})_{\Omega_e} + (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\,\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle \sqrt{\mu_e}\,\boldsymbol{G}_e\,\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\rangle_{\partial\Omega_e} \\ &\qquad - \langle [\![\sqrt{\mu_e}\,\boldsymbol{G}_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HDG}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} = 0, \end{aligned} \tag{6.68a}$$

$$\begin{aligned} &-(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HDG}})_{\Omega_e} + \langle \boldsymbol{w}_e, \widehat{\boldsymbol{\Phi}}_e^{\mathrm{HDG}}\boldsymbol{n}_e\rangle_{\partial\Omega_e} + \langle [\![\boldsymbol{w}_e \otimes \boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{\Phi}_e^{\mathrm{HDG}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} \\ &\qquad + \tau_{\Upsilon_e}^{\mathrm{HDG}}\langle [\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{u}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon\rangle_{\Upsilon_e}, \end{aligned} \tag{6.68b}$$

$$(\boldsymbol{\nabla} q_e, \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} - \langle [\![q_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HDG}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} = 0, \tag{6.68c}$$

$$\langle \hat{\boldsymbol{w}}_f, [\![\widehat{\boldsymbol{\Phi}}^{\mathrm{HDG}}\boldsymbol{n}]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f} = 0, \tag{6.68d}$$

for all $(\boldsymbol{G}_e, \boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{L}}_e \times \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_e^0$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, and for all $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$ and all $f \in \{1{:}\mathtt{n_{fc}^\circ}\}$. The numerical trace $\widehat{\boldsymbol{\Phi}}^{\mathrm{HDG}}$ is defined such that, for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1,2\}$,

$$\widehat{\boldsymbol{\Phi}}_{e,i}^{\mathrm{HDG}}\boldsymbol{n}_e := \boldsymbol{\Phi}_{e,i}^{\mathrm{HDG}}\boldsymbol{n}_e + \boldsymbol{s}_{e,i}^{\mathrm{HDG}}(\boldsymbol{u}_{e,i}^{\mathrm{HDG}}|_{\partial\Omega_e^i} - \hat{\boldsymbol{u}}_{e,i}^{\mathrm{HDG}}), \tag{6.69}$$

where $\boldsymbol{s}_{e,i}^{\mathrm{HDG}} : \hat{\boldsymbol{\mathcal{U}}}_e^{*,i} \to \hat{\boldsymbol{\mathcal{U}}}_e^i$ is a linear stabilization operator which is generally taken to be of the form

$$\boldsymbol{s}_{e,i}^{\mathrm{HDG}}(\hat{\boldsymbol{\delta}}) := \tau_{e,i}^{\mathrm{HDG}} \begin{cases} \hat{\boldsymbol{\delta}}, & \text{equal-order setting,} \\ \hat{\Pi}_e^i(\hat{\boldsymbol{\delta}}), & \text{mixed-order setting,} \end{cases} \tag{6.70}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^{*,i}$, with a positive weight $\tau_{e,i}^{\mathrm{HDG}} > 0$. It is important to notice that the unfitted HDG method presented in Chapter 4 and that we are discussing here does not employ any penalty term on the jumps across the interface, i.e., we set

$$\tau_{\Upsilon_e}^{\mathrm{HDG}} := 0. \tag{6.71}$$

The penalty term in (6.68b) has been introduced only for comparison reasons with the HHO formulation. Finally, we notice that (6.68b) can be rewritten as

$$\begin{aligned} &(\boldsymbol{w}_e, \boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_e^{\mathrm{HDG}})_{\Omega_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^{\mathrm{HDG}}(\boldsymbol{u}_e^{\mathrm{HDG}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HDG}})\right\rangle_{\partial\Omega_e} - \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, [\![\boldsymbol{\Phi}_e^{\mathrm{HDG}}\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}\rangle_{\Upsilon_e} \\ &\qquad + \tau_{\Upsilon_e}^{\mathrm{HDG}}\langle [\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{u}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon\rangle_{\Upsilon_e}. \end{aligned} \tag{6.72}$$

6.2.3.2 Static condensation

Static condensation is a procedure that allows one to eliminate locally (in every mesh cell) the discrete unknowns $\boldsymbol{L}_e$ and $\boldsymbol{u}_e$, as well as p_e up to one constant. Thus, the only globally coupled unknowns are the skeleton unknowns $\hat{\boldsymbol{u}}_f$ and one constant pressure per mesh cell. To handle the pressure decomposition, we write, as before,

$$\mathcal{P}^0 = \mathcal{P}_0^0 \oplus \mathcal{P}_\perp^0, \tag{6.73}$$

and, for all $q \in \mathcal{P}^0$, we write $q = q_0 + q_\perp$ and $q_e = q_{0,e} + q_{\perp,e}$, for all $e \in \{1{:}\mathtt{n_{el}}\}$.

Proposition 8 (Local and global problems). The quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times$

$\boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.68) if and only if:

(i) The local variables $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p_\perp^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}_\perp^0$ solve the following local problems as a function of the global variables $(\hat{\boldsymbol{u}}^{\mathrm{HDG}}, p_0^{\mathrm{HDG}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$:

$$-(\boldsymbol{G}_e, \boldsymbol{L}_e^{\mathrm{HDG}})_{\Omega_e} + (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\,\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle [\![\sqrt{\mu_e}\,\boldsymbol{G}_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HDG}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} = \langle \sqrt{\mu_e}\,\boldsymbol{G}_e\,\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\rangle_{\partial\Omega_e}, \tag{6.74a}$$

$$(\boldsymbol{w}_e, \boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_{\perp,e}^{\mathrm{HDG}})_{\Omega_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^{\mathrm{HDG}}(\boldsymbol{u}_e^{\mathrm{HDG}}|_{\partial\Omega_e})\right\rangle_{\partial\Omega_e} - \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, [\![\boldsymbol{\Phi}_{\perp,e}^{\mathrm{HDG}}\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}\rangle_{\Upsilon_e} + \tau_{\Upsilon_e}^{\mathrm{HDG}}\langle [\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{u}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e} = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon\rangle_{\Upsilon_e} + \left\langle \boldsymbol{w}_e, \boldsymbol{s}_e^{\mathrm{HDG}}(\hat{\boldsymbol{u}}_e^{\mathrm{HDG}})\right\rangle_{\partial\Omega_e}, \tag{6.74b}$$

$$(\boldsymbol{\nabla} q_e, \boldsymbol{u}_e^{\mathrm{HDG}})_{\Omega_e} - \langle [\![q_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HDG}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} = \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e}, \tag{6.74c}$$

for all $(\boldsymbol{G}_e, \boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{L}}_e \times \boldsymbol{\mathcal{U}}_e \times \mathcal{P}_{\perp,e}^0$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, where

$$\boldsymbol{\Phi}_{\perp,e}^{\mathrm{HDG}} := \sqrt{\mu_e}\boldsymbol{L}_e^{\mathrm{HDG}} + p_{\perp,e}^{\mathrm{HDG}}\boldsymbol{I}_{\mathtt{n_{sd}}} = \boldsymbol{\Phi}_e^{\mathrm{HDG}} - p_{0,e}^{\mathrm{HDG}}\boldsymbol{I}_{\mathtt{n_{sd}}}; \tag{6.75}$$

(ii) The global variables $(\hat{\boldsymbol{u}}^{\mathrm{HDG}}, p_0^{\mathrm{HDG}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}_0^0$ solve the following global problems where the local variables $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p_\perp^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}_\perp^0$ result from (6.74):

$$\langle \hat{\boldsymbol{w}}_f, [\![p_0^{\mathrm{HDG}}\boldsymbol{n} - \boldsymbol{s}^{\mathrm{HDG}}(\hat{\boldsymbol{u}}^{\mathrm{HDG}})\boldsymbol{n}]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f} = -\langle \hat{\boldsymbol{w}}_f, [\![\boldsymbol{\Phi}_\perp^{\mathrm{HDG}}\boldsymbol{n} + \boldsymbol{s}^{\mathrm{HDG}}(\boldsymbol{u}^{\mathrm{HDG}})\boldsymbol{n}]\!]_{\Gamma_f}^{\mathrm{HDG}}\rangle_{\Gamma_f}, \tag{6.76a}$$

$$\langle 1, \hat{\boldsymbol{u}}_e^{\mathrm{HDG}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} = 0, \tag{6.76b}$$

where (6.76a) holds for all $\hat{\boldsymbol{w}}_f \in \hat{\boldsymbol{\mathcal{U}}}_f$ and all $f \in \{1{:}\mathtt{n_{fc}^\circ}\}$, and (6.76b) holds for all $e \in \{1{:}\mathtt{n_{el}}\}$.

6.2.4 HHO formulation

In this section, we present the HHO formulation of the Stokes problem (6.53). For a generic pair $(\boldsymbol{v}, \hat{\boldsymbol{v}}) \in \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_0$, we use as before the HHO notation $\underline{\boldsymbol{v}} := (\boldsymbol{v}, \hat{\boldsymbol{v}}) \in \underline{\boldsymbol{\mathcal{U}}} := \boldsymbol{\mathcal{U}} \times \hat{\boldsymbol{\mathcal{U}}}_0$. Correspondingly, for the local space, we define $\underline{\boldsymbol{\mathcal{U}}}_e^i := \boldsymbol{\mathcal{U}}_e^i \times \hat{\boldsymbol{\mathcal{U}}}_e^i$, and $\underline{\boldsymbol{\mathcal{U}}}_e := \boldsymbol{\mathcal{U}}_e \times \hat{\boldsymbol{\mathcal{U}}}_e$, for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1, 2\}$. Recall that the unfitted HHO method is devised only in the mixed-order setting, since optimal error estimates are proven only in this case.

6.2.4.1 Global formulation

To discretize problem (6.53), HHO methods seek the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ such that the following holds true:

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{u}}^{\mathrm{HHO}}) - b^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, p^{\mathrm{HHO}}) = F(\underline{\boldsymbol{w}}), \tag{6.77a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{u}}^{\mathrm{HHO}}, q) = 0, \tag{6.77b}$$

for all $(\boldsymbol{w}, q, \hat{\boldsymbol{w}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$. The global problems are assembled cellwise as

$$a^{\mathrm{HHO}}(\underline{\boldsymbol{w}}, \underline{\boldsymbol{v}}) := \sum_{e\in\{1:\mathtt{n_{el}}\}} a_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e, \underline{\boldsymbol{v}}_e), \tag{6.78a}$$

$$b^{\mathrm{HHO}}(\underline{\boldsymbol{v}}, q) := \sum_{e\in\{1:\mathtt{n_{el}}\}} b_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e, q_e), \tag{6.78b}$$

$$F(\underline{\boldsymbol{w}}) := \sum_{e\in\{1:\mathtt{n_{el}}\}} \Big\{(\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle\{\boldsymbol{w}\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon\rangle_{\Upsilon_e}\Big\}, \tag{6.78c}$$

where the local bilinear forms are

$$\begin{aligned} a_e^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e, \underline{\boldsymbol{v}}_e) := \sum_{i\in\{1:2\}} \Big\{ & \mu_e^i\Big(\boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{w}}_e), \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e)\Big)_{\Omega_e^i} \\ & + \tau_{e,i}^{\mathrm{HHO}}\Big\langle \boldsymbol{s}_{e,i}^{\mathrm{HHO}}(\boldsymbol{w}_{e,i|\partial\Omega_e^i} - \hat{\boldsymbol{w}}_{e,i}), \boldsymbol{s}_{e,i}^{\mathrm{HHO}}(\boldsymbol{v}_{e,i|\partial\Omega_e^i} - \hat{\boldsymbol{v}}_{e,i})\Big\rangle_{\partial\Omega_e^i}\Big\} \\ & + \tau_{\Upsilon_e}^{\mathrm{HHO}}\langle[\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{u}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e}, \end{aligned} \tag{6.79a}$$

$$b_e^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e, q_e) := \sum_{i\in\{1:2\}} (D_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e), q_{e,i})_{\Omega_e^i}, \tag{6.79b}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$, with positive weights $\tau_{\Upsilon_e}^{\mathrm{HHO}} > 0$ and $\tau_{e,i}^{\mathrm{HHO}} > 0$ (typically, $\tau_{\Upsilon_e}^{\mathrm{HHO}} := \min(\mu_e^1, \mu_e^2)\, h_e^{-1}$ and $\tau_{e,i}^{\mathrm{HHO}} := \mu_e^i h_e^{-1}$). The following operators have been used for all $i \in \{1, 2\}$: The gradient reconstruction operator $\boldsymbol{L}_{e,i}^{\mathrm{HHO}} : \underline{\boldsymbol{\mathcal{U}}}_e \to \boldsymbol{\mathcal{L}}_e^i$ such that

$$(\boldsymbol{G}, \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e))_{\Omega_e^i} := (\boldsymbol{G}, \boldsymbol{\nabla}\boldsymbol{v}_{e,i})_{\Omega_e^i} + \langle \boldsymbol{G}\boldsymbol{n}_e, \hat{\boldsymbol{v}}_{e,i} - \boldsymbol{v}_{e,i|\partial\Omega_e^i}\rangle_{\partial\Omega_e^i} - \frac{1}{2}\langle \boldsymbol{G}\boldsymbol{n}_\Upsilon, [\![\boldsymbol{v}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e}, \tag{6.80}$$

for all $\underline{\boldsymbol{v}}_e \in \underline{\boldsymbol{\mathcal{U}}}_e$, and all $\boldsymbol{G} \in \boldsymbol{\mathcal{L}}_e^i$, the divergence reconstruction operator such that

$$D_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e) := \boldsymbol{I}_{\mathtt{n_{sd}}} : \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e), \tag{6.81}$$

and the stabilization operator $\boldsymbol{s}_{e,i}^{\mathrm{HHO}} : \hat{\boldsymbol{\mathcal{U}}}_e^{*,i} \to \hat{\boldsymbol{\mathcal{U}}}_e^i$ such that

$$\boldsymbol{s}_{e,i}^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}}) := \hat{\Pi}_e^i(\hat{\boldsymbol{\delta}}), \tag{6.82}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^{*,i}$. We notice that when the gradient reconstruction operator (6.80) is evaluated in an entire cut cell, we have

$$\sum_{i\in\{1:2\}} (\boldsymbol{G}, \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{v}}_e))_{\Omega_e^i} := (\boldsymbol{G}, \boldsymbol{\nabla}\boldsymbol{v}_e)_{\Omega_e} + \langle \boldsymbol{G}\boldsymbol{n}_e, \hat{\boldsymbol{v}}_e - \boldsymbol{v}_{e|\partial\Omega_e}\rangle_{\partial\Omega_e} - \langle\{\boldsymbol{G}\}_{\Upsilon_e}\boldsymbol{n}_\Upsilon, [\![\boldsymbol{v}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e}, \tag{6.83}$$

for all $\boldsymbol{G} \in \boldsymbol{\mathcal{L}}_e = \boldsymbol{\mathcal{L}}_e^1 \times \boldsymbol{\mathcal{L}}_e^2$. Finally, it is convenient to define the HHO viscous stress tensor such that

$$\boldsymbol{\Phi}_{e,i}^{\mathrm{HHO}} := -\mu_e^i \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) + p_{e,i}^{\mathrm{HHO}} \boldsymbol{I}_{\mathtt{n_{sd}}}, \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}, \qquad \forall i \in \{1, 2\}. \tag{6.84}$$

6.2.4.2 Static condensation

Recall the decomposition (6.73) of the pressure space, i.e., $\mathcal{P}^0 = \mathcal{P}^0_0 \oplus \mathcal{P}^0_\perp$. Notice that

$$b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_{0,e}^{\mathrm{HHO}}) = (\boldsymbol{L}^{\mathrm{HHO}}(\boldsymbol{w}_e, \boldsymbol{0}), p_{0,e}^{\mathrm{HHO}} \boldsymbol{I}_{\mathtt{n_{sd}}})_{\Omega_e} = -\langle \{p_{0,e}^{\mathrm{HHO}}\}_{\Upsilon_e} \boldsymbol{n}_\Upsilon, [\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}} \rangle_{\Upsilon_e}, \tag{6.85}$$

for all $\boldsymbol{w}_e \in \boldsymbol{\mathcal{U}}_e$ and all $e \in \{1{:}\mathtt{n_{el}}\}$, as a consequence of (6.80) after integration by parts. In particular, (6.85) gives

$$b_e^{\mathrm{HHO}}((\boldsymbol{u}_e^{\mathrm{HHO}}, \boldsymbol{0}), 1) = \langle 1, [\![\boldsymbol{u}_e^{\mathrm{HHO}}]\!]_{\Upsilon_e}^{\mathrm{HHO}} \boldsymbol{n}_\Upsilon \rangle_{\Upsilon_e}. \tag{6.86}$$

Proposition 9 (Local and global problems). The triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.77) if and only if:

(i) The local variables $(\boldsymbol{u}^{\mathrm{HHO}}, p_\perp^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0_\perp$ solve the following local problems as a function of the global variables $(\hat{\boldsymbol{u}}^{\mathrm{HHO}}, p_0^{\mathrm{HHO}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}^0_0$:

$$\begin{aligned} &a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), (\boldsymbol{u}_e^{\mathrm{HHO}}, \boldsymbol{0})) - b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_{\perp,e}^{\mathrm{HHO}}) + \langle \{p_{0,e}^{\mathrm{HHO}}\}_{\Upsilon_e} \boldsymbol{n}_\Upsilon, [\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}} \rangle_{\Upsilon_e} \\ &\qquad = (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle \{\boldsymbol{w}_e\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon \rangle_{\Upsilon_e} - a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), (\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})), \end{aligned} \tag{6.87a}$$

$$b_e^{\mathrm{HHO}}((\boldsymbol{u}_e^{\mathrm{HHO}}, \boldsymbol{0}), q_e) = -b_e^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}), q_e), \tag{6.87b}$$

for all $(\boldsymbol{w}_e, q_e) \in \boldsymbol{\mathcal{U}}_e \times \mathcal{P}^0_{\perp,e}$ and all $e \in \{1{:}\mathtt{n_{el}}\}$;

(ii) The global variables $(\hat{\boldsymbol{u}}^{\mathrm{HHO}}, p_0^{\mathrm{HHO}}) \in \hat{\boldsymbol{\mathcal{U}}}_0 \times \mathcal{P}^0_0$ solve the following global problems where the local variables $(\boldsymbol{u}^{\mathrm{HHO}}, p_\perp^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0_\perp$ result from (6.87):

$$\begin{aligned} &a^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), (\boldsymbol{0}, \hat{\boldsymbol{u}}^{\mathrm{HHO}})) - b^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), p_0^{\mathrm{HHO}}) \\ &\qquad = -a^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), (\boldsymbol{u}^{\mathrm{HHO}}, \boldsymbol{0})) + b^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{w}}), p_\perp^{\mathrm{HHO}}), \end{aligned} \tag{6.88a}$$

$$b_e^{\mathrm{HHO}}((\boldsymbol{0}, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}), 1) = \langle 1, [\![\boldsymbol{u}_e^{\mathrm{HHO}}]\!]_{\Upsilon_e}^{\mathrm{HHO}} \boldsymbol{n}_\Upsilon \rangle_{\Upsilon_e}, \tag{6.88b}$$

where (6.88a) holds for all $\hat{\boldsymbol{w}} \in \hat{\boldsymbol{\mathcal{U}}}_0$ and (6.88b) holds for all $e \in \{1{:}\mathtt{n_{el}}\}$.

6.2.5 Bridging HHO and HDG

Within the above setting, HDG and HHO methods depend on the choice of the local spaces $\boldsymbol{\mathcal{L}}_e^i$, $\boldsymbol{\mathcal{U}}_e^i$, $\mathcal{P}_e^i$, $\hat{\boldsymbol{\mathcal{U}}}_f^i$, of the stabilization operators $\boldsymbol{s}_{e,i}^{\mathrm{HDG}}$ and $\boldsymbol{s}_{e,i}^{\mathrm{HHO}}$ acting on $\partial\Omega_e^i$, and of the penalty terms $\tau_{\Upsilon_e}^{\mathrm{HDG}}$ and $\tau_{\Upsilon_e}^{\mathrm{HHO}}$ acting on Υ_e. Recall that the unfitted HHO method is devised only in the mixed-order setting; thus, we focus only on this case in the next proposition.

Proposition 10 (From HHO to HDG). Define the local polynomial spaces

$$\boldsymbol{\mathcal{L}}_e^i = \mathbb{P}^k(\Omega_e^i; \mathbb{R}^{\mathtt{n_{sd}} \times \mathtt{n_{sd}}}), \quad \boldsymbol{\mathcal{U}}_e^i = \mathbb{P}^{k+1}(\Omega_e^i; \mathbb{R}^{\mathtt{n_{sd}};\mathbb{R}}), \quad \mathcal{P}_e^i = \mathbb{P}^k(\Omega_e^i), \quad \hat{\boldsymbol{\mathcal{U}}}_f^i = \mathbb{P}^k(\Gamma_f^i; \mathbb{R}^{\mathtt{n_{sd}}}). \tag{6.89}$$

If the triple $(\boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.77) with the stabilization operator (6.82), then setting

$$\boldsymbol{Z}_{e,i}^{\mathrm{HHO}} := -\sqrt{\mu_e^i} \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}), \tag{6.90}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1, 2\}$, the quadruple $(\boldsymbol{Z}^{\mathrm{HHO}}, \boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times$

$\hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.68) with the stabilization operator

$$\boldsymbol{s}_{e,i}^{\mathrm{HDG}}(\hat{\boldsymbol{\delta}}) = \tau_{e,i}^{\mathrm{HHO}} \hat{\Pi}_e^i(\hat{\boldsymbol{\delta}}), \tag{6.91}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^{*,i}$ on the internal faces, with the penalty weight

$$\tau_{\Upsilon_e}^{\mathrm{HDG}} = \tau_{\Upsilon_e}^{\mathrm{HHO}}, \tag{6.92}$$

at the interface, and with the numerical flux trace such that

$$\widehat{\boldsymbol{\Phi}}_{e,i}^{\mathrm{HDG}} \boldsymbol{n}_e := \boldsymbol{\Phi}_{e,i}^{\mathrm{HHO}} \boldsymbol{n}_e + \boldsymbol{s}_{e,i}^{\mathrm{HDG}}(\boldsymbol{u}_{e,i}^{\mathrm{HHO}}|_{\partial\Omega_e^i} - \hat{\boldsymbol{u}}_{e,i}^{\mathrm{HHO}}), \tag{6.93}$$

for all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $i \in \{1,2\}$, where $\boldsymbol{\Phi}_{e,i}^{\mathrm{HHO}} := \sqrt{\mu_e^i}\boldsymbol{Z}_{e,i}^{\mathrm{HHO}} + p_{e,i}^{\mathrm{HHO}} \boldsymbol{I}_{\mathtt{n_{sd}}}$.

Proposition 11 (From HDG to HHO). Define the local polynomial spaces

$$\boldsymbol{\mathcal{L}}_e^i = \mathbb{P}^k(\Omega_e^i; \mathbb{R}^{\mathtt{n_{sd}} \times \mathtt{n_{sd}}}), \quad \boldsymbol{\mathcal{U}}_e^i = \mathbb{P}^\ell(\Omega_e^i; \mathbb{R}^{\mathtt{n_{sd}};\mathbb{R}}), \quad \mathcal{P}_e^i = \mathbb{P}^k(\Omega_e^i), \quad \hat{\boldsymbol{\mathcal{U}}}_f^i = \mathbb{P}^k(\Gamma_f;^i \mathbb{R}^{\mathtt{n_{sd}}}), \tag{6.94}$$

with either $\ell = k$ (equal-order) or $\ell = k+1$ (mixed-order).
If the quadruple $(\boldsymbol{L}^{\mathrm{HDG}}, \boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HDG problem (6.68) with $\ell = k+1$ and with the stabilization operator (6.70), then

$$\boldsymbol{L}_{e,i}^{\mathrm{HDG}} = -\sqrt{\mu_e^i}\, \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}), \qquad \forall e \in \{1{:}\mathtt{n_{el}}\}, \qquad \forall i \in \{1,2\}, \tag{6.95}$$

with the operator $\boldsymbol{L}_{e,i}^{\mathrm{HHO}}$ defined in (6.80), and the triple $(\boldsymbol{u}^{\mathrm{HDG}}, p^{\mathrm{HDG}}, \hat{\boldsymbol{u}}^{\mathrm{HDG}}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ solves the HHO problem (6.77) with the weight $\tau_{e,i}^{\mathrm{HHO}} = \tau_{e,i}^{\mathrm{HDG}}$, and stabilization operator

$$\boldsymbol{s}_{e,i}^{\mathrm{HHO}}(\hat{\boldsymbol{\delta}}) = \hat{\Pi}_e^i(\hat{\boldsymbol{\delta}}), \tag{6.96}$$

for all $\hat{\boldsymbol{\delta}} \in \hat{\boldsymbol{\mathcal{U}}}_e^{*,i}$ at the internal faces, and the penalty weight

$$\tau_{\Upsilon_e}^{\mathrm{HHO}} = 0 \tag{6.97}$$

at the interface.

Proof. We only prove Proposition 10 since the proof of Proposition 11 is similar. Moreover, we only detail the effects of the interface terms, and we rely on the proof of Proposition 5 for the one-fluid case for more details. We need to prove that the quadruple $(\boldsymbol{Z}^{\mathrm{HHO}}, \boldsymbol{u}^{\mathrm{HHO}}, p^{\mathrm{HHO}}, \hat{\boldsymbol{u}}^{\mathrm{HHO}}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}} \times \mathcal{P}^0 \times \hat{\boldsymbol{\mathcal{U}}}_0$ satisfies (6.68). We proceed as for the one-fluid case.

(i) For all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $\boldsymbol{G}_{e,i} \in \boldsymbol{\mathcal{L}}_e^i$, $i \in \{1,2\}$, we have

$$\begin{aligned}(\boldsymbol{G}_e, \boldsymbol{Z}_e^{\mathrm{HHO}})_{\Omega_e} &= \sum_{i\in\{1:2\}} (\boldsymbol{G}_{e,i}, \boldsymbol{Z}_{e,i}^{\mathrm{HHO}})_{\Omega_e^i}\\ &= -\sum_{i\in\{1:2\}} \Big(\sqrt{\mu_e^i}\boldsymbol{G}_e^i, \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big)_{\Omega_e^i}\\ &= -(\sqrt{\mu_e}\boldsymbol{G}_e, \boldsymbol{\nabla}\boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} - \langle\sqrt{\mu_e}\boldsymbol{G}_e\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}} - \boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e}\rangle_{\partial\Omega_e}\\ &\quad + \langle\{\sqrt{\mu_e}\boldsymbol{G}_e\}_{\Upsilon_e}\boldsymbol{n}_\Upsilon, [\![\boldsymbol{u}_e^{\mathrm{HHO}}]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e}\\ &= (\boldsymbol{\nabla}\cdot(\sqrt{\mu_e}\boldsymbol{G}_e), \boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} - \langle\sqrt{\mu_e}\boldsymbol{G}_e\boldsymbol{n}_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\rangle_{\partial\Omega_e} - \langle[\![\sqrt{\mu_e}\boldsymbol{G}_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e},\end{aligned}$$

where we used (6.16), and the identity

$$\sum_{i\in\{1:2\}} \langle\sqrt{\mu_e^i}\boldsymbol{G}_{e,i}\boldsymbol{n}^i, \boldsymbol{u}_{e,i}^{\mathrm{HHO}}\rangle_{\Upsilon_e} = \langle[\![\sqrt{\mu_e}\boldsymbol{G}_e\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} + \langle\{\sqrt{\mu_e}\boldsymbol{G}_e\}_{\Upsilon_e}, [\![\boldsymbol{u}_e^{\mathrm{HHO}}\otimes\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}\rangle_{\Upsilon_e}.$$

Hence, (6.68a) holds.

(ii) For all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $\boldsymbol{w}_{e,i} \in \boldsymbol{\mathcal{U}}_e^i$, $i \in \{1,2\}$, we have

$$\begin{aligned}-(\boldsymbol{\nabla}\boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HHO}})_{\Omega_e} &= -\sum_{i\in\{1:2\}} (\boldsymbol{\nabla}\boldsymbol{w}_{e,i}, \boldsymbol{\Phi}_{e,i}^{\mathrm{HHO}})_{\Omega_e^i}\\ &= \sum_{i\in\{1:2\}} \mu_e^i\Big(\boldsymbol{\nabla}\boldsymbol{w}_{e,i}, \boldsymbol{L}_{e,i}^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big)_{\Omega_e^i} - (\boldsymbol{\nabla}\boldsymbol{w}_{e,i}, p_{e,i}^{\mathrm{HHO}}\boldsymbol{I}_{\mathtt{n_{sd}}})_{\Omega_e^i}\\ &= -\langle\boldsymbol{w}_e, \boldsymbol{\Phi}_e^{\mathrm{HHO}}\boldsymbol{n}_e\rangle_{\partial\Omega_e} + \Big\langle\tau_e^{\mathrm{HHO}}\boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{w}_e|_{\partial\Omega_e}), \boldsymbol{s}_e^{\mathrm{HHO}}(\boldsymbol{u}_e^{\mathrm{HHO}}|_{\partial\Omega_e} - \hat{\boldsymbol{u}}_e^{\mathrm{HHO}})\Big\rangle_{\partial\Omega_e}\\ &\quad - \tau_{\Upsilon_e}^{\mathrm{HHO}}\langle[\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{u}_e^{\mathrm{HHO}}]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e} - \langle[\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, \{\boldsymbol{\Phi}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\boldsymbol{n}_\Upsilon\rangle_{\Upsilon_e}\\ &\quad + a_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), \underline{\boldsymbol{u}}_e^{\mathrm{HHO}}) - b_e^{\mathrm{HHO}}((\boldsymbol{w}_e, \boldsymbol{0}), p_e^{\mathrm{HHO}})\\ &= -\langle\boldsymbol{w}_e, \widehat{\boldsymbol{\Phi}}_e^{\mathrm{HHO}}\boldsymbol{n}_e\rangle_{\partial\Omega_e} - \boldsymbol{s}_{\Upsilon_e}^{\mathrm{HHO}}(\boldsymbol{w}_e, \boldsymbol{u}_e^{\mathrm{HHO}})\\ &\quad - \langle[\![\boldsymbol{w}_e\otimes\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{\Phi}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e} + (\boldsymbol{w}_e, \boldsymbol{f})_{\Omega_e} + \langle\{\boldsymbol{w}_e\}_{\Upsilon_e}, \boldsymbol{g}_\Upsilon\rangle_{\Upsilon_e},\end{aligned}$$

where we used (6.16) and we denote by

$$\boldsymbol{s}_{\Upsilon_e}^{\mathrm{HHO}}(\boldsymbol{w}, \boldsymbol{v}) := \tau_{\Upsilon_e}^{\mathrm{HHO}}\langle[\![\boldsymbol{w}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}, [\![\boldsymbol{v}_e]\!]_{\Upsilon_e}^{\mathrm{HHO}}\rangle_{\Upsilon_e},$$

the HHO penalty term. Hence, (6.68b) holds, if one sets $\tau_{\Upsilon_e}^{\mathrm{HDG}} = \tau_{\Upsilon_e}^{\mathrm{HHO}}$.

(iii) For all $e \in \{1{:}\mathtt{n_{el}}\}$ and all $q_{e,i} \in \mathcal{P}_e^0$, $i \in \{1,2\}$, we have

$$\begin{aligned}(\boldsymbol{\nabla}q_e, \boldsymbol{u}_e^{\mathrm{HHO}})_{\Omega_e} &= \sum_{i\in\{1:2\}} (\boldsymbol{\nabla}q_{e,i}, \boldsymbol{u}_{e,i}^{\mathrm{HHO}})_{\Omega_e^i}\\ &= -b_e^{\mathrm{HHO}}(\underline{\boldsymbol{u}}_e^{\mathrm{HHO}}, q_e) + \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} + \langle[\![q]\!]_{\Upsilon_e}^{\mathrm{HHO}}\boldsymbol{n}_\Upsilon, \{\boldsymbol{u}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e}\\ &= \langle q_e, \hat{\boldsymbol{u}}_e^{\mathrm{HHO}}\cdot\boldsymbol{n}_e\rangle_{\partial\Omega_e} + \langle[\![q\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e},\end{aligned}$$

where we used (6.16), and the identity

$$\sum_{i\in\{1:2\}} \langle q_{e,i}, \boldsymbol{u}_{e,i}^{\mathrm{HHO}}\boldsymbol{n}^i\rangle_{\Upsilon_e} = \langle\{q\}_{\Upsilon_e}, [\![\boldsymbol{u}_e^{\mathrm{HHO}}\otimes\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}\rangle_{\Upsilon_e} + \langle[\![q\boldsymbol{n}]\!]_{\Upsilon_e}^{\mathrm{HDG}}, \{\boldsymbol{u}_e^{\mathrm{HHO}}\}_{\Upsilon_e}\rangle_{\Upsilon_e}.$$

Hence, (6.68c) holds.

(iv) Since the interface terms depend only on the cell variable, the proof of this point is identical to the one-fluid case. Hence, (6.68d) holds, and the proof is complete.

□

Remark 6.5 (Interface stabilization). HHO methods offer optimal error estimates when employing the mixed-order setting, as demonstrated in [23]. The presence of the penalty term on the interface plays a relevant role in the proof. In contrast, the analysis of unfitted methods without a penalty term on the interface remains an open question left to future work.

Chapter 7

Conclusions and perspectives

This Thesis focused on solving the Stokes interface problem using the Hybrid High-Order (HHO) method and the Hybridizable Discontinuous Galerkin (HDG) method. Both approaches rely on geometrically unfitted techniques to handle fluid interfaces and curved external boundaries.

In the first part of the Thesis, we explored the unfitted HHO method coupled with a level-set scheme discretized using isoparametric elements. In particular, we investigated numerically the equilibrium between shear flows at infinity and surface tension. In the second part of the Thesis, we developed an unfitted HDG method coupled with a NURBS-enhanced finite element method (HDG-NEFEM) for the exact description of CAD geometries. In particular, we investigated numerically model problems related to microfluidics. Finally, in the third part of the Thesis, we compared the two methodologies in some simplified settings to highlight the close links between the two approaches.

The main achievements can be summarized as follows:

- The unfitted HHO solver was used to explore two main test cases. In the first one, we solved the equilibrium problem between a pure shear flow (enforced far away from the interface) and surface tension effects. In this case, the equilibrium interface is known to be an ellipse, and the relationship between shear and surface tension is expected to be linear. We investigated numerically the dependence of the ellipse eccentricity on the capillary number (quantifying the ratio of shear to surface tension force). In the second test case, we solved equilibrium problems with more complex boundary conditions, leading to unknown shapes for the equilibrium interface. We introduced a fixed-point iterative procedure to determine the equilibrium interface, and we were able to reduce the normal velocity at the interface by one to two orders of magnitude.

- The unfitted HDG-NEFEM solver was used to explore two main test cases. In the first one, we solved the one-fluid Stokes problem, and we observed robustness with respect to badly cut cells and optimal error estimates. In the second one, we solved the two-fluid Stokes problem within complex geometrical domains, where CAD representations allow for a seamless description without introducing geometrical errors. We employed a degree-adaptivity procedure to achieve results with at least two significant digits. Moreover, we investigated the conditioning of the resulting linear system and studied how a careful choice of the basis functions leads to moderate values.

- We proved that for the one-fluid Stokes problem, the HHO and HDG methods can differ only in the choice of the approximation spaces and of the stabilization operator. Moreover, for the two-fluid Stokes problem, the unfitted HHO and HDG methods can additionally differ in the choice of the penalty method.

Several perspectives for future work can be highlighted concerning either applications or methodological improvements.

Applications

A natural perspective is to embrace more challenging applications in fluid dynamics, in particular, unsteady flows and more complex governing equations, such as the Navier-Stokes equations. A further important step is to extend our approach to three dimensions. While the theoretical foundations for this extension are available, it poses numerical challenges, particularly in handling unfitted interfaces.

Moreover, the unfitted HHO and HDG-NEFEM methods can have important applications owing to their capacity to handle complex (and possibly moving) geometries within high-fidelity simulations. Future applications can include: (i) nonlinear rheology problems, such as Bingham fluids, with applications to mineral liquid foams for building insulation; (ii) multi-fluid flows for oil and gas reservoir modeling and for chemical engineering processes; (iii) fluid dynamics in complex geometrical domains, with applications to medicine (biomedical flows, blood flows in arteries and vessels, and drug delivery), to renewable energy (wind and tidal farms), and to optimal object design (optimizing the hydro- or hemo-dynamics, e.g., for stents in arteries).

Methodology

In view of improving the numerical methodologies and consolidating their mathematical formulations, we identify the following research directions:

- As mentioned above, developing the geometric tools for an efficient implementation of the unfitted mesh procedures in 3D is an important target.

- Studying mathematically the convergence of the fixed-point iterative procedure and investigating more systematically the role played by the various numerical procedures in the convergence process. For instance, the coupling of the unfitted HHO method with the level-set scheme can be improved by employing in the level-set scheme a post-processed velocity field that preserves the divergence-free condition. This modification can play an important role in mitigating the issue of area loss, thereby improving the accuracy and stability of simulations. The role of other procedures, such as the finite element solver to transport the level-set and the numerical approximation of the curvature, also deserves further attention.

- Our research suggests ideas on how to alleviate the local conditioning of the unfitted HDG-NEFEM method. Alternative basis functions or different element-extension techniques can be considered to reduce the sensitivity to cut cells. Such improvements can enhance the overall robustness of the HDG method in addressing a more comprehensive range of complex fluid dynamics problems. These investigations are relevant to the unfitted HHO method as well.

- Tackling moving interfaces remains an unexplored frontier within the unfitted HDG-NEFEM method. Future research could focus on extending the capabilities of this method to effectively handle dynamic interfaces, opening new avenues for applications in fluid dynamics simulations.

References

[1] M. Abdelgawad and A. R. Wheeler, "The digital revolution: A new paradigm for microfluidics," *Advanced Materials*, vol. 21, no. 8, pp. 920–925, 2009.

[2] S. Adjerid, N. Chaabane, T. Lin, and P. Yue, "An immersed discontinuous finite element method for the Stokes problem with a moving interface," *J. Comput. Appl. Math.*, vol. 362, pp. 540–559, 2019.

[3] J. Aghili, S. Boyaval, and D. A. Di Pietro, "Hybridization of mixed high-order methods on general meshes and application to the Stokes equations," *Comput. Methods Appl. Math.*, vol. 15, no. 2, pp. 111–134, 2015.

[4] M. Ainsworth, "Pyramid algorithms for Bernstein-Bézier finite elements of high, nonuniform order in any dimension," *SIAM J. Sci. Comput.*, vol. 36, no. 2, A543–A569, 2014.

[5] P. Angot, C.-H. Bruneau, and P. Fabrie, "A penalization method to take into account obstacles in incompressible viscous flows," *Numer. Math.*, vol. 81, no. 4, pp. 497–520, 1999.

[6] D. N. Arnold and F. Brezzi, "Mixed and nonconforming finite element methods : Implementation, postprocessing and error estimates," *Modél. Math. Anal. Numér.*, vol. 19, no. 1, pp. 7–32, 1985.

[7] D. Avraam and A. Payatakes, "Flow regimes and relative permeabilities during steady-state two-phase flow in porous media," *Journal of Fluid Mechanics*, vol. 293, no. 2, pp. 207–236, 1995.

[8] S. Badia, F. Verdugo, and A. F. Martín, "The aggregated unfitted finite element method for elliptic problems," *Comput. Methods Appl. Mech. Eng.*, vol. 336, pp. 533–553, 2018.

[9] J. W. Barrett and C. M. Elliott, "Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces," *IMA J. Numer. Anal.*, vol. 7, no. 3, pp. 283–300, 1987.

[10] M. Bayareh, M. N. Ashani, and A. Usefian, "Active and passive micromixers: A comprehensive review," *Chem. Eng. Process.: Process Intensif.*, vol. 147, 2020.

[11] É. Béchet, N. Moës, and B. Wohlmuth, "A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method," *Int. J. Numer. Methods Eng.*, vol. 78, no. 8, pp. 931–954, 2009.

[12] R. Becker, E. Burman, and P. Hansbo, "A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity," *Comput. Methods Appl. Mech. Engrg.*, vol. 198, no. 41-44, pp. 3352–3360, 2009.

[13] E. Bezchlebová, V. Dolejší, and M. Feistauer, "Discontinuous Galerkin method for the solution of a transport level-set problem," *Comput. Math. Appl.*, vol. 72, no. 3, pp. 455–480, 2016.

[14] S. P. Bordas, E. Burman, M. G. Larson, and M. A. Olshanskii, *Geometrically unfitted finite element methods and applications: Proceedings of the UCL workshop 2016*. Springer, 2018, vol. 121.

[15] L. Botti, D. A. Di Pietro, and J. Droniou, "A hybrid high-order method for the incompressible Navier-Stokes equations based on Temam's device," *J. Comput. Phys.*, vol. 376, pp. 786–816, 2019.

[16] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *J. Comput. Phys.*, vol. 100, no. 2, pp. 335–354, 1992.

[17] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods* (Texts in Applied Mathematics). Springer, 2008, vol. 15.

[18] E. Burman, "Ghost penalty; [La pénalisation fantôme]," *C. R. Math.*, vol. 348, no. 21-22, pp. 1217–1220, 2010.

[19] E. Burman, "Projection stabilization of Lagrange multipliers for the imposition of constraints on interfaces and boundaries," *Numer. Meth. Part. D. E.*, vol. 30, no. 2, pp. 567–592, 2014.

[20] E. Burman, M. Cicuttin, G. Delay, and A. Ern, "An unfitted hybrid high-order method with cell agglomeration for elliptic interface problems," *SIAM J. Sci. Comput.*, vol. 43, no. 2, A859–A882, 2021.

[21] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing, "CutFEM: Discretizing geometry and partial differential equations," *Internat. J. Numer. Methods Engrg.*, vol. 104, no. 7, pp. 472–501, 2015.

[22] E. Burman, G. Delay, and A. Ern, "An unfitted hybrid high-order method for the Stokes interface problem," *IMA J. Numer. Anal.*, vol. 41, no. 4, pp. 2362–2387, 2021.

[23] E. Burman and A. Ern, "An unfitted hybrid high-order method for elliptic interface problems," *SIAM J. Numer. Anal.*, vol. 56, no. 3, pp. 1525–1546, 2018.

[24] E. Burman and P. Hansbo, "Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method," *Comput. Methods Appl. Mech. Eng.*, vol. 199, no. 41-44, pp. 2680–2686, 2010.

[25] E. Burman and P. Hansbo, "Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method," *Appl. Numer. Math.*, vol. 62, no. 4, pp. 328–341, 2012.

[26] E. Burman and P. Hansbo, "Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem," *ESAIM: Math. Model. Numer. Anal.*, vol. 48, no. 3, pp. 859–874, 2014.

[27] E. Cáceres, J. Guzmán, and M. Olshanskii, "New stability estimates for an unfitted finite element method for two-phase Stokes problem," *SIAM J. Numer. Anal.*, vol. 58, no. 4, pp. 2165–2192, 2020.

[28] K. L. Cascavita, J. Bleyer, X. Chateau, and A. Ern, "Hybrid discretization methods with adaptive yield surface detection for Bingham pipe flows," *J. Sci. Comput.*, vol. 77, no. 3, pp. 1424–1443, 2018.

[29] K. L. Cascavita, F. Chouly, and A. Ern, "Hybrid high-order discretizations combined with Nitsche's method for Dirichlet and Signorini boundary conditions," *IMA J. Numer. Anal.*, vol. 40, no. 4, pp. 2189–2226, 2020.

[30] L. Cattaneo, L. Formaggia, G. F. Iori, A. Scotti, and P. Zunino, "Stabilized extended finite elements for the approximation of saddle point problems with unfitted interfaces," *Calcolo*, vol. 52, no. 2, pp. 123–152, 2015.

[31] M. Cicuttin, A. Ern, and N. Pignet, *Hybrid high-order methods. A primer with application to solid mechanics.* (SpringerBriefs in Mathematics). Cham, Switzerland: Springer Nature, 2021.

[32] B. Cockburn, "Static condensation, hybridization, and the devising of the HDG methods," *Lect. Notes Comp. Sci. Eng.*, vol. 114, pp. 129–177, 2016.

[33] B. Cockburn, D. A. Di Pietro, and A. Ern, "Bridging the hybrid high-order and hybridizable discontinuous Galerkin methods," *ESAIM Math. Model. Numer. Anal.*, vol. 50, no. 3, pp. 635–650, 2016.

[34] B. Cockburn, G. Fu, and W. Qiu, "A note on the devising of superconvergent HDG methods for Stokes flow by M-decompositions," *IMA J. Numer. Anal.*, vol. 37, no. 2, pp. 730–749, 2017.

[35] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, "Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems," *SIAM J. Numer. Anal.*, vol. 47, no. 2, pp. 1319–1365, 2009.

[36] B. Cockburn, J. Gopalakrishnan, N. Nguyen, J. Peraire, and F.-J. Sayas, "Analysis of HDG methods for Stokes flow," *Math. Comp.*, vol. 80, no. 274, pp. 723–760, 2011.

[37] B. Cockburn and K. Shi, "Conditions for superconvergence of HDG methods for Stokes flow," *Math. Comp.*, vol. 82, no. 282, pp. 651–671, 2013.

[38] D. Codony, O. Marco, S. Fernández-Méndez, and I. Arias, "An immersed boundary hierarchical B-spline method for flexoelectricity," *Comput. Methods Appl. Mech. Eng.*, vol. 354, pp. 750–782, 2019.

[39] R. G. Cox, "The deformation of a drop in a general time-dependent fluid flow," *J. Fluid Mech.*, vol. 37, no. 3, pp. 601–623, 1969.

[40] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *T. Am. Math. Soc.*, vol. 277, no. 1, pp. 1–42, 1983.

[41] D. A. Di Pietro, A. Ern, and S. Lemaire, "An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators," *Comput. Methods Appl. Math.*, vol. 14, no. 4, pp. 461–472, 2014.

[42] D. A. Di Pietro and J. Droniou, *The hybrid high-order method for polytopal meshes* (MS&A. Modeling, Simulation and Applications). Springer, Cham, 2020, vol. 19.

[43] D. A. Di Pietro, J. Droniou, and G. Manzini, "Discontinuous skeletal gradient discretisation methods on polytopal meshes," *J. Comput. Phys.*, vol. 355, pp. 397–425, 2018.

[44] D. A. Di Pietro and A. Ern, "A hybrid high-order locking-free method for linear elasticity on general meshes," *Comput. Methods Appl. Mech. Engrg.*, vol. 283, pp. 1–21, 2015.

[45] D. A. Di Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods* (Mathématiques et Applications). Berlin: Springer, 2012, vol. 69.

[46] D. A. Di Pietro, A. Ern, A. Linke, and F. Schieweck, "A discontinuous skeletal method for the viscosity-dependent Stokes problem," *Comput. Methods Appl. Mech. Engrg.*, vol. 306, pp. 175–195, 2016.

[47] D. A. Di Pietro and S. Krell, "A hybrid high-order method for the steady incompressible Navier–Stokes problem," *J. Sci. Comput.*, vol. 74, no. 3, pp. 1677–1705, 2018.

[48] Z. Dong and A. Ern, "Hybrid high-order and weak Galerkin methods for the biharmonic problem," *SIAM J. Numer. Anal.*, vol. 60, no. 5, pp. 2626–2656, 2022.

[49] S. Du and F.-J. Sayas, "An invitation to the theory of the hybridizable discontinuous Galerkin method projections, estimates, tools," *SpringerBriefs Math.*, pp. 1–9, 2019.

[50] H. Egger and J. Schöberl, "A hybrid mixed discontinuous Galerkin finite-element method for convection–diffusion problems," *IMA J. Numer. Anal.*, vol. 30, no. 4, pp. 1206–1234, 2010.

[51] A. Ern and J.-L. Guermond, *Finite Elements I: Approximation and Interpolation* (Texts in Applied Mathematics). Cham, Switzerland: Springer Nature, 2021, vol. 72.

[52] A. Ern and J.-L. Guermond, *Finite Elements II: Galerkin Approximation, Elliptic and Mixed PDEs*. Cham, Switzerland: Springer Nature, 2021, vol. 73.

[53] A. Ern and J.-L. Guermond, *Finite Elements III: First-Order and Time-Dependent PDEs*. Cham, Switzerland: Springer Nature, 2021, vol. 74.

[54] A. Ern, A. F. Stephansen, and P. Zunino, "A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity," *IMA J. Numer. Anal.*, vol. 29, no. 2, pp. 235–256, 2008.

[55] Å. Ervik, K. Y. Lervåg, and S. T. Munkejord, "A robust method for calculating interface curvature and normal vectors using an extracted local level set," *J. Comput. Phys.*, vol. 257, pp. 259–277, 2014.

[56] R. Eymard, T. Gallouët, and R. Herbin, "Finite volume methods," in *Handbook of numerical analysis, Vol. VII*, ser. Handb. Numer. Anal., VII, North-Holland, Amsterdam, 2000, pp. 713–1020.

[57] B. Fraeijs de Veubeke, "Displacement and equilibrium models in the finite element method," in *Stress Analysis*, John Wiley & Sons, 1965.

[58] T.-P. Fries and S. Omerović, "Higher-order accurate integration of implicit geometries," *Int. J. Numer. Methods Eng.*, vol. 106, no. 5, pp. 323–371, 2016.

[59] T.-P. Fries, S. Omerović, D. Schöllhammer, and J. Steidl, "Higher-order meshing of implicit geometries—Part I: Integration and interpolation in cut elements," *Comput. Methods Appl. Mech. Eng.*, vol. 313, pp. 759–784, 2017.

[60] C. Galusinski and P. Vigneaux, "On stability condition for bifluid flows with surface tension: Application to microfluidics," *J. Comput. Phys.*, vol. 227, no. 12, pp. 6140–6164, 2008.

[61] G. Gharib, I. Bütün, Z. Muganlı, G. Kozalak, I. Namlı, S. S. Sarraf, V. E. Ahmadi, E. Toyran, A. J. van Wijnen, and A. Koşar, "Biomedical applications of microfluidic devices: A review," *Biosensors*, vol. 12, no. 11, 2022.

[62] M. Giacomini, A. Karkoulias, R. Sevilla, and A. Huerta, "A superconvergent HDG method for Stokes flow with strongly enforced symmetry of the stress tensor," *J. Sci. Comput.*, vol. 77, no. 3, pp. 1679–1702, 2018.

[63] M. Giacomini, R. Sevilla, and A. Huerta, "HDGlab: An Open-Source Implementation of the Hybridisable Discontinuous Galerkin Method in MATLAB," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1941–1986, 2021.

[64] M. Giacomini, R. Sevilla, and A. Huerta, "Tutorial on hybridizable discontinuous Galerkin (HDG) formulation for incompressible flow problems," in *Modeling in engineering using innovative numerical methods for solids and fluids*, ser. CISM Courses and Lect. Vol. 599, Springer, Cham, 2020, pp. 163–201.

[65] G. Giorgiani, S. Fernández-Méndez, and A. Huerta, "Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations," *Comput. Fluids*, vol. 98, pp. 196–208, 2014.

[66] V. Girault and R. Glowinski, "Error analysis of a fictitious domain method applied to a Dirichlet problem," *Japan J. Ind. and Appl. Math.*, vol. 12, no. 3, pp. 487–514, 1995.

[67] R. Glowinski, T.-W. Pan, and J. Periaux, "A fictitious domain method for Dirichlet problem and applications," *Comput. Methods Appl. Mech. Eng.*, vol. 111, no. 3-4, pp. 283–303, 1994.

[68] R. Glowinski, T.-W. Pan, and J. Periaux, "A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations," *Comput. Methods Appl. Mech. Eng.*, vol. 112, no. 1-4, pp. 133–148, 1994.

[69] J. Gounley, G. Boedec, M. Jaeger, and M. Leonetti, "Influence of surface viscosity on droplets in shear flow," *J. Fluid Mech.*, vol. 791, pp. 464–494, 2016.

[70] J.-L. Guermond and M. Nazarov, "A maximum-principle preserving C^0 finite element method for scalar conservation equations," *Comput. Methods Appl. Mech. Engrg.*, vol. 272, pp. 198–213, 2014.

[71] J.-L. Guermond and B. Popov, "Invariant domains and first-order continuous finite element approximation for hyperbolic systems," *SIAM J. Numer. Anal.*, vol. 54, no. 4, pp. 2466–2489, 2016.

[72] J.-L. Guermond, M. Quezada de Luna, and T. Thompson, "A conservative anti-diffusion technique for the level set method," *J. Comput. Appl. Math.*, vol. 321, pp. 448–468, 2017.

[73] C. Gürkan, M. Kronbichler, and S. Fernández-Méndez, "EXtended hybridizable discontinuous Galerkin for incompressible flow problems with unfitted meshes and interfaces," *Internat. J. Numer. Methods Engrg.*, vol. 117, no. 7, pp. 756–777, 2019.

[74] R. J. Guyan, "Reduction of stiffness and mass matrices," *AIAA J.*, vol. 3, no. 2, pp. 380–380, 1965.

[75] A. Hansbo and P. Hansbo, "A finite element method for the simulation of strong and weak discontinuities in solid mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 193, no. 33-35, pp. 3523–3540, 2004.

[76] A. Hansbo and P. Hansbo, "An unfitted finite element method, based on Nitsche's method for elliptic interface problems," *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 47-48, pp. 5537–5552, 2002.

[77] P. Hansbo, M. G. Larson, and S. Zahedi, "A cut finite element method for a Stokes interface problem," *Appl. Numer. Math.*, vol. 85, pp. 90–114, 2014.

[78] J. Haslinger and Y. Renard, "A new fictitious domain approach inspired by the extended finite element method," *SIAM J. Numer. Anal.*, vol. 47, no. 2, pp. 1474–1499, 2009.

[79] F. Heimann, C. Engwer, O. Ippisch, and P. Bastian, "An unfitted interior penalty discontinuous Galerkin method for incompressible Navier-Stokes two-phase flow," *Internat. J. Numer. Methods Fluids*, vol. 71, no. 3, pp. 269–293, 2013.

[80] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Texts in Applied Mathematics). Springer, 2008, vol. 54.

[81] A. Huebner, S. Sharma, M. Srisa-Art, F. Hollfelder, J. B. Edel, and A. J. deMello, "Microdroplets: A sea of applications?" *Lab Chip*, vol. 8, pp. 1244–1254, 2008.

[82] A. Huerta, A. Angeloski, X. Roca, and J. Peraire, "Efficiency of high-order elements for continuous and discontinuous Galerkin methods," *Int. J. Numer. Methods Eng.*, vol. 96, no. 9, pp. 529–560, 2013.

[83] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39-41, pp. 4135–4195, 2005.

[84] E. B. Janetti, M. Riva, and A. Guadagnini, "Effects of pore-scale geometry and wettability on two-phase relative permeabilities within elementary cells," *Water (Switzerland)*, vol. 9, no. 4, 2017.

[85] H. Ji and Q. Zhang, "A simple finite element method for Stokes flows with surface tension using unfitted meshes," *Internat. J. Numer. Methods Fluids*, vol. 81, no. 2, pp. 87–103, 2016.

[86] B. Joe and R. B. Simpson, "Corrections to Lee's visibility polygon algorithm," *BIT Numer. Math.*, vol. 27, no. 4, pp. 458–473, 1987.

[87] A. Johansson and M. G. Larson, "A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary," *Numer. Math.*, vol. 123, no. 4, pp. 607–628, 2013.

[88] R. Kushnir, A. Ullmann, and A. Dayan, "Thermodynamic and hydrodynamic response of compressed air energy storage reservoirs: A review," *Rev. Chem. Eng.*, vol. 28, no. 2-3, pp. 123–148, 2012.

[89] M.-J. Lai and L. L. Schumaker, *Spline functions on triangulations* (Encyclopedia of Mathematics and its Applications). Cambridge University Press, Cambridge, UK, 2007, vol. 110.

[90] G. Legrain, "A NURBS enhanced extended finite element approach for unfitted CAD analysis," *Comput. Mech.*, vol. 52, no. 4, pp. 913–929, 2013.

[91] C. Lehrenfeld, "Hybrid Discontinuous Galerkin methods for solving incompressible flow problems," Ph.D. dissertation, RWTH Aachen University, 2010.

[92] C. Lehrenfeld and A. Reusken, "Optimal preconditioners for Nitsche-XFEM discretizations of interface problems," *Numer. Math.*, vol. 135, no. 2, pp. 313–332, 2017.

[93] O. Luévano-Rivas and F. Valdés-Parada, "Upscaling immiscible two-phase dispersed flow in homogeneous porous media: A mechanical equilibrium approach," *Chem. Eng. Sci.*, vol. 126, pp. 116–131, 2015.

[94] A. Main and G. Scovazzi, "The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems," *J. Comput. Phys.*, vol. 372, pp. 972–995, 2018.

[95] A. Main and G. Scovazzi, "The shifted boundary method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations," *J. Comput. Phys.*, vol. 372, pp. 996–1026, 2018.

[96] O. Marco, R. Sevilla, Y. Zhang, J. J. Ródenas, and M. Tur, "Exact 3D boundary representation in finite element analysis based on Cartesian grids independent of the geometry," *Int. J. Numer. Methods Eng.*, vol. 103, no. 6, pp. 445–468, 2015.

[97] A. Massing, M. G. Larson, A. Logg, and M. E. Rognes, "A stabilized Nitsche fictitious domain method for the Stokes problem," *J. Sci. Comput.*, vol. 61, no. 3, pp. 604–628, 2014.

[98] A. d. Montlaur, S. Fernández-Méndez, and A. Huerta, "Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations," *Int. J. Numer. Methods Fluids*, vol. 57, no. 9, pp. 1071–1092, 2008.

[99] H. Navarro-García, R. Sevilla, E. Nadal, and J. J. Ródenas, "High-order discontinuous Galerkin method for time-domain electromagnetics on geometry-independent Cartesian meshes," *Int. J. Numer. Methods Eng.*, vol. 122, no. 24, pp. 7632–7663, 2021.

[100] N. C. Nguyen, J. Peraire, and B. Cockburn, "A hybridizable discontinuous Galerkin method for Stokes flow," *Comput. Methods Appl. Mech. Eng.*, vol. 199, no. 9-12, pp. 582–597, 2010.

[101] N. C. Nguyen, J. Peraire, and B. Cockburn, "An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations," *J. Comput. Phys.*, vol. 230, no. 4, pp. 1147–1170, 2011.

[102] V. H. Nguyen, A. P. Sheppard, M. A. Knackstedt, and W. V. Pinczewski, "The effect of displacement rate on imbibition relative permeability and residual saturation," *J. Pet. Sci. Eng.*, vol. 52, no. 1-4, pp. 54–70, 2006.

[103] J. Nitsche, "Über ein Variationsprinzip zur Lösung von Dirichlet Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind," *Abh. Math. Semin. der Univ. Hambg.*, vol. 36, no. 1, pp. 9–15, 1971.

[104] C. Noiriel and C. Soulaine, "Pore-scale imaging and modelling of reactive flow in evolving porous media: Tracking the dynamics of the fluid–rock interface," *Transport Porous Med.*, vol. 140, no. 1, pp. 181–213, 2021.

[105] E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," *J. Comput. Phys.*, vol. 210, no. 1, pp. 225–246, 2005.

[106] E. Olsson, G. Kreiss, and S. Zahedi, "A conservative level set method for two phase flow. II," *J. Comput. Phys.*, vol. 225, no. 1, pp. 785–807, 2007.

[107] S. Osher and R. P. Fedkiw, "Level set methods: An overview and some recent results," *J. Comput. Phys.*, vol. 169, no. 2, pp. 463–502, 2001.

[108] S. Piccardo and A. Ern, "Surface tension effects between two immiscible Stokes fluids: A computational study using unfitted hybrid high-order methods and a level-set scheme," *HAL*, 2023.

[109] D. Picchi and I. Battiato, "The impact of pore-scale flow regimes on upscaling of immiscible two-phase flow in porous media," *Water Resour. Res.*, vol. 54, no. 9, pp. 6683–6707, 2018.

[110] S. Popinet, "Numerical models of surface tension," in *Annual review of fluid mechanics. Vol. 50*, ser. Annu. Rev. Fluid Mech. Vol. 50, Annual Reviews, Palo Alto, CA, 2018, pp. 49–75.

[111] R. Poya, R. Sevilla, and A. J. Gil, "A unified approach for a posteriori high-order curved mesh generation using solid mechanics," *Comput. Mech.*, vol. 58, no. 3, pp. 457–490, 2016.

[112] A. Reusken, "Analysis of an extended pressure finite element space for two-phase incompressible flows," *Comput. Vis. Sci.*, vol. 11, no. 4-6, pp. 293–305, 2008.

[113] J. A. Sethian and P. Smereka, "Level set methods for fluid interfaces," *Annu. Rev. Fluid Mech.*, vol. 35, pp. 341–372, 2003.

[114] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "Comparison of high-order curved finite elements," *Int. J. Numer. Methods Eng.*, vol. 87, no. 8, pp. 719–734, 2011.

[115] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "NURBS-enhanced finite element method (NEFEM)," *Internat. J. Numer. Methods Engrg.*, vol. 76, no. 1, pp. 56–83, 2008.

[116] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "NURBS-enhanced finite element method (NEFEM). A seamless bridge between CAD and FEM," *Arch. Comput. Methods Eng.*, vol. 18, no. 4, pp. 441–484, 2011.

[117] R. Sevilla and A. Huerta, "HDG-NEFEM with degree adaptivity for Stokes flows," *J. Sci. Comput.*, vol. 77, no. 3, pp. 1953–1980, 2018.

[118] R. Sevilla, L. Rees, and O. Hassan, "The generation of triangular meshes for NURBS-enhanced FEM," *Int. J. Numer. Methods Eng.*, vol. 108, no. 8, pp. 941–968, 2016.

[119] W. E. H. Sollie, O. Bokhove, and J. J. W. van der Vegt, "Space-time discontinuous Galerkin finite element method for two-fluid flows," *J. Comput. Phys.*, vol. 230, no. 3, pp. 789–817, 2011.

[120] H. A. Stone and S. Kim, "Microfluidics: Basic issues, applications, and challenges," *AIChE J.*, vol. 47, no. 6, pp. 1250–1254, 2001.

[121] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, "An adaptive level set approach for incompressible two-phase flows," *J. Comput. Phys.*, vol. 148, no. 1, pp. 81–124, 1999.

[122] M. Sussman, "A level set approach for computing solutions to incompressible two-phase flow," *J. Comput. Phys.*, vol. 114, no. 1, pp. 146–159, 1994.

[123] M. Sussman and E. Fatemi, "Efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM J. Sci. Comput.*, vol. 20, no. 4, pp. 1165–1191, 1999.

[124] G. I. Taylor, "The formation of emulsions in definable fields of flow," *Proc. Roy. Soc. London Ser. A*, vol. 146, no. 858, pp. 501–523, 1934.

[125] G. I. Taylor and A. E. Green, "Mechanism of the production of small eddies from large ones," *Proc. Roy. Soc. London Ser. A*, vol. 158, no. 895, pp. 499–521, 1937.

[126] J. Wang, N. Zhang, J. Chen, V. G. J. Rodgers, P. Brisk, and W. H. Grover, "Finding the optimal design of a passive microfluidic mixer," *Lab on a Chip*, vol. 19, no. 21, pp. 3618–3627, 2019.

[127] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, "High-order CFD methods: Current status and perspective," *Int. J. Numer. Methods Fluids*, vol. 72, no. 8, pp. 811–845, 2013.

[128] Z. Q. Xie, R. Sevilla, O. Hassan, and K. Morgan, "The generation of arbitrary order curved meshes for 3D finite element analysis," *Comput. Mech.*, vol. 51, no. 3, pp. 361–374, 2013.